UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Segmentação de sentenças e detecção de disfluências em narrativas transcritas de testes neuropsicológicos

Marcos Vinícius Treviso

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Marcos Vinícius Treviso

Segmentação de sentenças e detecção de disfluências em narrativas transcritas de testes neuropsicológicos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Sandra Maria Aluísio

USP – São Carlos
Janeiro de 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

T814s Treviso, Marcos Vinícius
Segmentação de sentenças e detecção de disfluências
em narrativas transcritas de testes
neuropsicológicos / Marcos Vinícius Treviso;
orientador Sandra Maria Aluísio. -- São Carlos, 2018.
208 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2017.

1. Segmentação de Sentenças. 2. Detecção de Limites
de Sentença. 3. Detecção de Disfluências. 4. Fala
Comprometida. I. Aluísio, Sandra Maria, orient. II.
Título.

Marcos Vinícius Treviso

Sentence segmentation and disfluency detection in narrative transcripts from neuropsychological tests

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Sandra Maria Aluísio

USP – São Carlos
January 2018

RESUMO

TREVISO, M. V. **Segmentação de sentenças e detecção de disfluências em narrativas transcritas de testes neuropsicológicos**. 2018. 208 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Contexto: Nos últimos anos, o Comprometimento Cognitivo Leve (CCL) tem recebido uma grande atenção, pois pode representar um estágio pré-clínico da Doença de Alzheimer (DA). Em termos de distinção entre idosos saudáveis (CTL) e pacientes com CCL, vários estudos têm mostrado que a produção de discurso é uma tarefa sensível para detectar efeitos de envelhecimento e para diferenciar indivíduos com CCL dos saudáveis. Ferramentas de Processamento de Língua Natural (PLN) têm sido aplicadas em transcrições de narrativas em inglês e também em português brasileiro, por exemplo, o ambiente Coh-Matrix-Dementia. **Lacunas:** No entanto, a ausência de informações de limites de sentenças e a presença de disfluências em transcrições impedem a aplicação direta de ferramentas que dependem de um texto bem formado, como *taggers* e *parsers*. **Objetivos:** O objetivo principal deste trabalho é desenvolver métodos para segmentar as transcrições em sentenças e detectar/remover as disfluências presentes nelas, de modo que sirvam como uma etapa de pré-processamento para ferramentas subsequentes de PLN. **Métodos e Avaliação:** Propusemos um método baseado em redes neurais recorrentes convolucionais (RCNNs) com informações prosódicas, morfossintáticas e *word embeddings* para a tarefa de segmentação de sentenças (SS). Já para a detecção de disfluências (DD), dividimos o método e a avaliação de acordo com as categorias de disfluências: (i) para preenchimentos (pausas preenchidas e marcadores discursivos), propusemos a mesma RCNN com as mesmas *features* de SS em conjunto com uma lista pré-determinada de palavras; (ii) para disfluências de edição (repetições, revisões e recomeços), adicionamos *features* tradicionalmente empregadas em trabalhos relacionados e introduzimos um modelo de CRF na camada de saída da RCNN. Avaliamos todas as tarefas intrinsecamente, analisando as *features* mais importantes, comparando os métodos propostos com métodos mais simples, e identificando os principais acertos e erros. Além disso, um método final, chamado DeepBonDD, foi criado combinando todas as tarefas, e foi avaliado extrinsecamente com 9 métricas sintáticas do Coh-Matrix-Dementia. **Conclusão:** Para SS, obteve-se $F_1 = 0.77$ em transcrições de CTL e $F_1 = 0.74$ de CCL, caracterizando o estado-da-arte para esta tarefa em fala comprometida. Para detecção de preenchimentos, obteve-se em média $F_1 = 0.90$ para CTL e $F_1 = 0.92$ para CCL, resultados que estão dentro da margem de trabalhos relacionados da língua inglesa. Ao serem ignorados os recomeços na detecção de disfluências de edição, obteve-se em média $F_1 = 0.70$ para CTL e $F_1 = 0.75$ para CCL. Na avaliação extrínseca, apenas 3 métricas mostraram diferença significativa entre as transcrições de CCL manuais e as geradas pelo DeepBonDD, sugerindo que, apesar das variações de limites de sentença e de disfluências, o DeepBonDD é capaz de gerar transcrições para serem processadas

por ferramentas de PLN.

Palavras-chave: Segmentação de Sentenças, Detecção de Limites de Sentença, Detecção de Disfluências, Fala Comprometida.

ABSTRACT

TREVISIO, M. V. **Sentence segmentation and disfluency detection in narrative transcripts from neuropsychological tests**. 2018. 208 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Background: In recent years, mild cognitive impairment (MCI) has received great attention because it may represent a pre-clinical stage of Alzheimer’s Disease (AD). In terms of distinction between healthy elderly (CTL) and MCI patients, several studies have shown that speech production is a sensitive task to detect aging effects and to differentiate individuals with MCI from healthy ones. Natural language processing tools have been applied to transcripts of narratives in English and also in Brazilian Portuguese, for example, Coh-Metrix-Dementia. **Gaps:** However, the absence of sentence boundary information and the presence of disfluencies in transcripts prevent the direct application of tools that depend on well-formed texts, such as taggers and parsers. **Objectives:** The main objective of this work is to develop methods to segment the transcripts into sentences and to detect the disfluencies present in them (independently and jointly), to serve as a preprocessing step for the application of subsequent Natural Language Processing (NLP) tools. **Methods and Evaluation:** We proposed a method based on recurrent convolutional neural networks (RCNNs) with prosodic, morphosyntactic and word embeddings features for the sentence segmentation (SS) task. For the disfluency detection (DD) task, we divided the method and the evaluation according to the categories of disfluencies: (i) for fillers (filled pauses and discourse marks), we proposed the same RCNN with the same SS features along with a predetermined list of words; (ii) for edit disfluencies (repetitions, revisions and restarts), we added features traditionally employed in related works and introduced a CRF model after the RCNN output layer. We evaluated all the tasks intrinsically, analyzing the most important features, comparing the proposed methods to simpler ones, and identifying the main hits and misses. In addition, a final method, called DeepBonDD, was created combining all tasks and was evaluated extrinsically using 9 syntactic metrics of Coh-Metrix-Dementia. **Conclusion:** For SS, we obtained $F_1 = 0.77$ in CTL transcripts and $F_1 = 0.74$ in MCI, achieving the state of the art for this task on impaired speech. For the filler detection, we obtained, on average, $F_1 = 0.90$ for CTL and $F_1 = 0.92$ for MCI, results that are similar to related works of the English language. When restarts were ignored in the detection of edit disfluencies, $F_1 = 0.70$ was obtained for CTL and $F_1 = 0.75$ for MCI. In the extrinsic evaluation, only 3 metrics showed a significant difference between the manual MCI transcripts and those generated by DeepBonDD, suggesting that, despite result differences in sentence boundaries and disfluencies, DeepBonDD is able to generate transcriptions to be properly processed by NLP tools.

Keywords: Sentence Segmentation, Sentence Boundary Detection, Disfluency Detection, Im-

paired Speech.

LISTA DE ILUSTRAÇÕES

Figura 1 – Trecho de uma transcrição antes e depois da aplicação de métodos de SS e DD.	5
Figura 2 – Relações de cada um dos 4 subprojetos (DeepBonDD, ANNA-Dementia, Coh-Metrix-Dementia, e Ensemble de anomalia por meio de agrupamento de dados para a detecção de exceções), no escopo do projeto Aging@Brazil.	8
Figura 3 – Exemplo de delimitações de sentenças linguísticas e acústicas.	11
Figura 4 – Exemplo de desambiguação de sinais de pontuação.	13
Figura 5 – Exemplo de segmentação de sentenças.	14
Figura 6 – Exemplo de recuperação de pontuação.	16
Figura 7 – Exemplo de avaliação baseada em <i>slots</i>	17
Figura 8 – Organização do corpus da Constituição Brasileira de 1988.	27
Figura 9 – Modelos do <i>word2vec</i>	38
Figura 10 – Modelos do <i>wang2vec</i>	38
Figura 11 – Modelo de skip-gram do <i>fastText</i>	39
Figura 12 – Curva de contorno de <i>pitch</i> de uma sentença no modo declarativa e interrogativa.	41
Figura 13 – Arquitetura de uma MLP com apenas uma camada oculta.	43
Figura 14 – Arquitetura de uma rede neural com uma camada recorrente.	45
Figura 15 – Unidade recorrente desenrolada.	47
Figura 16 – Unidade de LSTM básica.	48
Figura 17 – Arquitetura de uma rede neural com uma camada convolucional 1D.	50
Figura 18 – Arquiteturas das camadas convolucionais de Che <i>et al.</i> (2016).	59
Figura 19 – Modelo de CRF de cadeia linear proposto por Liu <i>et al.</i> (2005a).	62
Figura 20 – Classificador de dois passos proposto por Tilk e Alumäe (2015).	65
Figura 21 – Arquitetura da MLP para tratar SS e DD em conjunto proposta por Cho <i>et al.</i> (2015).	78
Figura 22 – Passos de processamento do DeepBonDD	88
Figura 23 – Sequência de figuras e estória prototípica da Cinderela.	89
Figura 24 – Exemplos de transcrições com e sem marcações da estória da Cinderela.	90
Figura 25 – Quantidade de preenchimentos por pedaço das transcrições.	96
Figura 26 – Quantidade média de preenchimentos por tamanho de sentença.	97
Figura 27 – Quantidade de disfluências de edição por pedaço das transcrições.	99
Figura 28 – Quantidade média de disfluências de edição por tamanho de sentença.	100
Figura 29 – Exemplo de um artigo da constituição após agrupar arquivos por artigos.	102

Figura 30 – Exemplo de um artigo pré-processado da constituição após retirar pistas
lexicais. 102

Figura 31 – Arquitetura da RCNN para o modelo textual e o prosódico. 106

Figura 32 – F_1 para variação de *embeddings* no corpus Cinderela v2. 117

Figura 33 – *SER* para variação de *embeddings* no corpus Cinderela v2. 118

LISTA DE TABELAS

Tabela 1 – Propriedades dos corpúscos utilizados no NIST-RT 2003 para EMD.	32
Tabela 2 – Propriedades dos corpúscos utilizados no NIST-RT 2004 para EMD.	33
Tabela 3 – Propriedades dos corpúscos mais utilizados para SS e DD.	34
Tabela 4 – Resultados para SS por Fraser <i>et al.</i> (2015).	68
Tabela 5 – Estatísticas dos corpúscos utilizados por Treviso, Shulby e Aluísio (2017) . . .	69
Tabela 6 – Resumo das características dos trabalhos de fala preparada para SS.	71
Tabela 7 – Resumo das características dos trabalhos de fala espontânea para SS.	73
Tabela 8 – Resumo das características dos trabalhos de fala comprometida para SS. . .	75
Tabela 9 – Resumo das características dos trabalhos de detecção de disfluências de edição. .	84
Tabela 10 – Informações demográficas dos participantes na avaliação com a estória da Cinderela.	90
Tabela 11 – Estatísticas do corpúscos Cinderela v1.	91
Tabela 12 – Estatísticas do corpúscos Cinderela para cada anotador.	92
Tabela 13 – Concordância entre anotadores para a Cinderela versão 2.	93
Tabela 14 – Estatísticas de preenchimentos do corpúscos Cinderela v2 para cada anotador. .	94
Tabela 15 – Concordância entre anotadores para anotação de preenchimentos.	95
Tabela 16 – Estatísticas de disfluências de edição do corpúscos Cinderela v2.	98
Tabela 17 – Informações demográficas e propriedades do corpúscos BALE.	101
Tabela 18 – Estatísticas dos corpúscos Constituição L e Constituição S.	103
Tabela 19 – Sumário dos Corpúscos utilizados nos experimentos.	103
Tabela 20 – Hiper-parâmetros da RCNN.	114
Tabela 21 – Resultados das <i>baselines</i> em termos de F_1	114
Tabela 22 – Resultados dos <i>baselines</i> em termos de <i>SER</i>	115
Tabela 23 – Resultados de cada conjunto de <i>features</i> para Cinderela v2 e Constituição em termos de F_1	119
Tabela 24 – Resultados para conjuntos de <i>features</i> para Cinderela v2 em termos de <i>SER</i> . .	120
Tabela 25 – Resultados em termos de F_1 para cada método.	122
Tabela 26 – Resultados para os testes de robustez.	124
Tabela 27 – Dicas lexicais mais frequentes nas transcrições de CTL e CCL	125
Tabela 28 – Erros mais frequentes cometidos pela RCNN para transcrições de CTL e CCL	126
Tabela 29 – Trigramas mais frequentes que precedem um limite de sentença.	127
Tabela 30 – Palavras selecionadas para <i>baselines</i> para cada tipo de preenchimento. . . .	130
Tabela 31 – Resultados dos <i>baselines</i> para preenchimentos.	131

Tabela 32 – Resultados de variação de <i>embeddings</i> para preenchimentos.	132
Tabela 33 – Resultados de cada conjunto de <i>features</i> para detecção de preenchimentos. .	134
Tabela 34 – Resultados de cada método para detecção de preenchimentos.	136
Tabela 35 – Acertos e erros mais frequentes para detecção de preenchimentos em trans- crições de CTLs	137
Tabela 36 – Acertos e erros mais frequentes para detecção de preenchimentos em trans- crições de CCLs	137
Tabela 37 – Detecção de pausas preenchidas para transcrições originais e revisadas ao combinar a RCNN com a lista de palavras.	138
Tabela 38 – Resultados dos <i>baselines</i> para disfluências de edição.	142
Tabela 39 – Resultados de variação de <i>embeddings</i> para disfluências de edição.	143
Tabela 40 – Resultados de cada conjunto de <i>features</i> para disfluências de edição.	145
Tabela 41 – Resultados de cada método para disfluências de edição.	148
Tabela 42 – Exemplos de sequências de palavras duplicadas que não são repetições. . . .	150
Tabela 43 – Acertos e erros mais frequentes de acordo com o tamanho das repetições. . .	150
Tabela 44 – Contextos de FPs para repetições de tamanho 2 e 3.	151
Tabela 45 – Contextos de FNs para repetições de tamanho 2 e 3.	152
Tabela 46 – Resultados para cada tamanho de repetição.	153
Tabela 47 – Contextos de erros e acertos mais frequentes de acordo com o tamanho das revisões.	153
Tabela 48 – Resultados para cada tamanho de revisão.	155
Tabela 49 – Resultados para detecção binária de disfluências de edição.	156
Tabela 50 – Avaliação de métricas sintáticas no Coh-Matrix-Dementia.	160

LISTA DE ABREVIATURAS E SIGLAS

AD	Arvore de Decisão
BRNN	Rede Neural Recorrente Bidirecional
CCL	Comprometimento Cognitivo Leve
CER	<i>Classification Error Rate</i>
CNN	Rede Neural Convolucional
CRF	<i>Conditional Random Fields</i>
CTL	Transcrições de Controle
CTS	Conversações Telefônicas
DA	Doença de Alzheimer
DD	Detecção de Disfluências
DI	Diarização de Interlocutor
DSP	Desambiguação de Sinais de Pontuação
EMD	Extração de Metadados
GRU	<i>Gated Recurrent Unit</i>
HMM	Modelo Oculto de Markov
LDC	<i>Linguistic Data Consortium</i>
LSTM	<i>Long-Short Term Memory</i>
ML	Modelo de Língua
MLP	<i>Multilayer Perceptron</i>
NIST	<i>Nacional Institute of Standards and Technology</i>
PLN	Processamento de Língua Natural
PNJ	Programas de Notícias Jornalísticas
RAV	Reconhecimento Automático de Voz
RL	Regressão Logística
RNA	Rede Neural Artificial
RNN	Rede Neural Recorrente
RSP	Recuperação de Sinais de Pontuação
SER	<i>Slot Error Rate</i>
SS	Segmentação de Sentenças
SU	<i>Sentence-like Unit</i>
TFV	Textos Falados em Voz alta
WSJ	<i>Wall Street Journal</i>

SUMÁRIO

1	INTRODUÇÃO	1
1.1	O Cenário e os Problemas de Pesquisa	1
1.2	As Tarefas de Pesquisa	3
1.3	Questões de Pesquisa, Objetivos e Lacunas	7
1.4	Organização da Monografia	9
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Sentença e Enunciado	11
2.1.1	<i>Desambiguação de Sinais de Pontuação</i>	12
2.1.2	<i>Segmentação de Sentenças</i>	14
2.1.3	<i>Recuperação de Sinais de Pontuação</i>	15
2.1.4	<i>Métricas de Avaliação</i>	17
2.2	Deteccção de Disfluências	20
2.2.1	<i>Categorização</i>	21
2.2.1.1	<i>Preenchimentos</i>	21
2.2.1.2	<i>Disfluências de Edição</i>	22
2.2.2	<i>Métricas de Avaliação</i>	24
2.3	Cópus de Fala	26
2.3.1	<i>Projeto FalaBrasil - Constituição</i>	27
2.3.2	<i>Switchboard</i>	28
2.3.3	<i>Projeto Hub-4</i>	29
2.3.4	<i>Projeto Hub-5</i>	30
2.3.5	<i>Projeto TDT</i>	31
2.3.6	<i>Projeto NIST-RT</i>	32
2.3.7	<i>Reuniões (Meetings)</i>	33
2.3.8	<i>Palestras (TED Talks)</i>	34
2.3.9	<i>Sumário dos Córpus da Língua Inglesa</i>	34
2.4	Representação Lexical	35
2.4.1	<i>Word Embeddings</i>	36
2.5	Representação Prosódica	39
2.5.1	<i>Duração e Pausa</i>	40
2.5.2	<i>Pitch</i>	41
2.5.3	<i>Energia</i>	41

2.6	Redes Neurais Artificiais	42
2.6.1	<i>Multilayer Perceptron</i>	42
2.6.2	<i>Redes Neurais Recorrentes</i>	45
2.6.3	<i>Redes Neurais Convolucionais</i>	49
3	TRABALHOS RELACIONADOS	53
3.1	Segmentação de Sentenças	54
3.1.1	<i>Fala Preparada</i>	56
3.1.2	<i>Fala Espontânea</i>	60
3.1.3	<i>Fala Comprometida</i>	67
3.1.3.1	<i>Fraser et al. (2015)</i>	67
3.1.3.2	<i>Treviso, Shulby e Aluísio (2017)</i>	68
3.1.4	<i>Síntese</i>	71
3.2	Detecção de Disfluências	75
3.2.1	<i>Disfluências de Edição</i>	76
3.2.2	<i>Preenchimentos</i>	82
3.2.3	<i>Síntese</i>	84
4	PROJETO DEEPBONDD	87
4.1	Cópus	88
4.1.1	<i>Cópus Cinderela</i>	88
4.1.1.1	<i>Anotação de Limites de Sentença: Cinderela versão 1</i>	90
4.1.1.2	<i>Anotação de Limites de Sentença: Cinderela versão 2</i>	92
4.1.1.3	<i>Anotação de Preenchimentos</i>	93
4.1.1.4	<i>Anotação de Disfluências de Edição</i>	96
4.1.2	<i>Cópus BALE</i>	100
4.1.3	<i>Cópus Constituição</i>	101
4.2	Abordagem RCNN para SS e DD	104
4.2.1	<i>Problemas de Classificação Sequenciais</i>	104
4.2.2	<i>Modelo Neural Recorrente Convolucional</i>	105
4.2.2.1	<i>Camada de Entrada</i>	106
4.2.2.2	<i>Camada Convolucional</i>	107
4.2.2.3	<i>Camada Recorrente Bidirecional</i>	107
4.2.2.4	<i>Camada Totalmente Conectada</i>	108
4.2.2.5	<i>Treinamento</i>	109
4.2.3	<i>Metodologia de Avaliação</i>	110
5	AVALIAÇÕES PARA SEGMENTAÇÃO DE SENTENÇAS	113
5.1	<i>Baselines</i>	113
5.2	<i>Avaliação de Word Embeddings</i>	115

5.3	Importância das Features	118
5.4	Comparação de Métodos	122
5.5	Testes de Robustez	123
5.6	Análise de Erros	124
5.7	Considerações Finais	128
6	AVALIAÇÕES PARA DETECÇÃO DE DISFLUÊNCIAS	129
6.1	Preenchimentos	130
6.1.1	<i>Baselines</i>	130
6.1.2	<i>Avaliação de Word Embeddings</i>	131
6.1.3	<i>Importância das Features</i>	133
6.1.4	<i>Comparação de Métodos</i>	135
6.1.5	<i>Análise de Erros</i>	136
6.2	Disfluências de Edição	139
6.2.1	<i>Modificações na RCNN</i>	139
6.2.2	<i>Baselines</i>	141
6.2.3	<i>Avaliação de Word Embeddings</i>	143
6.2.4	<i>Importância das Features</i>	145
6.2.5	<i>Comparação de Métodos</i>	148
6.2.6	<i>Análise de Erros</i>	149
6.3	Avaliação Extrínseca no Coh-Metrix-Dementia	157
6.4	Considerações Finais	161
7	CONCLUSÕES	163
7.1	Contribuições	163
7.2	Limitações	168
7.3	Trabalhos Futuros	169
	REFERÊNCIAS	173
APÊNDICE A	MANUAL DE ANOTAÇÃO DE PROPOSIÇÕES . . .	189
APÊNDICE B	MANUAL DE ANOTAÇÃO DE DISFLUÊNCIAS . . .	195
B.1	Primeira etapa: Preenchimentos	195
B.2	Segunda etapa: Disfluências de Edição	197
APÊNDICE C	TRANSCRIÇÃO DE CCL APÓS PASSOS DO DEEP- BONDD	201
ANEXO A	NORMAS DE TRANSCRIÇÃO NURC	207

INTRODUÇÃO

1.1 O Cenário e os Problemas de Pesquisa

O envelhecimento da população é uma tendência social bem conhecida nos países desenvolvidos que se tornou cada vez mais destacada nos países em desenvolvimento. No Brasil, por exemplo, a pirâmide populacional está mudando em sua forma de acordo com os censos de 2000 a 2010¹ do Instituto Brasileiro de Geografia e Estatística (IBGE). O crescimento da expectativa de vida agregado com uma alta qualidade de vida é essencial para todo cidadão. Entretanto, isso leva ao surgimento de vários problemas financeiros e sociais, particularmente na saúde, pois o envelhecimento pode vir acompanhado de doenças neurodegenerativas, exigindo novos recursos e instalações médicas. Assim, cresce em importância as pesquisas sobre detecção precoce e gestão de doenças cognitivas que normalmente afetam pessoas com idade avançada.

Recentemente, o estudo da síndrome conhecida como Comprometimento Cognitivo Leve (CCL), que é definida como um declínio cognitivo maior do que o esperado para indivíduos de mesma idade e escolaridade, tornou-se cada vez mais importante. A interferência causada pelo CCL nas atividades do dia a dia é mínima, pois se manifesta somente em situações complexas e não pode ser considerado como um tipo de demência. Entretanto, o tipo mais frequente, CCL amnésico, tem a maior taxa de conversão para Doença de Alzheimer (DA) (CLEMENTE; RIBEIRO-FILHO, 2008), que é a demência mais prevalente no mundo. A DA gera déficits cognitivos sérios o bastante para interferir na vida diária de um indivíduo e pioram com o passar do tempo, causando perda de memória, de habilidades de raciocínio, de julgamento e de comunicação (SHEN *et al.*, 2014).

Como demências são doenças progressivas crônicas, é importante identificá-las o mais cedo possível, pois a detecção precoce proporciona maior chance de sucesso para estratégia de tratamento não-farmacológicas, como treinamento cognitivo, atividade física e socialização (TEI-

¹ <<http://www.ibge.gov.br/apps/populacao/projecao/>>

XEIRA *et al.*, 2012). Apesar de não haver tratamentos que curem essas doenças, o consenso na área é que, quando tratamentos se tornarem disponíveis, será imperativo iniciar o tratamento muito antes que danos clinicamente significativos tenham ocorrido ao cérebro (JARROLD *et al.*, 2010). Assim, melhorar os mecanismos de diagnóstico precoce é fundamental para alterar o desenvolvimento da doença. Quanto ao CCL, o desafio é a identificação do subgrupo de indivíduos que progredirá para DA, o que também dará tranquilidade ao grupo que não evoluirá para DA.

Existem vários testes para identificar demências, como o uso de biomarcadores, imagem de ressonância magnética e neuroimagem molecular (MCKHANN *et al.*, 2011; MAPSTONE *et al.*, 2014), mas nenhum deles são soluções baratas para hospitais públicos. Além disso, a definição de critérios dos diagnósticos é realizada principalmente pelos sintomas cognitivos apresentados pelos pacientes em testes padronizados e por deficiências funcionais na vida diária (MCKHANN *et al.*, 2011).

A linguagem é uma das fontes de informação mais eficiente para avaliar as funções cognitivas. Alterações na linguagem são frequentemente observadas em pacientes com demência e normalmente são as primeiras alterações a serem observadas por eles mesmos e seus familiares. Portanto, a análise automática de produção discursiva é vista como uma solução promissora para ajudar no diagnóstico de CCL, pois sua detecção precoce garante maior sucesso no tratamento de fatores potencialmente reversíveis (MUANGPAISAN; PETCHARAT; SRINONPRASERT, 2012).

Testes neuropsicológicos que requerem algum grau de memorização são usualmente incluídos em testes verbais de memória. Esse é o caso do teste de memória lógica (*Logical Memory Test*), no qual um indivíduo reproduz (ou reconta) uma história depois de ouvi-la. Quanto maior o número de elementos (chamados de *story elements*, que são trechos contínuos da narrativa) recontados da narrativa, maior a pontuação (ou escore) de memória (BAYLES; TOMOEDA, 1993; MORRIS *et al.*, 2006). Em histórias mais longas, clínicos usam proposições, que remetem às principais ideias de uma história, que devem ser lembradas. Em ambos os casos, o teste produz um número de elementos que será o escore do paciente, que é comparado com o escore total da história.

A avaliação da linguagem de outro ponto de vista apresenta, na produção discursiva (principalmente em narrativas, embora descrições também sejam usadas), uma alternativa atraente porque permite a análise de microestruturas linguísticas (ANDREETTA; CANTAGALLO; MARINI, 2012), incluindo componentes fonético-fonológicos, morfossintáticos, semântico-lexicais assim como macro estruturas semântico-pragmáticas. Por ser uma forma natural de comunicação, favorece a observação da funcionalidade do paciente, na vida cotidiana. Além disso, fornece dados para observação da interface linguagem-habilidades cognitivas como funções executivas (planejamento, organização, atualização de dados, monitoramento).

Entretanto, seja no teste de memória lógica ou em testes que usam métricas que avaliam

os níveis linguísticos, as maiores dificuldades de sua implantação são: (i) o tempo requerido para realizar a análise de métricas de cada nível linguístico, uma vez que é realizada de forma manual em clínicas e hospitais; (ii) e a subjetividade do clínico na verificação da presença das principais ideias (ou proposições) da narrativa recontada pelo paciente, pois devem ser levadas em conta as paráfrases. Na língua fala, as paráfrases podem ser vistas como uma atividade de constituição textual, que o falante recorre para reformular etapas do desenvolvimento da sua própria formulação textual (PRETI *et al.*, 1997, Pág. 126).

Publicações recentes têm mostrado que a produção de discurso é uma tarefa sensível para detectar efeitos do envelhecimento e diferenciar indivíduos com CCL de idosos saudáveis, usando o teste Wechsler Logical Memory (WLM) (PRUD'HOMMEAUX; MITCHELL; ROARK, 2011; PRUD'HOMMEAUX; ROARK, 2016). A narrativa original usada nesse teste é curta, permitindo a utilização da saída de sistemas de Reconhecimento Automático de Voz (RAV) para discursos de pacientes, mesmo sem capitalização de palavras e segmentação de sentenças, como mostrado para dados em inglês por Lehr *et al.* (2012).

Ferramentas automáticas de análise de discurso baseadas em recursos de Processamento de Língua Natural (PLN) que visam o diagnóstico de demências já foram desenvolvidas para a língua inglesa há longo tempo (ROARK *et al.*, 2011; YANCHEVA; FRASER; RUDZICZ, 2015; FRASER; MELTZER; RUDZICZ, 2016). No entanto, um ambiente de PLN disponível publicamente, projetado para o português do Brasil, foi apenas recentemente desenvolvido no NILC, em um projeto de mestrado (ALUÍSIO; CUNHA; SCARTON, 2016). Esse trabalho desenvolveu uma ferramenta chamada Coh-Metrix-Dementia² para extrair 73 métricas textuais a partir de transcrições de narrativas transcritas, compreendendo vários níveis de análise linguística de contagem de palavras para semântica e discurso. No entanto, a ausência de limites de sentença e a presença de disfluências em transcrições da fala impedem a aplicação direta de métodos de PLN que dependem de um texto bem formado para funcionarem apropriadamente. No Coh-Metrix-Dementia o próprio pesquisador/clínico deve realizar a segmentação de sentenças e anotar as disfluências nos textos a serem analisados pelas métricas.

1.2 As Tarefas de Pesquisa

Para permitir a funcionalidade correta de ferramentas de PLN que analisam transcrições de fala potencialmente comprometida, propomos um método para enriquecer automaticamente essas transcrições, de modo a deixá-las preparadas para a extração de informação para diagnóstico de demências. O enriquecimento automático nesse caso se deve à segmentação de sentenças (SS) e à detecção de disfluências (DD). O enriquecimento automático de transcrições de fala, ou extração de metadados (EMD), é um tópico avançado de reconhecimento de fala responsável por realizar tarefas de pós-processamento em transcrições de fala (JURAFSKY; MARTIN, 2014).

² <<http://nilc.icmc.usp.br/coh-metrix-dementia/>>

Entre essas tarefas estão incluídas a detecção de limites de sentenças, a remoção de disfluências, a recuperação de capitalização e a atribuição de interlocutor por trecho de fala (conhecida como diarização de interlocutor).

A tarefa de SS consiste basicamente em determinar se entre duas palavras há um limite de sentença. Já a tarefa de DD geralmente se preocupa em encontrar regiões de disfluências e categorizá-las em seus tipos, sendo que a categorização mais comum de disfluências inclui os elementos: (i) preenchimentos (*fillers*), que usualmente são empregados pelo interlocutor para indicar hesitação ou manter o controle de uma conversa como, por exemplo, “ah, hm, bom, então, digo”. (ii) e disfluências de edição, que ocorrem quando o interlocutor faz um enunciado que não é completo e fluente como, por exemplo, “na casa toda” da [Figura 1](#), que foi rephraseada por “no castelo”.

As tarefas de EMD ganharam bastante destaque nas últimas décadas devido à utilização de sistemas de RAV, os quais produzem uma saída formada por uma sequência de palavras em caixa baixa e sem sinais de pontuação, com números representados por extenso e, possivelmente, contendo diversos tipos de disfluências ([BATISTA; MAMEDE, 2011](#)). Enriquecer essas saídas é um passo fundamental não apenas para futuras tarefas de PLN, mas também para melhorar a inteligibilidade do texto gerado, ou seja, a facilidade de leitura de um texto. Uma característica importante para o desempenho dessas tarefas é o tipo de fala das transcrições, por exemplo, [Kolár, Liu e Shriberg \(2009\)](#) verificou que informações prosódicas, como pausas entre palavras, são mais eficientes para segmentar fala preparada do que espontânea.

Nas narrativas tratadas nesse projeto de mestrado, os pacientes recontam a história da Cinderela (dentre outras), a pedido de um avaliador, após a leitura de um livro de figuras sem texto. Além da espontaneidade, as narrativas são compostas de enunciados que podem estar comprometidos devido à natureza inerente dos pacientes, que possuem idade avançada e potenciais problemas cognitivos. Essas características fazem com que os enunciados produzidos contenham muitas disfluências, não respeitando regras de sintaxe, semântica e prosódia, que indicam o uso de pausas para separar enunciados completos. A [Figura 1](#) traz um trecho de uma transcrição manual obtida a partir de uma narrativa da história da Cinderela, a qual possui uma estrutura semelhante à saída de um sistema de RAV, isto é, sem sinais de pontuação e de capitalização, e com números e símbolos escritos na forma extensa. É possível ver na [Figura 1](#) que após a transcrição passar por métodos responsáveis por segmentar sentenças e remover disfluências³, a transcrição resultante é mais mais inteligível, que é o que faz um texto ser mais fácil de entender, e mais apropriada para subseqüentes tarefas de PLN.

A tarefa de segmentação de sentenças (SS) ou detecção de limites de sentença pode ser vista como um caso específico da tarefa de recuperação de sinais de pontuação, a qual tenta não apenas detectar os limites de sentença, mas também os tipos de pontuação que ocorrem nesses lugares. Já foi mostrado por [Jones et al. \(2003\)](#) que a segmentação de sentenças aumenta

³ Remover disfluências corresponde a primeiramente detectá-las para então eliminá-las.

Figura 1 – Trecho de uma transcrição antes e depois da aplicação de métodos de SS e DD.

Fonte: Elaborada pelo autor.

a inteligibilidade ou leiturabilidade (do inglês *readability*) de transcrições de fala em um nível próximo a transcrições anotadas manualmente.

Além disso, [Matusov, Mauser e Ney \(2006\)](#) mostrou que informações de limites de sentença inseridas automaticamente em transcrições de fala espontânea acabam ajudando sistemas de tradução automática. Já [Mrozinski et al. \(2006\)](#) verificou que segmentar apropriadamente transcrições de fala é um passo importante para que métodos de sumarização automática tenham um bom desempenho. Seguindo a mesma linha, segundo [Wang et al. \(2010\)](#), [Hassan et al. \(2014\)](#) a remoção de disfluências produz resultados significativamente melhores para traduções automáticas a partir de textos transcritos automaticamente da fala.

Além de informações textuais serem um forte indicativo de final de sentença e de regiões de disfluências, como por exemplo, na [Figura 1](#), a palavra “e” aparece após limites de sentença frequentemente, e a palavra “que” no trecho “foi que que que ela”. Limites de sentenças e regiões de disfluências estão muitas vezes associadas a informações acústicas ([BEAR; DOWDING; SHRIBERG, 1992](#); [SHRIBERG et al., 2000](#); [BATISTA et al., 2012](#);

FERGUSON; DURRETT; KLEIN, 2015). Mais especificamente, as informações acústicas associadas a estrutura da palavra porém não a identidade da palavra em si, que são conhecidas como informações prosódicas, como duração de pausas, mudança de *pitch* e mudança de energia. Uma vez que essas informações fornecem evidências complementares a uma sequência de palavras, elas fornecem uma fonte adicional potencialmente valiosa de informação para a detecção de limites de sentença e disfluências (LIU, 2004). Além disso, é possível argumentar que informações prosódicas são menos suscetíveis a erros produzidos por sistemas de RAV, fornecendo um conhecimento mais confiável. Por exemplo, analisando apenas informações lexicais, a sentença abaixo:

se o homem soubesse o valor que tem a mulher estaria sempre à sua procura .

poderia ser pontuada de duas formas diferentes, embora as duas sejam possíveis:

se o homem soubesse o valor que tem, a mulher estaria sempre à sua procura .

se o homem soubesse o valor que tem a mulher, estaria sempre à sua procura .

Apesar de muitos idiomas não associarem a vírgula como um limite de sentença, o exemplo nos mostra que essa ambiguidade pode ser resolvida facilmente ao analisar a duração de pausa entre as palavras “tem” e “a”. Na verdade, conforme mostrado por Vassière (1983), informações prosódicas conseguem discernir bem os diferentes tipos de enunciados, como perguntas ou declarações. A combinação entre informações textuais e prosódicas geralmente reflete em um melhor desempenho tanto para SS como para DD (STOLCKE *et al.*, 1998; SHRIBERG *et al.*, 2009; HUANG *et al.*, 2014; FERGUSON; DURRETT; KLEIN, 2015).

Os primeiros métodos propostos na literatura para lidar com as tarefa de SS e DD baseiam-se em modelos generativos para tratar informações textuais, como HMMs (do inglês *Hidden Markov Models*) e modelos de língua (HEEMAN; ALLEN, 1994; CHEN, 1999; GOTOH; RENALS, 2000), e árvores de decisão para tratar informações prosódicas (SHRIBERG; BATES; STOLCKE, 1997; STOLCKE *et al.*, 1998; KIM; WOODLAND, 2003). Métodos discriminativos que levam em consideração a estrutura da predição, como CRFs (do inglês *Conditional Random Fields*), foram logo substituindo os métodos anteriores e melhorando os resultados para ambas as tarefas (LIU *et al.*, 2005a; LIU *et al.*, 2005), juntamente com variações de CRFs (QIAN; LIU, 2013; FERGUSON; DURRETT; KLEIN, 2015) e redes neurais (CHRISTENSEN; GOTOH; RENALS, 2006; HUANG *et al.*, 2014). Nos anos recentes, redes neurais profundas estão ganhando cada vez mais visibilidade entre os métodos de aprendizado de máquina, e com elas, resultados estado-da-arte foram reportados em diversas tarefas de PLN. Para a tarefa de SS, trabalhos mais recentes fazem o uso de arquiteturas convolucionais, que conseguem aprender novas representações a partir da entrada (CHE *et al.*, 2016), e de arquiteturas recorrentes, que conseguem modelar facilmente uma sequência de palavras e as dependências entre elas (TILK;

ALUMÄE, 2015; TILK; ALUMÄE, 2016; HOUGH; SCHLANGEN, 2017). Analogamente, modelos neurais profundos estão sendo bastante utilizados em conjunto com mecanismos de decodificação para detectar regiões de disfluência, com um maior foco em arquiteturas recorrentes (HOUGH; SCHLANGEN, 2015; ZAYATS; OSTENDORF; HAJISHIRZI, 2016; HOUGH; SCHLANGEN, 2017; WANG *et al.*, 2017).

Os métodos propostos nesta dissertação consistem em utilizar ambas as fontes de informação em um modelo de rede neural que é recorrente e convolucional (TREVISIO; SHULBY; ALUÍSIO, 2017), o qual foi avaliado na tarefa de SS para o cenário de fala espontânea e comprometida desta pesquisa e também para fala preparada. O mesmo modelo foi avaliado para DD apenas para o cenário de fala espontânea e comprometida, já que disfluências raramente ocorrem em fala preparada.

1.3 Questões de Pesquisa, Objetivos e Lacunas

O projeto de pesquisa desse mestrado está incluído num projeto de escopo maior que envolve a colaboração de outros alunos de mestrado e doutorado. No projeto maior, chamado de *Aging@Brazil*, serão usadas ferramentas automáticas para auxiliar o estudo e a identificação de pacientes com CCL/DA avaliados pela equipe coordenada pela Profa. Dra. Letícia Lessa Mansur da FMUSP e seus alunos e colaboradores. Os dados utilizados neste projeto de mestrado foram coletados pela doutora Cíntia Toledo (narrativas da estória da Cinderela) e outros alunos da Profa. Letícia coletaram também narrativas do relato imediato e tardio da bateria *Arizona Battery for Communication Disorders of Dementia* (ABCD) (BAYLES; TOMOEDA, 1993), apresentando juntamente as transcrições da fala dos pacientes.

Na [Figura 2](#) são ilustradas as relações de cada projeto, a infraestrutura computacional e de uso de técnicas de aprendizado de máquina para detecção de CCL em estudos longitudinais a serem desenvolvidos no projeto de escopo maior. O projeto DeepBonDD (a **Deep** neural approach to **Boundary and Disfluency Detection**) gera textos adequados ao processamento de ferramentas e métodos de PLN a serem usados nos projetos ANNA-Dementia e Coh-Matrix-Dementia. Os dois últimos geram as *features* a serem usadas no projeto *Ensemble de anomalia por meio de agrupamento de dados para a detecção de exceções*.

Objetivos

O objetivo principal deste projeto de mestrado foi o desenvolvimento e avaliação de métodos de processamento das transcrições, para tratar duas tarefas da área de PLN (segmentação de sentenças e detecção de disfluências), usando recursos da área de EMD. Para alcançar o objetivo geral definimos os seguintes objetivos específicos:

- Desenvolver um método para segmentar sentenças em transcrições de fala comprometida;

Figura 2 – Relações de cada um dos 4 subprojetos (DeepBonDD, ANNA-Dementia, Coh-Metrix-Dementia, e Ensemble de anomalia por meio de agrupamento de dados para a detecção de exceções), no escopo do projeto Aging@Brazil.

Fonte: Elaborada pelo autor.

- Desenvolver um método para detectar disfluências em transcrições de fala comprometida;
- Avaliar os métodos de maneira intrínseca para cada tarefa, e de maneira extrínseca com métricas do Coh-Metrix-Dementia.

Questões

Para alcançar o objetivo geral deste trabalho, definimos as principais questões de pesquisa que este trabalho pretende responder:

- Informações prosódicas são úteis nas tarefas de SS e DD para fala comprometida?
- É possível usar métodos tradicionais de SS e DD para fala comprometida e obter bons resultados?
- É possível usar métodos preditivos com a quantidade de dados que temos atualmente e obter bons resultados?
- Quais informações linguísticas, textuais ou prosódicas mais influenciam as tarefas SS e DD?
- A troca da estória em testes neuropsicológicos, isto é, realizar treinamento com uma estória e avaliar com outra afeta as tarefas de SS?
- Quais são as evidências lexicais que os métodos de SS e DD levam em consideração?
- As tarefas de SS e DD podem ser feitas automaticamente para a extração de métricas no Coh-Metrix-Dementia de forma significativa?

Lacunas

As lacunas que esta dissertação buscou superar são tanto da área médica como de PLN:

- Necessidade de se automatizar exames de baterias neuropsicológicas que trabalham com narrativas para identificação de demências, dada a demanda crescente destes exames nos anos atuais;
- Escassez de métodos de SS e DD para fala comprometida, principalmente para o Português Brasileiro;

1.4 Organização da Monografia

Os capítulos desta monografia de dissertação de mestrado estão dispostos da seguinte maneira: no [Capítulo 2](#) apresentamos os fundamentos para as tarefas de SS e DD, e os fundamentos do método proposto; no [Capítulo 3](#) apresentamos uma revisão vasta da literatura de SS, separando os trabalhos de acordo com o tipo da fala estudada por eles, e também uma vasta revisão da literatura de DD, separando os trabalhos de acordo com o tipo de disfluência: preenchimento ou disfluência de edição. No [Capítulo 4](#) apresentamos o projeto de pesquisa chamado de DeepBonDD, descrevendo os materiais disponíveis e os métodos propostos para as duas tarefas. As avaliações para cada tarefa foram feitas separadamente: no [Capítulo 5](#) realizamos as avaliações para SS, e no [Capítulo 6](#) para DD, dividindo-o em detecção de preenchimentos ([Seção 6.1](#)) e detecção de disfluências de edição ([Seção 6.2](#)). Por fim, as conclusões deste trabalho e perspectivas de trabalhos futuros são apresentadas no [Capítulo 7](#)

FUNDAMENTAÇÃO TEÓRICA

2.1 Sentença e Enunciado

Quando trabalhamos com sistemas de Reconhecimento Automática de Voz (RAV) para converter um áudio em texto, a saída esperada é uma sequência de palavras sem informações de capitalização e pontuação. Para enriquecer essa saída, geralmente são realizadas tarefas de pós-processamento sobre a sequência de palavras, sendo que uma das principais tarefas é segmentar a sequência em sentenças.

Sentenças delimitadas acusticamente geralmente não refletem uma estrutura linguística; elas podem fragmentar sentenças gramaticalmente bem formadas ou unidades semânticas, ou agrupar pedaços de unidades não relacionadas (STOLCKE; SHRIBERG, 1996). Em contraste, sentenças linguísticas delimitam uma ideia completa, mas não representam, necessariamente, uma sentença gramaticalmente bem formada, e nem inícios e fins marcados por longos silêncios ou mudança de conversação (LIU, 2004).

A Figura 3 mostra segmentações em sentenças linguísticas (<s>) e acústicas (<a>) para um mesmo trecho de uma narrativa transcrita, sobre a estória da Cinderela.

Figura 3 – Exemplo de delimitações de sentenças linguísticas e acústicas.

(a) Delimitações linguísticas

a cinderela é tava com o pai né <s>
depois ahn o pai morreu ela ficou
sozinha <s> foi depois foi pra casa da
da madrasta viu <s> uhn ta tinha as as
duas duas filhas feias né e fizeram ela
de faxineira <s>

(b) Delimitações acústicas

a cinderela é <a> tava com o pai né
<a> depois ahn <a> o pai morreu ela
ficou sozinha foi depois foi pra casa da
da madrasta viu <a> uhn ta <a> tinha
as as duas duas filhas feias né <a> e
fizeram ela de faxineira <a>

Fonte: Elaborada pelo autor.

Apesar de segmentos acústicos proverem um modo de delimitar a fala, é possível ver na [Figura 3](#) que existem casos em que sentenças linguísticas e acústicas não correspondem ao mesmo trecho, isto é, não se sobrepõem.

Devido ao fato que na língua falada as sentenças não são tão bem definidas como em textos escritos, [Strassel \(2004a\)](#) definiu as unidades linguísticas nessa modalidade como unidades quase-sentenças (ou SUs - *sentence-like units*), informando que elas diferem das sentenças de textos escritos em várias características; como o fato de SUs geralmente terem maior número de pronomes e serem compostas de fragmentos menores do que sentenças linguísticas.

Além do texto transcrito da fala normalmente conter vários tipos de disfluências, sentenças geradas pela fala espontânea geralmente são acompanhadas de características fonológicas, prosódicas e acústicas não encontradas, naturalmente, em sentenças de textos escritos ([JURAFSKY; MARTIN, 2014](#)).

Definimos sentenças linguísticas da fala como SUs ou enunciados (*utterances*, em inglês¹), os quais podem ser sentenças bem formadas, um trecho, ou uma única palavra. Apesar disso, enunciados são unidades completas ([LIU, 2004](#)). A partir daqui usaremos o termo sentença de modo ambíguo, sendo que a distinção vem do cenário da qual a sentença foi extraída: textos escritos ou de fala (enunciados).

Nesta dissertação, tratamos de transcrições de narrativas recontadas em voz alta por pacientes, ou seja, as transcrições são integrais, seguindo os princípios do NURC/SP No. 338 EF e 331 D2 ([Anexo A](#)) e assim vem acompanhadas pelas anotações de características da fala. Além disso, como comentaremos em detalhes na [Subsubseção 4.1.1.1](#), a segmentação realizada nessas transcrições procurou delimitar sentenças linguísticas bem formadas que fossem alvo de anotação de proposições, tentando evitar segmentos curtos como “*uhn ta*” da [Figura 3b](#).

2.1.1 Desambiguação de Sinais de Pontuação

O conceito de sentença em textos escritos ou falados é importante em diversas tarefas de PLN, como análise morfosintática ([KEPLER; FINGER, 2010](#); [FONSECA; ALUÍSIO, 2016](#)), atribuição de autoria ([MARINHO; HIRST; AMANCIO, 2016](#)), análise de sentimentos ([ANCHI-ÊTA et al., 2015](#); [BRUM; ARAUJO; KEPLER, 2016](#)), sumarização ([NÓBREGA; PARDO, 2016](#)), entre outros. Entretanto, os sinais de pontuação que constituem um limite de sentença são ambíguos; por exemplo, na maioria dos idiomas o ponto final (“.”) geralmente é empregado como um final de sentença, enquanto também pode ser usado em abreviações (Sr. John), acrônimos (E.U.A), números ordinais (0.01), e-mails (john.doe@mail.com), URLs (www.johndoe.com), entre outros ([LÓPEZ; PARDO, 2015](#)).

¹ Utterance. any speech sequence consisting of one or more words and preceded and followed by silence: it may be coextensive with a sentence. Dictionary.com Unabridged. Random House, Inc. <<http://www.dictionary.com/browse/utterance>> (acessado em: 18 de Fevereiro de 2017).

Esse tipo de problema acontece principalmente em textos escritos, os quais possuem sinais de pontuação presentes e informação de capitalização, que ajuda no desempenho de algoritmos de desambiguação (STEVENSON *et al.*, 1999). Os sinais de pontuação presentes podem variar de acordo com seu tipo (ex: “., : ; ? ! — ...”) e podem delimitar uma sentença.

O volume de textos digitalizados vem crescendo exponencialmente com o uso da Internet, sendo a maioria deles de baixa qualidade (BALDWIN; JOSEPH, 2009). Um meio bastante usado para gerar esses textos são sistemas de redes sociais, nos quais os usuários escrevem suas mensagens rapidamente, com possíveis ruídos, que geralmente são tratados por sistemas de normalização (BERTAGLIA; NUNES, 2016).

Apesar da normalização deixar o texto mais coerente, ainda é necessário especificar os trechos que correspondem a sentenças para futuras ferramentas de PLN. A tarefa de Desambiguação de Sinais de Pontuação (DSP) analisa os sinais de pontuação no texto e diz se eles correspondem a um final de sentença. A Figura 4 demonstra a saída em sentenças para uma entrada com pontuações ambíguas.

Figura 4 – Exemplo de desambiguação de sinais de pontuação.

Fonte: Elaborada pelo autor.

É possível ver na Figura 4 que foi necessário verificar que o ponto final do pronome de tratamento “Dr. Humberto” não correspondia a um limite de sentença. Também foi necessário desambiguar os sinais de ponto final do trecho “após às 23 p.m.”. A tarefa de desambiguação pode ser resolvida com o uso de regras criadas com expressões regulares precisas (PALMER; HEARST, 1994; STAMATATOS; FAKOTAKIS; KOKKINAKIS, 1999), aplicadas em textos escritos. Porém, métodos mais recentes focam em resolver um problema de classificação binária usando algoritmos de aprendizado de máquina supervisionados e não supervisionados (READ *et al.*, 2012; LÓPEZ; PARDO, 2015).

Formalizamos matematicamente esse problema com uma sequência de *tokens* $\hat{w} = w_1 w_2 \dots w_n$ presente num documento $\mathcal{D}$, de modo que cada *token* w_i está contido no alfabeto Σ . E definimos *tokens* como sequências de caracteres agrupados que servem como unidade semântica, sendo que essas sequências não são necessariamente separados por espaço.

Para a tarefa de DSP, o alfabeto Σ é composto pelo vocabulário de palavras de *tokens* $\mathcal{V}$ em união com um conjunto de sinais de pontuação $\mathcal{P}$, isto é, $\Sigma = \mathcal{V} \cup \mathcal{P}$, onde $\mathcal{V}$ e $\mathcal{P}$ são conjuntos disjuntos $\mathcal{V} \cap \mathcal{P} = \emptyset$. Sendo que o conjunto de sinais de pontuação $\mathcal{P}$ é definido previamente, como por exemplo $\mathcal{P} = \{. \ ; \ : \ ! \ ?\}$. Portanto, a tarefa de DSP pode ser vista como um problema de classificação binária, para a qual cada $w_i \in \hat{w} \wedge w_i \in \mathcal{P}$ deve ser etiquetado como uma das classes em $\mathcal{Y} = \{B, NB\}$, onde B corresponde a um limite de sentença e NB não.

Resumindo, o objetivo da tarefa de DSP é responder a pergunta: **Dentre os *tokens* de sinais de pontuação no texto, quais deles correspondem a limites de sentença?**

2.1.2 Segmentação de Sentenças

O termo Segmentação de Sentenças (SS) é sinônimo de Detecção de Limites de Sentenças (SBD - *Sentence Boundary Detection*), um termo bastante usado na literatura. Segmentar sentenças é uma tarefa muito similar a desambiguar sinais de pontuação, sendo que ambas tentam quebrar o texto de entrada em unidades sequenciais que correspondem a sentenças. Entretanto, as diferenças entre as duas tarefas são visíveis quando verificamos os textos de entrada. A [Figura 5](#) mostra um exemplo de segmentação de sentenças para a mesma entrada de texto da [Figura 4](#), com a diferença que não possui informações de pontuação e capitalização, e que *tokens* foram normalizados para a forma extensa (vinte e três para 23; doutor para Dr.).

Figura 5 – Exemplo de segmentação de sentenças.

Fonte: Elaborada pelo autor.

Segundo [Jurafsky e Martin \(2014\)](#), a tarefa de SS faz parte da área de Extração de Metadados (EMD) ou Enriquecimento Automático de Transcrições (*Rich Transcriptions*), a qual preocupa-se em enriquecer a saída de um sistema de RAV para aumentar a sua inteligibilidade, e também facilitar o uso de futuras ferramentas de PLN sobre essas transcrições. Encontrar limites de sentença em textos escritos geralmente é mais fácil, pois há sinais de pontuação como pontos finais que ajudam na detecção. Entretanto, detectar limites de sentenças em textos de fala é mais complicado devido a falta de informações como pontuação e capitalização, além da saída de texto ser suscetível a erros de reconhecimento, no caso de serem usados sistemas de RAV

para transcrições automáticas (GOTOH; RENALS, 2000; LIU, 2004). Em contrapartida, textos transcritos da fala contêm informações do áudio que evidenciam limites de sentenças, como por exemplo a duração de pausas (BATISTA *et al.*, 2012).

Além do uso de sistemas de RAV terem aumentado o número de trabalhos de EMD, houve iniciativas governamentais norte-americanas que ajudaram no crescimento da tarefa de SS. A tarefa foi um dos principais eventos anotados no projeto NIST-RT (Subseção 2.3.6), em que realizaram uma série de avaliações com objetivo de melhorar o estado-da-arte para tarefas de EMD (STRASSEL, 2004a).

Uma das possíveis soluções para o problema de segmentar sentenças em textos sem pontuações, é utilizar métodos supervisionados de aprendizado de máquina para detectar se após uma palavra há um limite de sentença. Essa abordagem pode ser feita utilizando informações lexicais vindas da transcrição (STOLCKE; SHRIBERG, 1996), ou até mesmo em conjunto com informações prosódicas vindas do áudio (GOTOH; RENALS, 2000; SHRIBERG *et al.*, 2000), ou ambas.

Para a tarefa de SS, o alfabeto Σ é composto apenas do vocabulário de *tokens* $\mathcal{V}$, isto é, $\Sigma = \mathcal{V}$. Portanto, a tarefa de SS pode ser vista como um problema de classificação binária, para a qual cada $w_i \in \hat{w}$ deve ser etiquetado como uma das classes em $\mathcal{Y} = \{B, NB\}$, onde B corresponde a um limite de sentença e NB não. Em outras palavras, deve ser detectado se após um *token* há um limite de sentença.

Entretanto, o que significa um limite de sentença B pode variar de uma anotação para outra. Geralmente é realizada a função de mapeamento de sinais de pontuação para limites de sentenças $\mathcal{P} \mapsto \mathcal{Y}$ da Equação 2.1.

$$\text{ann}(x) = \begin{cases} B & \text{se } x \in \{. \quad ? \quad ! \quad : \quad ;\} \\ BN & \text{se } x \in \{, \quad / \quad *\} \end{cases} \quad (2.1)$$

Perceba que podemos aplicar o conceito de SS na tarefa de DSP, com textos escritos, onde basta tratar os elementos do conjunto de sinais de pontuação $\mathcal{P}$ como sendo parte do vocabulário de *tokens* $\mathcal{V}$, isto é $\Sigma = \mathcal{V} \cup \mathcal{P}$. E então classificar apenas os *tokens* da sequência $\hat{w}$ que são sinais de pontuação utilizando a mesma função de mapeamento da Equação 2.1.

Resumindo, o objetivo da tarefa de SS é responder a pergunta: **Após quais *tokens* do texto há limites de sentença?**

2.1.3 Recuperação de Sinais de Pontuação

Recuperar sinais pontuação corresponde a primeiramente detectar sinais de pontuação em lugares do texto onde eles não estão presentes, e então determinar o seu tipo. A Recuperação de Sinais de Pontuação (RSP), ou recuperação de pontuação, está relacionada com SS, em que são detectados limites de sentenças mas não seu tipo. Entretanto, é possível recuperar de sinais de

pontuação que não correspondem a um final de sentença, como vírgulas. É possível generalizar a tarefa de RSP para que seja aplicada não apenas em textos transcritos da fala, mas também em textos escritos, os quais podem conter outros sinais de pontuações além dos desejados pela recuperação (ZHOU; ZHUANG, 2004).

Usamos o mesmo exemplo das subseções anteriores para demonstrar o processo de RSP em textos escritos na Figura 6a e em textos transcritos da fala na Figura 6b.

Figura 6 – Exemplo de recuperação de pontuação.

Fonte: Elaborada pelo autor.

No português, assim como em outras línguas, recuperar sentenças interrogativas depende de palavras usadas no começo e/ou no fim da sentença, como por exemplo as questões *wh* (e.g., quem disse o quê?/*who said what?*; quando o computador chegou?/*when the computer arrived?*) (BATISTA; MAMEDE, 2011). É possível ver nesses casos que apenas informações locais podem não ser suficientes para recuperar o sinal de interrogação, sendo necessário o uso de informações de toda a sentença. Esse comportamento é provavelmente ainda mais difícil de ser previsto para casos em que é necessário recuperar sinais de pontuação mais raros, como pontos de exclamação, requerendo um discernimento mais semântico (REYNAR; RATNAPARKHI, 1997). Por outro lado, vírgulas são mais frequentes, mas ainda assim sua predição pode ser complicada porque ela serve para vários propósitos diferentes, e pode ser usada em diferentes contextos sintáticos (BATISTA; MAMEDE, 2011).

A maioria dos trabalhos disponíveis focam na recuperação de pontos finais ou vírgulas para textos transcritos da fala (BEEFERMAN; BERGER; LAFFERTY, 1998; KIM; WOOLAND, 2003; LIU *et al.*, 2006; HUANG *et al.*, 2014), e ainda há vários trabalhos que mostraram a dificuldade de recuperar pontos de interrogação (CHRISTENSEN; GOTOH; RENALS, 2006; BATISTA *et al.*, 2012; TILK; ALUMÄE, 2016; CHE *et al.*, 2016), já que eles ocorrem mais raramente no texto. Um problema semelhante é a detecção de unidades de ação de diálogo, em que o objetivo é segmentar uma sequência de palavras da fala em unidades de diálogo, como declarações, questões e *backchannels* (SHRIBERG *et al.*, 2009). Sendo que para esse problema ainda é necessário desambiguar o sentido dos sinais de pontuação, por exemplo, o trecho "é

- **Cobertura:** O número de *slots* classificados corretamente, dividido pelo número total de *slots* anotados na referência:

$$Cob = \frac{C}{\#Ref} = \frac{C}{C + S + R} \quad (2.3)$$

- **Medida-F:** Média harmônica entre precisão e cobertura:

$$F_1 = 2 \cdot \frac{Prec \cdot Cob}{Prec + Cob} \quad (2.4)$$

- **Classification Error Rate:** O número total de *slots* errados hipotetizados pelo classificador, dividido pelo número total de instâncias ($C^* = VP + VN$):

$$CER = \frac{\#Erros}{\#Instancias} = \frac{I + S + R}{C^* + I + S + R} \quad (2.5)$$

- **Slot Error Rate:** O número total de *slots* errados hipotetizados pelo classificador, dividido pelo número total de *slots* anotados na referência:

$$SER = \frac{\#Erros}{\#Ref} = \frac{I + S + R}{C + S + R} \quad (2.6)$$

Como SS é uma tarefa que sofre de dados desbalanceados, pois há menos limites de sentença (*B*) após os *tokens* do que palavras (ou espaços em branco) (*NB*), é importante informar que calculamos as métricas acima somente para as instâncias positivas, ou seja, não consideramos verdadeiros negativos (VN) como acertos, somente verdadeiros positivos (VP).

Em outras palavras, para a tarefa de segmentação de sentenças é feita a avaliação levando-se em consideração apenas as instâncias que correspondem a limites de sentença. Por exemplo, na [Figura 7](#) não consideramos o acerto de não ter hipotetizado algo após a palavra *w1*.

Outra consideração importante a fazer é em relação ao erro de substituição (*S*) de *slot*, o qual não existe para a tarefa de SS, visto que é um problema binário. Porém, é interessante utilizar essa notação para trabalhos que fazem recuperação de sinais de pontuação. Em suma, para a tarefa de SS abordada nesta dissertação basta argumentar que o número de substituições *S* é zero.

Embora vários trabalhos reportaram as avaliações usando a métrica *CER* ([CHEN, 1999](#); [SHRIBERG et al., 2000](#); [LIU et al., 2005a](#)), ela não é uma avaliação muito justa, pois leva em consideração a quantidade de verdadeiros negativos no denominador, assim como a acurácia.

A métrica *SER* para SS corresponde a métrica proposta na série de avaliações NIST-RT conhecida como *NIST SU error rate* ([LIU; SHRIBERG, 2007](#)), já que não existem *slots* de substituição. Tanto o F_1 como o *CER* são métricas que assumem valores no intervalo entre 0 e 1, porém esse não é o caso para *SER*, o qual pode ter valores maiores que 100%.

Por exemplo, para a tarefa de SS da [Figura 7](#), teríamos uma precisão de 0.50, uma cobertura de 0.50, e consequentemente $F_1 = 0.50$. O valor retornado pelo *CER* é de 0.22, já a

métrica *SER* é muito mais punitiva, retornando uma taxa de erro de 100%, visto que há mais erros do que acertos. Por isso, (MAKHOUL *et al.*, 1999) argumenta que a métrica preferida para SS deve ser o *SER*, já que a medida F_1 faz o sistema parecer muito melhor do que ele realmente é.

Liu e Shriberg (2007) avaliaram métricas que retornam um único escalar e curvas de decisão. Eles verificaram que o desbalanceamento dos dados afeta as métricas, sendo que métricas únicas provêm informações limitadas, enquanto curvas de decisão ilustram melhor qual é o modelo preferível para uma região específica. Dentre as curvas de decisão, as que mostraram melhor comportamento foram as curvas de precisão-cobertura, para as quais simplesmente é plotado o que acontece com a precisão e a cobertura de acordo com o variação do limiar da classificação.

Quando é utilizada uma estratégia de validação cruzada para avaliar o modelo, é necessário verificar se a separação dos dados em *folds* irá deixar a avaliação enviesada. Com esse propósito, Forman e Scholz (2010) estudaram o comportamento de viés e variância da métrica F_1 para a validação cruzada de duas maneiras diferentes:

1. **Média macro (F_{avg}):** Consiste na média da F_1 calculada independentemente para cada k -ésimo *fold*:

$$F_{avg} = \frac{1}{k} \cdot \sum_{i=1}^k F_1^{(i)} \quad (2.7)$$

2. **Média micro (F_μ):** Consiste da F_1 calculada para os valores de acertos e erros acumulados após todos os k *folds*:

$$Prec_\mu = \sum_{i=1}^k \frac{C^{(i)}}{C^{(i)} + S^{(i)} + I^{(i)}} \quad (2.8)$$

$$Cob_\mu = \sum_{i=1}^k \frac{C^{(i)}}{C^{(i)} + S^{(i)} + R^{(i)}} \quad (2.9)$$

$$F_\mu = 2 \cdot \frac{Prec_\mu \cdot Cob_\mu}{Prec_\mu + Cob_\mu} \quad (2.10)$$

Nesse estudo, eles sugeriram o uso da estratégia F_μ , mostrando que ela é menos enviesada para dados desbalanceados, já que é difícil garantir em muitos casos a estratificação dos *folds*, isso é, que todos os *folds* tenham uma distribuição de classes similares. Infelizmente, eles não realizaram o mesmo estudo sobre a métrica *SER*, mas podemos tirar as mesmas conclusões, pois assim como a F_1 , a *SER* também ignora falsos negativos, penalizando o modelo por cometer mais erros.

Nesta dissertação, usamos como métrica de avaliação a F_1 e a *SER*, pois são as métricas mais utilizadas na literatura. Desse modo é possível comparar nossos resultados com outros

trabalhos diretamente, desde que os testes sejam feitos sobre o mesmo cenário. Como utilizamos validação cruzada para avaliar o modelo, adotamos a estratégia de acumular os acertos e erros ao longo dos *folds*, chamada de média micro.

2.2 Detecção de Disfluências

Para entender disfluências, é necessário primeiramente entender o conceito de fluência. Como sintetizado por [Lennon \(2000, pág. 26\)](#), a definição de fluência “funcional” pode ser escrita como:

The rapid, smooth, accurate, lucid, and efficient translation of thought or communicative intention into language under the temporal constraints of on-line processing.

Ou seja, fluência é a maneira efetiva de como usamos a linguagem, com respeito a todos os níveis de gramática. Já o termo disfluência exhibe conotação depreciativa ligada ao erro, contudo, não é evidente se os conceitos de fluência incluem/excluem os mecanismos ditos de disfluência ([MONIZ *et al.*, 2010](#)). Podemos sintetizar considerando que falantes mais fluentes não produzem um discurso fragmentado, e respeitam regras prosódicas que indicam uma continuação discursiva. Já o contrário pode não ser válido para discursos com disfluências.

Disfluências ocorrem frequentemente em conversação natural, caracterizada por fala espontânea ([GODFREY; HOLLIMAN; MCDANIEL, 1992](#); [SHRIBERG, 1994](#)). Quando os interlocutores não conseguem formular um enunciado completo, ou quando mudam sua opinião sobre algo que estão falando no momento, eles podem suspender o enunciado e produzir disfluências, como pausas, repetição ou substituição de palavras já faladas ([LIU, 2004](#)).

Em textos escritos, esse fenômeno não ocorre naturalmente, pois o escritor tem a tendência de racionalizar sobre o trecho escrito e, então, reescrevê-lo novamente. Entretanto, dependendo do gênero linguístico (textos da Web, por exemplo), as disfluências podem ocorrer devido a erros de digitação ([WENGELIN, 2001](#)), ou quando trata-se de conteúdos escritos por usuários, principalmente em *chats*, trechos de repetição ocorrem com frequência ([TUDINI, 2016](#)).

Em vários corpú e línguas, disfluências ocorrem em taxas maiores do que a cada 20 palavras, e podem afetar até um terço dos enunciados ([SHRIBERG, 1994](#)). Apesar de disfluências já terem sido vistas como “erros”, um crescente número de trabalhos da literatura sobre o tópico tem surgido para lidar com elas como sendo uma parte integral da língua natural e da conversação ([SHRIBERG, 2005](#)). Elas refletem tanto aspectos cognitivos de produção linguística, como a gestão de interação ([BEAR; DOWDING; SHRIBERG, 1992](#); [CLARK; WASOW, 1998](#); [LIU, 2004](#)).

A tarefa de Detecção de Disfluências (DD) também faz parte da área de EMD, e preocupa-se em primeiramente detectar disfluências e então removê-las para aumentar a inteligibilidade de uma transcrição (JURAFSKY; MARTIN, 2014). Segundo Moniz (2013), as disfluências podem ser descritas de duas perspectivas principais: (i) como erros pronunciados que interrompem a entrega ideal da fala; (ii) como dispositivos linguísticos fluentes usados para gerenciar a fala.

A terminologia usada para detecção de disfluências é variada, com anotações diferentes. Entre essas anotações, a mais utilizada na literatura é a proposta por Shriberg (1994), em que uma sequência de disfluência é estruturada e categorizada em componentes menores. Nas subseções seguintes apresentamos essa proposta, a qual foi utilizada para a anotação dos nossos corpus de trabalho. Por fim, mostramos quais são as métricas de avaliação propostas de acordo com trabalhos da literatura.

2.2.1 Categorização

No trabalho de Shriberg (1994) foi proposto um conjunto com vários tipos de disfluências. Liu (2004) baseou-se no manual da séries de avaliações NIST-RT (STRASSEL, 2004b), para separar o conjunto de Shriberg (1994) em duas categorias: Preenchimentos (*Fillers*) e Disfluências de edição.

2.2.1.1 Preenchimentos

A categoria de preenchimentos inclui pausas preenchidas, marcadores discursivos e termos de edição explícitos.

Pausas preenchidas são não-lexemas (*non-words*) que falantes empregam para indicar hesitação ou para manter controle de uma conversa enquanto pensam o que falar a seguir (LIU, 2004). Segundo a terminologia usado por Preti *et al.* (1997), as pausas preenchidas são marcadores linguísticos verbais não lexicalizados sem conteúdo semântico e independentes sintaticamente.

Segundo Almeida (2009), no português brasileiro, as pausas preenchidas podem ser: “éh”, “ah”, “ahn”, “mm” ou “uhn”. No exemplo abaixo temos duas delas.

ah a cinderela é a história de uma princesa **ahn** cuja mãe faleceu . e e o pai se casou novamente com **ahn** uma uma outra mulher que era viúva

Marcadores discursivos são uma ou mais palavras que funcionam primariamente como uma unidade estrutural da língua falada (LIU, 2004). Geralmente, aparecem no início ou no fim de um enunciado, marcando a mudança de turno do discurso. Mais especificamente, a definição dada por Zorraquino e Portolés (1999, pág. 4057) é:

“Marcadores discursivos são unidades linguísticas invariáveis que não exercem uma função sintática dentro da predicação oracional (são, portanto, elementos externos às orações) e, no discurso, têm a tarefa de orientar, de acordo com as diferentes propriedades morfosintáticas, semânticas e pragmáticas, as inferências que são feitas na comunicação.”

Ainda, de acordo com os autores, podem ser classificados em cinco categorias: (i) Estruturadores da informação (organizam informativamente o discurso, ordenando a dinâmica entre tópico e comentário); (ii) Conectores (relacionam semântica e pragmaticamente dois ou mais enunciados e subdividem-se em: conectores aditivos (enlaçam os enunciados com a mesma orientação argumentativa); conectores contra-argumentativos (unem enunciados em que o segundo atenua ou anula as inferências que se pudessem obter do primeiro) e conectores da causalidade (associam enunciados que têm uma relação de causa-efeito); (iii) Operadores (condicionam as possibilidades argumentativas dos enunciados em que se encontram). Subclassificam-se em: operadores de reforço argumentativo, operadores que apresentam o enunciado que possuem um exemplo do tema mais geral do discurso e operadores aditivos; (iv) Reformuladores (introduzem os enunciados mais adequados informativamente, já que esclarecem, retificam ou sintetizam os anteriores); (v) Marcadores conversacionais (enlaçam enunciados, destacando aspectos informativos e interativos como relações argumentativas e meta discursivas, mudanças de tema, recursos fáticos ou de recepção da informação, etc.). Seguem exemplos desses últimos: “na verdade”, “veja”, “agora”, “então”, “bom”, “né”, “ok”, etc. Por exemplo:

e ela tinha assim uma vida muito livre **né** . **agora** acontece que ela ela
perdeu o pai dela num acidente

O terceiro tipo de preenchimento é o termo de edição explícito. São expressões que remetem explicitamente a uma mensagem de edição dentro de uma disfluência de edição (Subsubseção 2.2.1.2). Por exemplo, em

“ela foi pra casa **digo** pra festa”

a palavra “digo” é um termo de edição explícito. Segundo Liu (2004), pausas preenchidas e marcadores discursivos também podem funcionar como um término explícito em uma disfluência de edição.

O manual usado nesta dissertação (Apêndice B) para anotar os preenchimentos engloba as mesmas categorias do manual utilizado por Liu (2004) e das avaliações do NIST-RT 2004 (STRASSEL, 2003).

2.2.1.2 Disfluências de Edição

Disfluências de edição ocorrem quando o interlocutor faz um enunciado que não é completo e fluente, de modo que ele próprio corrige ou altera seu enunciado. Disfluências de

edição seguem uma estrutura mais complexa do que preenchimentos (LEVELT, 1989; NAKATANI; HIRSCHBERG, 1994; SHRIBERG, 1994). A estrutura proposta nas avaliações NIST-RT (STRASSEL, 2004a) é composta por quatro unidades – reparando, ponto de interrupção, termo de edição e correção da disfluência:

[(enunciado original) * <termo de edição> correção]

- **Enunciado original:** também chamado de “reparando”, é a região do enunciado que é reparada por completo;
- **Ponto de interrupção (*):** é o ponto em que o interlocutor quebra o enunciado original com, por exemplo, um termo de edição explícito, repetição, revisão ou recomeço;
- **Termo de edição:** é opcional, e consiste de uma pausa preenchida, um marcador discursivo, ou um termo de edição explícito; e
- **Correção:** é o trecho que corrige o enunciado original, sendo o que de fato o interlocutor quis enunciar.

A estrutura interna das disfluências de edição pode ser dividida em quatro subtipos:

- **Repetições:** O interlocutor repete parte do enunciado. Exemplo:
[(pra ela) * <ahn> pra ela] ir pro castelo
- **Revisões:** O interlocutor modifica o enunciado original com uma estrutura sintaticamente similar. Exemplo:
que já tinha [(uma filha) * duas filhas]
- **Recomeços:** O interlocutor abandona completamente o enunciado e começa de novo. Em recomeços, o trecho de correção pode ser vazio, significando que o enunciado original não foi repetido e nem revisto. Exemplo:
[(ela tá) *] aqui é uma noiva
- **Disfluências Complexas:** O interlocutor produz uma série de disfluências seguidas, ou recursivas. Exemplo de repetição dentro de uma revisão:
[(não tô (achando) * achando) * <ahn> não tô vendo] o sapatinho

De acordo com o manual de anotação de disfluências usado nesta dissertação, anotamos as disfluências de edição de modo muito similar ao de Liu (2004), seguindo o protocolo proposto por Strassel (2003), porém especificamos o tipo de cada enunciado corrigido, pois isso pode influenciar nas tarefas de pesquisa futuras nas quais este projeto de mestrado está incluído. Além

disso, não anotamos disfluências complexas. Limitamos a ignorar a disfluência mais abrangente - mantendo a mais interna. Ou seja, se uma repetição ocorreu dentro de uma revisão, anotamos somente a repetição. Mais detalhes sobre o manual pode ser visto no [Apêndice B](#).

2.2.2 Métricas de Avaliação

As métricas utilizadas para a tarefa de DD são essencialmente as mesmas utilizadas para a tarefa de SS (F_1), usando a mesma estratégia de avaliações baseadas em *slots* (*SER*). Há alguns trabalhos que usam a métrica *Edit Word Error Rate* ([LIU, 2004](#)), que é uma variação do *SER* que não leva em consideração as substituições. Porém, nesta dissertação, focamos no uso do F_1 porque é a métrica com a qual a maioria dos trabalhos relacionados reportam seus resultados.

O modo de avaliação para a tarefa de DD depende de como o problema é modelado. Destacamos três modelagens descritas na literatura, sendo que cada uma pode ser avaliada de maneira diferente. Uma coisa em comum entre os vários trabalhos é que, geralmente, as tarefas de detecção de preenchimentos e detecção de disfluências de edição são feitas separadamente.

A primeira modelagem consiste em especificar as classes para detectar os subtipos de disfluências de preenchimento: pausas preenchidas (PP); marcadores discursivos (MD); e termo de edição explícito (TEE). E também as estruturas de disfluências de edição: enunciado original (EO); ponto de interrupção (PI); e termo de edição (TE). É possível também especificar o tipo do enunciado original: repetição (EO-REP); revisão (EO-REV); e recomeço (EO-REC).

A maneira como essas classes podem ser detectadas também varia de algoritmos de aprendizado de máquina a algoritmos baseados em regras. Por exemplo, [Liu \(2004\)](#) detecta primeiramente o ponto de interrupção e então usa um conjunto de regras para voltar palavra-a-palavra até encontrar o início do enunciado original. Outra característica bastante comum na literatura, é a falta de detecção do termo de edição explícito, pois geralmente a ocorrência dele nos corpus é pouco frequente.

De maneira análoga à tarefa de SS, temos uma sequência de *tokens* $\hat{w} = w_1 w_2 \dots w_n$ presente em um documento $\mathcal{D}$, de modo que cada w_i pode ser classificado como uma das etiquetas de $\mathcal{Y}$. Com essa modelagem, formalizamos matematicamente o problema de modo que o conjunto de classes de disfluências de preenchimento $\mathcal{Y}_{pre} = \{PP, MD, TEE, ND\}$, e disfluências de edição $\mathcal{Y}_{ed} = \{EO, PI, TE, ND\}$ ou $\mathcal{Y}_{ed} = \{EO-REP, EO-REV, EO-REC, ND\}$, em que *ND* significa uma etiqueta para *tokens* que não são disfluências. Desse modo, temos um problema de classificação multi-classe (diferente do problema de SS, que é binário).

Como temos um problema de classificação multi-classe, os resultados de F_1 podem ser reportados separadamente para cada classe, ou em conjunto. Para reportar essas métricas em conjunto, podemos utilizar a mesma ideia que usamos para SS em tirar a média do F_1 e *SER* de maneira macro ([Equação 2.11](#)) e micro ([Equação 2.12](#)), com a diferença de que tiramos a média

da métrica específica iterando sobre as classes, e não sobre os *folds*:

$$F_{avg,y} = \frac{1}{|\mathcal{Y}|} \cdot \sum_{i=1}^{|\mathcal{Y}|} F_1^{(i)} \quad (2.11)$$

$$F_{\mu,y} = 2 \cdot \frac{Prec_{\mu,y} \cdot Cob_{\mu,y}}{Prec_{\mu,y} + Cob_{\mu,y}} \quad (2.12)$$

sendo que $Prec_{\mu,y}$ e $Cob_{\mu,y}$ realizam a mesma iteração que $F_{avg,y}$, porém acumulando os erros e acertos de cada classe $y \in \mathcal{Y}$.

Segundo [Sokolova e Lapalme \(2009\)](#), é mais adequado utilizar a média macro para problemas multi-classe quando deseja-se focar nas classes minoritárias. Como avaliamos nossos métodos usando uma estratégia de validação cruzada, ainda temos que tirar a média de todos os *folds*. Isso significa que temos que tirar duas médias: entre classes e entre *folds*.

Uma possível solução para combinar ambas as médias é tirar a média macro para precisão e cobertura entre classes e entre os *k folds*:

$$Prec_{avg} = \frac{1}{k \cdot |\mathcal{Y}|} \sum_{j=1}^k \sum_{i=1}^{|\mathcal{Y}|} Prec^{(i,j)} \quad (2.13)$$

$$Cob_{avg} = \frac{1}{k \cdot |\mathcal{Y}|} \sum_{j=1}^k \sum_{i=1}^{|\mathcal{Y}|} Cob^{(i,j)} \quad (2.14)$$

e então calcular o F_{avg} final usando os valores médios de precisão e cobertura:

$$F_{avg} = 2 \cdot \frac{Prec_{avg} \cdot Cob_{avg}}{Prec_{avg} + Cob_{avg}} \quad (2.15)$$

A segunda maneira de modelar o problema de DD é proposta por [Qian e Liu \(2013\)](#), sendo que simplesmente muda-se o modo como cada *token* de uma disfluência é anotado. Mais especificamente, usam-se 5 classes num esquema conhecido como BIES (*begin-inside-end-single*), em que anotam-se disfluências de preenchimento e de edição com as etiquetas de início de uma disfluência (BE), dentro de uma disfluência (IE), fim de uma disfluência (FE), apenas uma palavra de disfluência (S), e palavras que não são disfluências (ND), ou seja $\mathcal{Y} = \{BE, IE, FE, S, ND\}$. Usando essa modelagem, as métricas são as mesmas usadas na primeira modelagem, com exceção de que geralmente é calculado o F_1 usando média micro entre as classes.

A última maneira permite tratar o problema como binário ($\mathcal{Y} = \{D, ND\}$), em que cada *token* é classificado como sendo parte da disfluência (*D*) ou não (*ND*). Como o problema é equivalente ao problema de classificação binária de SS, as métricas utilizadas podem ser as mesmas que as especificadas na [Subseção 2.1.4](#), inclusive mantendo o *SER* caso desejado.

Como podemos traçar um paralelo entre a nossa anotação e a proposta por [Shriberg \(1994\)](#), utilizada nas avaliações NIST-RT, é possível usar as ideias discutidas para a primeira modelagem. Essa modelagem é interessante caso deseja-se que as próximas tarefas saibam quais são, especificamente, os tipos de disfluências contidas no texto. Mas se as futuras ferramentas aplicadas sobre as narrativas simplesmente optarem por trabalhar sem disfluências independente de sua categoria, então é possível aplicar a segunda e a terceira modelagem para eliminar qualquer tipo de disfluência.

2.3 Córpus de Fala

Nesta seção, são descritos os principais corpúscos usados no escopo de Segmentação de Sentenças (SS) e Detecção de Disfluências (DD), em que o primeiro é para o português brasileiro e os demais são para a língua inglesa. A grande maioria desses conjuntos de dados foram desenvolvidos pelo *Linguistic Data Consortium* (LDC), e são disponibilizados para membros e também a não membros da LDC, após o pagamento de taxas.

Nesta seção, os corpúscos estão organizados de acordo com o seu ano de publicação ou de acordo com o ano de início do projeto no qual eles estavam incluídos. Vários corpúscos de fala preparada e espontânea foram disponibilizados em um mesmo ano através de baterias de avaliações da série *Rich Transcription* do instituto norte-americano *National Institute of Standards and Technology* (NIST).

A produção de transcrições com anotações ricas geralmente envolvem o processo de recuperar informações estruturais e extrair metadados a partir de informações da fala. As principais tarefas de pós-processamento de uma transcrição gerada por um RAV são:

- **Recuperação de Pontuação:** Consiste em colocar pontos finais, vírgulas, pontos de interrogação e exclamação em suas corretas posições na transcrição. E conforme vimos na [Seção 2.1](#), é uma generalização da tarefa de SS ou Detecção de Limites de Sentenças (SBD).
- **Recuperação de Capitalização:** Consiste na recuperação da caixa das palavras, informando, por exemplo, se a primeira letra é maiúscula ou minúscula, ou até mesmo se a palavra corresponde a uma forma composta somente por caracteres maiúsculos.
- **Detecção de Disfluências (DD):** Consiste em detectar as disfluências, anotá-las com etiquetas ou em um arquivo separado e removê-las da transcrição para que futuras ferramentas não sofram com essas peculiaridades da fala.
- **Diarização de Interlocutor (DI):** Consiste em identificar qual é o falante de um diferente trecho da transcrição. É uma tarefa importante pois características de prosódia, como *pitch*, são normalizadas pelos valores produzidos por um mesmo interlocutor.

Lidar com transcrições manuais ou automáticas é um desafio para o PLN, pois trata-se de um estilo de texto diferente do escrito. Além disso, a Extração de Metadados (EMD) é considerada um tópico avançado e recente em PLN, por exemplo, a primeira edição do livro *Speech and Language Processing* não possuía uma seção para esse tópico. Já na última edição (JURAFSKY; MARTIN, 2014), os autores separaram um capítulo especial para tópicos avançados relacionados ao reconhecimento de fala e reservaram uma seção apenas para explicar os conceitos de recuperação de pontuação, recuperação de capitalização, detecção de disfluências e diarização.

Jurafsky e Martin (2014) definem que informações relativas a pontuações, diarização, capitalização e disfluências são metadados da transcrição, e uma transcrição é chamada de rica (RT do inglês *Rich Transcription*) quando esses metadados estão incluídos nela. Ou seja, enquanto um sistema de RAV foca em retornar as palavras corretamente (em nível morfológico) para uma dada entrada de áudio, um sistema de EMD preocupa-se em estruturar a saída do RAV, de modo a maximizar a inteligibilidade para humanos e futuras ferramentas de PLN.

2.3.1 *Projeto FalaBrasil - Constituição*

O FalaBrasil é um projeto do Laboratório de Processamento de Sinais da Universidade Federal do Pará (BATISTA *et al.*, 2013). Nesse projeto foi distribuído um cópus de artigos da Constituição Brasileira de 1998, no qual 1255 textos foram lidos em voz alta e gravados com alta qualidade, sendo que cada uma das gravações tem média 30 segundos de duração. Sem contar as palavras que indicam o começo de conteúdos (como “Título I”, “Capítulo II”, “Artigo V”, etc.), há um total de 63275 palavras nesse cópus.

Como cada arquivo contém apenas 30s de duração, a estrutura do cópus está quebrada em partes, ou seja, um artigo pode começar num arquivo e terminar em outro. Para que seja possível realizar etapas de pré-processamento, é necessário saber a organização da Constituição. Segundo o *website* do governo federal², a organização pode ser vista na Figura 8, sendo que conteúdos referentes as caixas tracejadas podem não existir. A fala do cópus Constituição é caracterizada

Figura 8 – Organização do cópus da Constituição Brasileira de 1988.

Fonte: Elaborada pelo autor.

como preparada, de texto lido em voz alta, o que corresponde de certo modo a uma mescla

² <www.planalto.gov.br/ccivil_03/dicas/Estrutur.htm> (acessado em: 9 de julho de 2016).

entre características de textos escritos e textos transcritos de fala. Isto é, possui informações fonológicas, prosódicas e acústicas, entretanto não possui disfluências e os enunciados geralmente refletem sentenças linguisticamente bem formadas.

2.3.2 *Switchboard*

Com o patrocínio da DARPA (*Defense Advanced Research Projects*), o corpus de fala Switchboard (SWB) de Conversas Telefônicas (CTS) foi originalmente coletado pela *Texas Instruments* de 1990 a 1991 (GODFREY; HOLLIMAN; MCDANIEL, 1992). Essa versão do corpus foi divulgada pelo NIST e distribuída pela LDC de 1992 a 1997. Além disso, nesse período o SWB teve erros encontrados e corrigidos nos arquivos de áudios pela LDC (GODFREY; HOLLIMAN, 1993), resultando num corpus com cerca de 240h de fala gravada e 1.4M de palavras.

Na correção de 1997, o SWB foi usado como base para transcrições melhoradas e publicadas como parte do *Penn Treebank-3 Switchboard Corpus* (MARCUS; MARCINKIEWICZ; SANTORINI, 1993; MARCUS *et al.*, 1999). Foram feitas anotações de sintaxe e disfluência nessa versão do SWB, assim como a divisão de cada diarização em enunciados e a etiquetagem morfosintática em cada palavra dos enunciados (TAYLOR; MARCUS; SANTORINI, 2003).

Como o SWB do Penn Treebank-3 ainda continha erros e não estava alinhado temporalmente com os sinais de fala, o Institute for Signal and Information Processing (ISIP) da Mississippi State University desenvolveu um projeto no qual 1126 transcrições de conversas foram manualmente verificadas e corrigidas (CALHOUN *et al.*, 2010). O alinhamento temporal foi realizado automaticamente com algumas correções manuais (DESHMUKH *et al.*, 1998; HARKINS, 2003). No total há 657 falantes (358 femininos e 299 masculinos) com cerca de 303h de fala gravada, sendo que cada conversa tem em média 6 minutos de duração, resultando em um conjunto com aproximadamente 3M palavras e 156K enunciados.

Diversas correções e anotações foram feitas sobre o SWB usando diferentes formatos por diferentes grupos de pesquisa. Além disso, muitas dessas mudanças foram feitas em subconjuntos disjuntos de transcrições das conversações (CALHOUN *et al.*, 2010). Pelo fato de haver diferentes versões do SWB, a análise dos experimentos obtidos em cada corpus torna-se difícil, pois há discordâncias em diferentes níveis de transcrição e em suas anotações. Um dos últimos trabalhos de manutenção no SWB foi o de Calhoun *et al.* (2010), no qual eles superaram a dificuldade de integrar as diferentes partes existentes das anotações num único conjunto de dados com um formato conhecido como NITE XML Toolkit (CARLETTA *et al.*, 2005).

A LDC compilou ainda outras versões do SWB (conhecidas como Switchboard-2 Fase 2 e Fase 3) para futuras séries de avaliações do NIST, como para o projeto de reconhecimento de falas telefônicas Hub-5. Além dessas, a LDC coletou falas de conversação telefônicas de celulares; o corpus é utilizado em algumas avaliações do NIST e contém aproximadamente 109h

de gravações³; esse cópus será chamado nesta dissertação como Switchboard-C.

Por fim, a maioria dos trabalhos nesta dissertação usaram a versão do SWB modificada pelo ISIP. Para simplificar o entendimento, vamos chamar a versão do SWB original e a versão modificada pelo Penn Treebank-3 de Switchboard-1, e a versão modificada pelo ISIP de Switchboard-2.

2.3.3 *Projeto Hub-4*

O Hub-4 foi um projeto criado pela agência norte-americana DARPA (*Defense Advanced Research Projects*) com o objetivo de promover pesquisas em adaptação às condições de mudança na fala, e também em robustez em relação à degradação da fala (RUDNICKY, 1996).

O Hub-4 tem duas versões disponibilizadas pela LDC:

- O Hub-4 de 1996 contém gravações de fala com as correspondentes transcrições de Programas de Notícias Jornalísticos (PNJ) de televisão e rádio (GRAFF; ALABISO, 1997);
- O Hub-4 de 1997 foi preparado para servir como material alternativo da coleção do ano anterior, e contém 97h de fala de PNJ de rádio e televisão (ALABISO; MACINTYRE; GRAFF, 1998).

Tanto para a versão de 1996 como para a de 1997 as gravações foram transcritas e anotadas manualmente com: (i) alinhamento temporal para cada enunciado; (ii) identificação de limites de enunciados; (iii) mudança de falantes; e (iv) informação sobre o gênero dos interlocutores.

É interessante notar que o NIST realizou uma série de quatro avaliações usando o Hub-4⁴, sendo que as duas primeiras foram nos anos 1996 e 1997 usando o Hub-4 de 1996. Já as duas últimas foram nos anos 1998 e 1999 usando o Hub-4 de 1997. Os conjuntos de treinamento e de teste não são os mesmos em todas as avaliações, sendo que em algumas há programas de notícias jornalísticas novos, e em outros há mais dados de outros cópus da época para o treinamento de modelos acústicos.

Vale notar que há outras versões do Hub-4 para o idioma Espanhol e Mandarim com características diferentes da versão do Inglês. A versão de 1996 tem cerca de 104h de duração de fala, e em conjunto com as gravações foi disponibilizado um conjunto de dados para treinamento de modelos de língua com cerca de 130M palavras. Já a versão de 1997 tem um pouco menos - cerca de 97h de gravações, e segundo Kim e Woodland (2003), há cerca de 770K palavras transcritas dessas gravações.

³ <<https://catalog.ldc.upenn.edu/LDC2001S15>>

⁴ <<http://www.itl.nist.gov/iad/mig/tests/bnr/>>

2.3.4 Projeto Hub-5

O projeto do Hub-5 foi dedicado ao avanço de tecnologias de reconhecimento de fala para os idiomas: Inglês, Árabe, Alemão, Mandarim, e Espanhol. O projeto focou em fala de conversação telefônicas (CTS) com a tarefa particular de transcrever esse tipo de fala em texto. Os objetivos do projeto foram explorar novas áreas no reconhecimento de falas conversacionais e desenvolver novas tecnologias nessas áreas com uma correspondente medição de desempenho (GROUP, 2002).

O Hub-5 é um corpus montado especialmente para uma série de quatro avaliações do NIST para reconhecimento de fala de conversas telefônicas⁵. Para cada avaliação, a LDC montou os seguintes conjuntos de treinamento e de validação para o idioma Inglês:

- **1997:** Para treinamento foi usado todo o Switchboard-1 e as primeiras 100 conversas do corpus CALLHOME (CANAVAN; GRAFF; ZIPPERLEN, 1997). Para a validação foram usadas 20 conversas do corpus CALLHOME.
- **1998:** Para treinamento foi usado todo o Switchboard-1, as primeiras 100 conversas do corpus CALLHOME, e 20 as conversas utilizadas no conjunto de validação do ano anterior. Para a validação foram usadas 20 conversas do Switchboard-2 Fase 1 em conjunto com 20 conversas do corpus CALLHOME.
- **2000:** Para treinamento foi usado todo o Switchboard-1, todo o Switchboard-2 Fase 1, as primeiras 100 conversas do corpus CALLHOME, e as 60 conversas utilizadas nos conjuntos de validação dos anos anteriores. Para a validação foram usadas 20 conversas do Switchboard-2 Fase 1 em conjunto com 20 conversas do corpus CALLHOME.
- **2001:** Para treinamento foi usado todo o Switchboard-1, todo o Switchboard-2 Fase 1, as primeiras 100 conversas do corpus CALLHOME, e todas conversas utilizadas nos conjuntos de validação dos anos anteriores. Para a validação, foram usadas 20 conversas do Switchboard-1, 20 conversas do Switchboard-2 Fase 1, e 20 conversas do corpus CALLHOME.

É importante salientar que os dados usados para validação não estavam contidos no conjunto de treinamento. Além disso, alguns interlocutores de conversas do conjunto de validação também aparecem em conversas que estão no conjunto de treinamento (PRZYBOCKI; MARTIN; LE, 2006).

Embora o objetivo do Hub-5 tenha sido a avaliação de sistemas de RAV, os corpus utilizados por eles tem informações de limites de enunciados, e portanto podem ser usados para a tarefa de SS. Todos os trabalhos relacionados de SS usaram os dados da avaliação de 2000.

⁵ <<http://www.itl.nist.gov/iad/mig/tests/ctr/>>

2.3.5 *Projeto TDT*

A ideia básica para o TDT (*Topic Detection and Tracking*)⁶ originou em 1996, quando a agência americana DARPA percebeu que precisava de tecnologias para determinar a estrutura de tópicos a partir de trechos de notícias sem intervenção humana (WAYNE, 2000). A LDC desenvolveu cinco cópus para apoiar o projeto: TDT-1 a TDT-5. A maioria desses cópus são coleções de PNJs que incluem arquivos de áudio e suas transcrições para o idioma Inglês e Mandarim.

O TDT-1 foi o cópus piloto, usado principalmente para verificar questões de viabilidade. Ele contém cerca de 26K relatos jornalísticos em Inglês de jornais eletrônicos e transcrições de programas de televisão.

O TDT-2 contém aproximadamente 54K relatos jornalísticos de seis jornais eletrônicos e programas de televisão e rádio em Inglês. Os materiais dos programas de televisão e rádio foram salvos em formatos de áudio digital, sendo transcritos posteriormente por humanos e também por sistemas de RAV. Vale lembrar que o cópus também contém relatos de três jornais eletrônicos e programas de televisão em Mandarim.

O TDT-3 contém aproximadamente 31K relatos jornalísticos de oito jornais eletrônicos e programas de televisão em Inglês o qual foi organizado da mesma maneira que O TDT-2. Nessa versão, também há material disponibilizado em Mandarim. A diferença do TDT-3 para o TDT-2 está na forma de construção do cópus, sendo que a LDC usou diferentes critérios de anotação.

O TDT-4 incluiu mais um idioma no seu conjunto: o Árabe. Ele contém aproximadamente 28K relatos jornalísticos de seis jornais eletrônicos e programas de televisão e rádio em Inglês, sendo que também foi organizado conforme a versão anterior. A diferença dessa versão para a antiga foi na definição do que constitui um relato. No total, há 313h de gravações correspondentes aos dados de programas de televisão e rádio.

O TDT-5 foi o último cópus do projeto e é caracterizado apenas por cópus de jornais eletrônicos. Porém, diferentemente das versões anteriores, os dados provindos de jornais eletrônicos foram restringidos levando em consideração limites do significado de um relato (cada documento é um relato). No total, o cópus contém 278K relatos para o idioma Inglês. Lembrando que há também dados para os idiomas Árabe e Mandarim.

Os cópus do projeto TDT estiveram presentes em várias séries de avaliações do NIST (como as RT-03 e RT-04). E, embora não fosse o objetivo do projeto, ele desempenhou um papel importante para pesquisas relacionadas a RAV e EMD (WAYNE, 2000).

⁶ <<http://www.itl.nist.gov/iad/mig/tests/tdt/>>

2.3.6 Projeto NIST-RT

A principal iniciativa para a extração de metadados foi o projeto *Rich Transcription* do programa EARS (*Effective, Affordable, Reusable Speech-to-Text*) (STRASSEL, 2004a), o qual gerou uma série de avaliações coordenadas pelo NIST, conhecidas na literatura como NIST-RT⁷.

O projeto *Rich Transcription* do programa EARS durou cinco anos, e foi patrocinado pela agência norte-americana DARPA. Seu objetivo era avançar o estado-da-arte em tecnologias que transformam automaticamente a fala em transcrições ricas para diferentes idiomas e estilos de fala. Para auxiliar o EARS, a LDC forneceu transcrições de Programas de Notícias Jornalísticos (PNJ) e de Conversas Telefônicas (CTS), criando córpis anotados e diretrizes de anotação para apoiar os tópicos de EMD e RAV.

A série de avaliações NIST-RT promoveu avanços no estado-da-arte em diversas tecnologias para ambas as tarefas: RAV e EMD. Os objetivos mais importantes da série foram o avanço do estado-da-arte e a criação de paradigmas de avaliação flexíveis para as tarefas (novos formatos e novos *softwares*) (BATISTA; MAMEDE, 2011). A série de avaliações NIST-RT começou em 2002 e terminou em 2005. Focamos especialmente nos anos de 2003 e 2004, pois eles consistiram de tarefas de EMD para SBD e DD.

RT 2003

Essa avaliação foi dividida em duas etapas, sendo que a primeira focava em tarefas de RAV (chamada de RT-03S) e a segunda tratava apenas de tarefas de EMD (chamada de RT-03F): DI, SBD, e SBD. A Tabela 1 mostra quais córpis foram utilizados para a tarefa de EMD em Inglês, da etapa RT-03F (não apresentamos dados para a etapa RT-03S pois ela apenas cobriu tópicos de RAV, que não é o foco desta dissertação).

Tabela 1 – Propriedades dos córpis utilizados no NIST-RT 2003 para EMD.

Conjunto	Córpis	Durações	Dur. Total
Treinamento	Hub-4 + TDT-4 + Hub-5	1h + 3h + 1h 40m	5h 40m
Validação	TDT-4 + SWB-1 + SWB-2 + SWB-C	90m + 30m + 30m + 30m	3h
Teste	TDT-4 + Fisher + SWB-C	90m + 45m + 45m	3h

Fonte: Elaborada pelo autor.

Os córpis Hub-4 e Hub-5 são os mesmos da avaliação de 2002. Já o córpis Fisher corresponde a uma distribuição da LDC que foi criada seguindo os protocolos de córpis de CTS anteriores, como o CALLHOME e o Switchboard. Ele foi criado com o intuito de oferecer aos pesquisadores um recurso para construir sistemas de RAV robustos. Em termos quantitativos, o córpis contém 5850 arquivos de áudio, sendo que cada um contém uma conversa de até 10 minutos, resultando em aproximadamente 975 horas de gravações (CIERI *et al.*, 2004).

⁷ <<http://www.nist.gov/speech/tests/rt/>>

RT 2004

Assim como no ano anterior, a avaliação de 2004 foi dividida em duas etapas: uma da primavera (RT-04S) e outra do outono (RT-04F). Na primeira etapa a única tarefa de EMD realizada foi a DI para falas de reuniões. Já na etapa RT-04F foram avaliadas as tarefas de DI, SBD e DD para cópurs de CTS e PNJ. Mais especificamente, os cópurs utilizados para o idioma Inglês na avaliação RT-04F são mostrados na [Tabela 2](#).

Tabela 2 – Propriedades dos cópurs utilizados no NIST-RT 2004 para EMD.

Conjunto	Cópus	Durações	Dur. Total
Treinamento	Hub-4 + SWB-2	20h + 40h	60h
Validação	TDT-4 + E-PNJ + Fisher	180m + 180m + 360m	12h
Teste	E-PNJ + Fisher	180m + 180m	6h

Fonte: Elaborada pelo autor.

O cópus E-PNJ corresponde a um cópus compilado pela EARS em parceria com a LDC para a avaliação de outono. Infelizmente não encontramos informações adicionais sobre ele e, portanto, não sabemos se é composto de outros cópurs de PNJ.

É importante informar que na etapa RT-04F houve avaliações para os idiomas Mandarim e Árabe para cópus de CTS, além do Inglês para cópus de CTS e PNJ. A avaliação dessa etapa é bastante usada por trabalhos relacionados de SS e DD.

RT 2002 e 2005

A avaliação de 2002 foi a primeira série de avaliação. Ela incluía avaliações para RAV e EMD: diarização de interlocutor. Os cópurs de fala utilizados eram compostos de gravações de PNJs e de reuniões. Já a avaliação de 2005 concentrou-se em cópus de reuniões e palestras.

Além dessas avaliações, o NIST também realizou avaliações (fora da série RT) nos anos de 2006, 2007 e 2009. Em todas essas avaliações o foco foi o desenvolvimento de tecnologias para fala de reuniões, mais especificamente para tarefas de RAV e EMD relacionadas diretamente ao interlocutor, como DI.

2.3.7 Reuniões (Meetings)

Há vários cópus de reuniões compostos por gravações e suas respectivas transcrições, como os desenvolvidos pelo laboratório Interactive Systems Laboratories (ISL) da Carnegie Mellon University (LDC2004S05), e também os desenvolvidos pelo NIST em parceria com a LDC, conhecido como NIST-Meeting Pilot Cópus (LDC2004S09). Entretanto, vamos descrever apenas o cópus MRDA (LDC2004S02), pois ele é o mais usado na literatura de SS.

O cópus MRDA (ou ICSI Meeting) é composto por transcrições de 72 horas de fala a partir de 75 reuniões que ocorreram de forma natural. Tem cerca de 53 falantes únicos no

córpus, e uma média de 6 falantes por reunião (SHRIBERG *et al.*, 2004). Esse córpus foi bastante utilizado em avaliações do NIST-RT, especialmente nas versões depois de 2005, que se concentraram em realizar avaliações para falas de reuniões espontâneas.

2.3.8 Palestras (TED Talks)

O córpus de palestras conhecido como TED Talks (TED-T) faz parte do conjunto de dados do International Workshop on Spoken Language Translation (IWSLT). No córpus há pontuação nas transcrições manuais e limites de sentenças nas transcrições produzidas por RAV (CHE *et al.*, 2016). As palestras do TED Talks são disponibilizadas de maneira gratuita na Web⁸.

Cada trabalho relacionado utiliza, geralmente, uma versão de um ano específico do conjunto de dados do IWSLT. Mas a maioria ainda usa a versão do IWSLT2011 para teste, contendo cerca de 13K palavras. E uma divisão da versão IWSLT2012 para treinamento e validação, contendo cerca de 2.1M e 296K palavras, respectivamente.

2.3.9 Sumário dos Córpus da Língua Inglesa

A Tabela 3 sumariza os córpus mais utilizados na literatura da língua inglesa para SS e DD. Para muitos deles não conseguimos encontrar informações relativas às colunas especificadas na tabela.

Tabela 3 – Propriedades dos córpus mais utilizados para SS e DD.

Córpus	Ano	Tipo	Desc.	# Transcrições	# Palavras	# Duração
SWB-1	1997	Espontânea	CTS	2430	~1.4M	~240h
SWB-2	1998	Espontânea	CTS	3638	~3M	~303h
Hub-4	1996	Preparada	PNJ	-	~825K*	~104h
Hub-4	1997	Preparada	PNJ	-	~770K	~97h
Hub-5	2000	Espontânea	CTS	3798	~3.1M*	~316h*
TDT-3	1999	Preparada	PNJ	31K	-	~346h*
TDT-4	2001	Preparada	PNJ	28K	-	~313h
MRDA	2004	Espontânea	Reuniões	75	-	~72h
TED-T	2011	Preparada	Palestras	-	~13K	-
TED-T	2012	Preparada	Palestras	-	~2.3M	-

Fonte: Elaborada pelo autor.

* Valores estimados usando regra de três, conforme a função da Equação 2.16.

Muitos córpus estão com informações faltantes na Tabela 3 porque os respectivos valores não foram encontrados nos artigos da literatura, nem no *website* do projeto ou nas páginas da LDC. Além disso, muitos trabalhos da literatura usaram um subconjunto específico desses córpus e, portanto, não podemos apresentar as informações quantitativas descritas neles. Entretanto,

⁸ <<https://www.ted.com/talks>>

podemos aproximar algumas informações com base em outros corpúsimilares. Assumindo que a distribuição de ambos os corpúsim é a mesma, as quantidades se tornam grandezas diretamente proporcionais, e podemos calcular o valor desconhecido de acordo com a [Equação 2.16](#).

$$approx(x,y) = x_1 = \frac{y_1 \cdot x_2}{y_2} \quad (2.16)$$

Por exemplo, o número de palavras para o corpúsim Hub-4 de 1996 pode ser aproximado usando $approx(\text{Palavras}, \text{Duração})$, em que x_1, y_1 correspondem a quantidade desconhecida de palavras e duração do Hub-1996, e x_2, y_2 correspondem aos valores já conhecidos para o Hub-1997. Mais especificamente, podemos ver que o Hub-4 de 1996 tem cerca de 7% a mais de duração que o Hub-4 de 1997, resultando em ~825K palavras. As quantidades do Hub-5 foram relacionadas com as do SWB-2, já que ambos são falas de CTS e o Hub-5 é composto pelo SWB-2. Já as quantidades do TDT-3 foram relacionadas com as do TDT-4.

Classificamos os corpúsim de CTS e reuniões como fala espontânea, porque os participantes das reuniões são livres para discutir um tópico sem seguir um roteiro, sem ter planejado falar sobre aquele assunto. E os participantes de CTS conversam sobre um assunto específico determinado no momento da ligação telefônica, sem ter planejado um roteiro para a conversa. Desse modo, para ambos os casos é muito provável que existam mais problemas com disfluências do que em corpúsim de fala preparada.

Categorizamos os corpúsim de PNJ e palestras em fala preparada. O motivo para PNJ ser classificado dessa maneira é bastante claro: os participantes (geralmente jornalistas) seguem um *script* do que vão apresentar no programa, como se estivessem lendo em voz alta, o que caracteriza uma fala planejada. Já para os corpúsim de palestras a decisão foi um pouco mais difícil, pois apesar dos palestrantes se programarem para falar em frente do público, muitas vezes pessoas do auditório acabam fazendo perguntas aos palestrantes, fazendo com que eles mudem para a fala espontânea. Apesar disso, classificamos como fala preparada pois, em princípio, a maior parte do tempo da palestra é reservada para o que foi planejado pelo palestrante.

2.4 Representação Lexical

Sequências de palavras contidas em um texto podem ser representadas por *features* lexicais. É possível usar, por exemplo, informações de classes morfossintáticas ou semânticas para representar uma palavra. Ou até mesmo usar n-gramas dessa informações para representar uma sequência de palavras. Vários trabalhos mostraram que as informações lexicais são importantes para detecção de limites de sentença e de disfluências ([GOTOH; RENALS, 2000](#); [SNOVER; DORR; SCHWARTZ, 2004](#); [BATISTA et al., 2007](#); [CHO et al., 2015](#); [CHE et al., 2016](#)).

Apesar de técnicas baseadas em n-gramas obterem bons resultados para SS e DD, elas são representações simbólicas atômicas das palavras. Essa representação é induzida a um espaço

vetorial esparsa, de modo que cada vetor para uma dada palavra tem o mesmo tamanho do vocabulário de treinamento $|\mathcal{V}|$, com o único escalar 1 na posição relativa da palavra em $\mathcal{V}$ e 0 em todas as outras posições. Essa representação é conhecida na literatura como “one-hot”, e tem o problema de não conseguir capturar similaridades entre vetores de palavras (MIKOLOV; YIH; ZWEIG, 2013).

2.4.1 Word Embeddings

Aplicações de PLN geralmente recebem textos como entrada, portanto palavras podem ser consideradas como unidade básica de processamento. Sendo assim, é importante que elas sejam representadas de alguma maneira que carregue informações relevantes.

Com o desenvolvimento de técnicas mais sofisticadas de PLN e aprendizado de máquina, foi natural que surgissem representações mais efetivas de palavras. O trabalho de Bengio *et al.* (2003) foi um dos pioneiros em empregar redes neurais para aprender automaticamente representações vetoriais. Esse tipo de representação são atualmente conhecidas por *word embeddings*. As *word embeddings* conseguem capturar informações semânticas, sintáticas e morfológicas a partir de grandes corpus não anotados (MIKOLOV *et al.*, 2013b; LING *et al.*, 2015; LAI *et al.*, 2015).

A geração dessas representações de palavras está vinculada ao método de indução utilizado. Turian, Ratnoff e Bengio (2010) categoriza as representações nas três categorias seguintes:

1. **Representações baseadas em clusters:** são baseadas na clusterização de palavras em K classes, de modo que cada palavra seja representada por um vetor de tamanho K , onde cada dimensão possui informações mútuas relativas a cada k -ésimo *cluster*. Uma técnica baseada nesse funcionamento é conhecida como Brown Clustering (BROWN *et al.*, 1992), a qual cria *clusters* maximizando a informação mútua entre bigramas.
2. **Representações distribucionais:** são baseadas numa matriz de co-ocorrência de palavras $M \in \mathbb{R}^{|\mathcal{V}| \times C}$, onde cada linha M_w corresponde a uma representação inicial da palavra w , e cada coluna M_c é o contexto da palavra w . O valor de C depende da técnica utilizada para montar a matriz de co-ocorrência, variando o que é considerado como contexto e como a contagem é feita. Mas em geral tenta-se obter uma nova matriz M' , onde cada palavra tem uma dimensão reduzida $d \ll C$ através de uma função de mapeamento $f(M) = M' \in \mathbb{R}^{|\mathcal{V}| \times d}$. Tipicamente utiliza-se decomposição de valores singulares (ou SVD) como a função f de redução de dimensionalidade.

Entre essas técnicas, uma das primeiras a se destacar foi o LSA (DUMAIS *et al.*, 1988; LANDAUER; FOLTZ; LAHAM, 1998), que utiliza SVD sobre uma matriz de co-ocorrência de palavras montada utilizando documentos como contextos de palavras. Outra técnica bastante utilizada é o HAL (*Hyperspace Analogue to Language*), que monta a matriz de

co-ocorrência utilizando k palavras à esquerda e k palavras à direita como contexto de w , e induzem a função de mapeamento f utilizando um método de redução de dimensionalidade que captura apenas dimensões com maior variância (LUND; BURGESS, 1996).

Todavia, uma técnica recente dessa categoria é conhecida como GloVe (PENNINGTON; SOCHER; MANNING, 2014), e como na técnica HAL, ela consiste em montar a matriz M de acordo com uma janela de palavras vizinhas, com tamanho predefinido d . A diferença está no modo em como essa matriz é populada, em que não é realizada uma contagem, mas sim é populada com a probabilidade conjunta das palavras w_i e w_j de acordo com a equação:

$$P(w_i, w_j) = \log(M_{i,j}) = \mathbf{M}_i^\top \mathbf{M}_j + b_i + b_j \quad (2.17)$$

Onde cada linha M_w é inicializada aleatoriamente e ajustada com uma função de custo associada à $P(w_i, w_j)$ usando gradiente descendente.

3. **Representações distribuídas:** são conhecidas na literatura simplesmente como *word embeddings*. O uso de *word embeddings* é uma abordagem recente que vem obtendo sucesso, as quais possuem uma representação baseada em vetores reais distribuídos em um espaço multidimensional induzidos através de aprendizado não-supervisionado (TURIAN; RATINOV; BENGIO, 2010). Cada dimensão da *embedding* de uma palavra representa uma *feature* latente, de modo a capturarem distribuidamente propriedades semânticas, sintáticas ou morfológicas dessa palavra (COLLOBERT *et al.*, 2011). O número de dimensões dos vetores pode variar, e em geral, melhores representações podem ser alcançadas com um maior número de dimensões. Mas se esse número for muito grande, o processamento de indução pode ser demorado.

Collobert e Weston (2008) usa um modelo de língua neural utilizando uma rede neural com uma camada oculta que tenta prever a próxima palavra dada k anteriores. A indução das representações é feita através do ajuste de pesos usando o algoritmo de retro-propagação (GOODFELLOW; BENGIO; COURVILLE, 2016), em que a primeira camada e a última camada possuem dimensões iguais de $\mathbb{R}^{|V| \times d}$, de modo que a rede retorne escores altos para janelas de palavras que ocorrem regularmente num *corpus* não etiquetado, e escores baixos para janelas com palavras inseridas aleatoriamente.

Uma técnica bastante utilizada e conhecida em PLN é o *word2vec* (MIKOLOV *et al.*, 2013b), a qual segue o mesmo princípio do modelo de língua natural de Collobert e Weston (2008), com a exceção de que não utiliza a camada oculta da rede, formando um modelo log-linear mais rápido computacionalmente. Essa técnica foi dividida em duas modelagens, sendo a primeira a *Continuous Bag-of-Words* (CBOW) mostrada na Figura 9a, que dada uma janela de palavras de entrada a rede tenta prever a palavra do meio na saída. Já o modelo Skip-gram, mostrado na Figura 9b, tenta prever as palavras da janela de contexto dada a palavra do meio como entrada.

Figura 9 – Modelos do *word2vec*.

Fonte: Adaptada de Mikolov *et al.* (2013b).

O *word2vec* não leva em consideração a ordem com que as palavras ocorrem dentro da janela, o que torna o processo menos sintático, já que ordem é um fenômeno importante para a sintaxe. Para lidar com isso, foi proposta uma modificação nos modelos do *word2vec* para que lidassem com ordem através da concatenação das entradas no modelo CBOW (ao invés da soma), e uma ponderação crescente para a saída do modelo Skip-gram. Essa técnica é conhecida na literatura como *wang2vec* (LING *et al.*, 2015) e os respectivos modelos são mostrados na Figura 10.

Figura 10 – Modelos do *wang2vec*.

Fonte: Adaptada de Ling *et al.* (2015).

Uma das técnicas de indução de *word embeddings* mais recente é o *fastText* (BOJANOWSKI *et al.*, 2016; JOULIN *et al.*, 2016), que consiste em usar n-gramas de caracteres de uma dada palavra na esperança de capturar informações morfológicas. Para fazer isso, foi adaptado o modelo Skip-gram do *word2vec*, de modo que a função de escore da saída da rede seja calculada baseando-se nos vetores dos n-gramas de caracteres, os quais são somados com os de vetores de contexto para representar uma palavra w_t . A Figura 11 mostra essa adaptação.

Nesta dissertação, usaremos os termos *embeddings*, vetores de palavras, *embeddings* de palavras

Figura 11 – Modelo de skip-gram do *fastText*.

Fonte: Adaptada de [Bojanowski et al. \(2016\)](#).

e *word embeddings* indistintamente. Além disso, atribuímos esses termos não somente para representações distribuídas, mas também para representações distribucionais e baseadas em *clusters*.

Portanto, para a obtenção de uma *embedding* para cada palavra no vocabulário $\mathcal{V}$ há a transformação f de uma palavra $w \in \mathcal{V}$ para um espaço real $\mathcal{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$ de *embeddings*:

$$f : \mathcal{V} \mapsto \mathcal{W} \quad (2.18)$$

Como mencionado, as *embeddings* conseguem capturar similaridades entre as palavras (o tipo de similaridade varia com o método usado para gerá-las) de acordo com sua distância euclidiana, ou outro tipo de cálculo de similaridade entre vetores (por exemplo, similaridade cosseno ou similaridade de Jaccard).

2.5 Representação Prosódica

Informações prosódicas são amplamente usadas na literatura para as tarefas de SS e DD. As unidades de sentenças linguísticas são crucialmente delimitadas por evidências como angulação de energia, contorno de núcleo e pausas ([VASSIÈRE, 1983](#)). E as subestruturas de disfluências de edição podem apresentar propriedades acústicas que ajudam a distinguí-las entre si ([HINDLE, 1983](#)).

A terminologia usada na literatura entre informações prosódicas e acústicas é confusa, por exemplo, [Huang et al. \(2014\)](#) faz uma categorização clara entre *features* prosódicas e acústicas, já [Ferguson, Durrett e Klein \(2015\)](#) usa o termo prosódico para ambas categorias de *features*. Nesta dissertação definimos esses termos da seguinte maneira:

- **Informações acústicas:** São informações auditivas provindas do áudio que não estão relacionadas necessariamente com estruturas linguísticas. Por exemplo: informações de

mudança de interlocutor; gênero do interlocutor⁹; qualidade do sinal; frequência de modularização.

- **Informações prosódicas:** São informações acústicas que estão relacionadas com estruturas linguísticas independentemente extraídas de supra-segmentos¹⁰. Por exemplo: altura do som; intensidade; duração.

Optamos por focar apenas em informações prosódicas, visto que elas são amplamente usadas na literatura de SS e DD (SHRIBERG *et al.*, 2000; LIU *et al.*, 2006; MONIZ, 2013). As informações prosódicas podem ser extraídas em diferentes níveis, alinhando as informações com os espaços entre palavras ou com cada fonema das palavras transcritas de forma manual ou automática. Entre essas informações, destacam-se na literatura a utilização de *features* de pausas, duração, *pitch* e energia. Especificamos nas subseções seguintes cada uma dessas informações.

2.5.1 Duração e Pausa

A duração está relacionada à sustentação sonora de um fonema (DÜBOIS *et al.*, 1973). Já pausa é constituída por um silêncio no fluxo locutivo (SEARA, 2013). Para textos escritos, as pausas são associadas com sinais de pontuação, por exemplo, as vírgulas geralmente indicam uma pausa para o leitor conseguir respirar durante a leitura em voz alta. Ou seja, as pausas trabalham na construção de sentido, agindo sobre a sintaxe, a semântica e organizando o discurso (COTES, 2007).

Em geral, para as tarefas de SS e DD, calculam-se as durações dos fonemas de todas as palavras na transcrição, de modo a normalizá-las para cada fonema, seja ele vogal, consoante vozeada ou consoante não vozeada:

$$dur_p = \frac{dur_p - \mu_p}{\sigma_p} \quad (2.19)$$

Onde μ_p e σ_p são a média e o desvio padrão para o fonema p calculados sobre todo o conjunto de dados. As pausas são identificadas por silêncio entre palavras, de modo a calcular suas durações. Em alguns casos, as pausas são categorizadas empiricamente de acordo com sua durações em pausas curtas ($dur \leq x$), pausas longas ($dur > x$), e sem pausa ($dur = 0$), mas em geral essa categorização é feita automaticamente pelo classificador prosódico utilizado (SHRIBERG *et al.*, 2000).

⁹ Isso não é necessariamente verdade, pois como mostrado por Vassière (1983), é possível descobrir o gênero do falante através do *pitch*.

¹⁰ Segundo Jurafsky e Martin (2014), o termo supra-segmentos significa além dos níveis de segmentos e fonemas.

2.5.2 Pitch

O *pitch* está relacionado ao acento de entoação (de altura), referente à frequência de f_0 (DÜBOIS *et al.*, 1973), ou seja, nos faz perceber o quão baixo ou alto um som soa a um ouvinte. A Figura 12 mostra o contorno de *pitch* para a sentença “Renato leu o livro” no modo declarativo (à esquerda) e interrogativo (à direita).

Figura 12 – Curva de contorno de *pitch* de uma sentença no modo declarativa e interrogativa.

Fonte: Seara (2013).

A informação de *pitch* está bastante correlacionada com o fim de enunciados (MONIZ, 2013). É possível ver, por exemplo, na Figura 12 que usando o contorno de *pitch* é ainda possível distinguir entre uma declaração (ascendente no início e cai ao final) e uma pergunta (elevado no final da sentença e então se torna descendente).

O contorno de *pitch* pode ser obtido analisando uma série de extrações de *pitch* de fonemas. Com esse contorno é possível obter outras *features*, como: informação de limite (mínimo, máximo, média); e informação de inclinação (angulação do contorno).

2.5.3 Energia

A energia (ou intensidade) está relacionada à força com que o ar é expelido dos pulmões (DÜBOIS *et al.*, 1973), ou seja, tem relação com a amplitude de um som.

Informações de energia podem ser processadas da mesma maneira que de *pitch*, isto é, podemos extrair informações de limites e de inclinação. Por fim, tanto o *pitch* como a energia variam de pessoa para pessoa e, portanto, é importante normalizá-los para cada falante antes de usá-los para distinguir fenômenos linguísticos (VASSIÈRE, 1983).

2.6 Redes Neurais Artificiais

Apesar da complexidade do cérebro humano, a origem de redes neurais artificiais (ou simplesmente Redes Neurais - RN) tem como inspiração o processo de funcionamento de neurônios naturais (RUMELHART *et al.*, 1986). Redes neurais artificiais são, na verdade, modelos de aprendizado de máquina que consistem numa série de funções paramétricas encadeadas aplicadas sobre uma entrada de dados, conhecidas como camadas.

Elas já foram aplicadas com sucesso em uma grande variedade de tarefas de PLN, incluindo análise sintática (FONSECA; ALUISIO *et al.*, 2015b), reconhecimento de voz (GRAVES; MOHAMED; HINTON, 2013), sumarização (CHENG; LAPATA, 2016) e tradução automática (DEVLIN *et al.*, 2014), entre outros. O sucesso recente está relacionado com a utilização de computadores mais rápidos e paralelos, com o desenvolvimento de novas ideias de aprendizado de máquina sobre esparsidade, regularização, e otimização, e também devido à disponibilidade de grandes conjuntos de dados, principalmente devido a era de “*big data*” da atualidade (SOCHER, 2014).

Devido a essas características, redes neurais foram se tornando mais profundas ao longo do tempo ao incluir mais camadas ocultas em suas arquiteturas, fazendo com que o termo *deep learning* fosse cunhado na literatura para designar um novo conjunto de técnicas e algoritmos. Entretanto, *deep learning* apenas “parece” ser uma nova ideia, quando de fato data da década de 40. Isso aconteceu porque sua popularidade decresceu e cresceu através dos anos usando diferentes nomes, e só recentemente a literatura passou a usar frequentemente o termo, refletindo a influência de diferentes pesquisadores e perspectivas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Primeiramente, apresentaremos noções básicas e a terminologia de redes neurais, para depois então generalizar sobre arquiteturas mais complexas, como redes neurais recorrentes e redes neurais convolucionais.

2.6.1 Multilayer Perceptron

Redes neurais multi camadas são conhecidas como MLPs (ou *Multilayer Perceptrons*), e consistem de uma sequência de camadas compostas por unidades conhecidas como neurônios, os quais são responsáveis por calcular funções de ativação sobre os dados de entrada. A Figura 13 mostra uma arquitetura simples de uma MLP com apenas uma camada oculta.

A primeira camada de uma MLP é chamada de “camada de entrada” e a camada final de “camada de saída”, que é responsável por retornar o valor final computado sobre a hipótese $h_{\theta}(x)$. As camadas intermediárias entre a entrada e saída são chamadas de “camadas ocultas”. Os nodos da j -ésima camada oculta são nomeados de $a_0^{(j)}, a_1^{(j)}, \dots, a_n^{(j)}$ e conhecidos como unidades de ativação (ou neurônios). O primeiro neurônio da camada de entrada x_0 e os primeiros neurônios das camadas ocultas $a_0^{(j)}$ são conhecidos como unidade viés (do inglês *bias unit*), e sempre

Figura 13 – Arquitetura de uma MLP com apenas uma camada oculta.

Fonte: Elaborada pelo autor.

produzem um valor igual a 1 como ativação. De fato, a unidade viés está presente em todas as arquiteturas de RNs que descrevemos nesta seção.

O funcionamento do fluxo de uma rede neural começa com a entrada da matriz de dados $X \in \mathbb{R}^{m \times n}$, em que m é o número de instâncias de treinamento e n é a quantidade de *features* associada a cada instância. As entradas são passadas para a próxima camada, onde cada neurônio a_i aprende uma informação associada a cada entrada anterior $\mathbf{x} \in \mathbf{X}$, de modo a produzir uma ativação, de acordo com a [Equação 2.20](#).

$$\mathbf{a} = f(\mathbf{z}) = f(\boldsymbol{\theta}^\top \mathbf{x}) \quad (2.20)$$

Onde f é a função de ativação, e comumente é uma função não-linear aplicada elementarmente sobre os dados, e $\boldsymbol{\theta}$ são os pesos associados entre as duas camadas. Exemplos comuns de funções de ativação são a função sigmóide ([Equação 2.21](#)), tangente hiperbólica ([Equação 2.22](#)), e ReLU ([Equação 2.23](#)).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.21)$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.22)$$

$$\text{relu}(z) = \max(0, z) \quad (2.23)$$

A hipótese final do modelo $h_\theta(X)$ é calculada através dos passos sucessivos da [Equação 2.20](#) até chegar na camada de saída. Esses passos podem ser generalizados de acordo com a

[Equação 2.24](#) para cada camada $\ell \in \{1, \dots, L\}$ de modo a realizar predições sobre a entrada X .

$$h_{\theta}(X) = \mathbf{a}^{(\ell)} = f\left(\theta^{(\ell)} \mathbf{a}^{(\ell-1)}\right) \quad (2.24)$$

Onde o vetor $\mathbf{a}^{(1)}$ é definido como sendo camada de entrada X , e $\theta^{(\ell)}$ é a matriz de pesos conectando a camada $\ell - 1$ com a camada ℓ , obtendo uma matriz real com dimensão $n_{\ell} \times (n_{\ell-1} + 1)$, sendo que n_{ℓ} representa o número de unidades de ativação na ℓ -ésima camada.

O número de unidades de ativação da última camada da rede $y = h_{\theta}(X) = \mathbf{a}^{(L)}$ define a quantidade de etiquetas que o modelo classifica. Por exemplo, num problema de classificação binária $y \in \{\mathcal{Y}_1, \mathcal{Y}_2\}$, a camada de saída possui apenas um neurônio com a função de ativação sigmóide. De modo que a probabilidade da instância $\mathbf{x}$ pertencer a classe $\mathcal{Y}_1$ ou não é:

$$\begin{aligned} P(\mathcal{Y}_1 | \mathbf{x}) &= h_{\theta}(\mathbf{x}) \\ P(\mathcal{Y}_2 | \mathbf{x}) &= 1 - P(\mathcal{Y}_1 | \mathbf{x}) \end{aligned}$$

Já em um problema multi-classe $y \in \{\mathcal{Y}_1, \dots, \mathcal{Y}_K\}$ é possível generalizar ao fazer com que a camada de saída possua exatamente K unidades de ativação. E a probabilidade de uma instância $\mathbf{x}$ ser da classe $\mathcal{Y}_k$ pode ser calculada com uma função *softmax*, descrita na [Equação 2.25](#).

$$P(\mathcal{Y}_k | \mathbf{x}) = h_{\theta}(\mathbf{x})_k = \frac{\exp(y_k)}{\sum_{j=1}^K \exp(y_j)} \quad (2.25)$$

Ao adicionar todas as camadas ocultas nas redes neurais com funções de ativação não-lineares, permite-se a produção de hipóteses não lineares mais complexas. Ao chegar na hipótese final, conclui-se os passos conhecidos na literatura como *Forward Propagation*.

Para encontrar o conjunto de pesos θ ideais, é necessário definir uma função de custo que mensura o quão errado está uma predição $\hat{y} = h_{\theta}(\mathbf{x})$ da rede. Dessa maneira é possível aplicar métodos de gradiente descendente para minimizar a função de custo em relação aos pesos. Mas para que isso seja feito com estabilidade, é necessário também que a função de custo seja convexa, para que os gradientes não divirjam. Entre essas funções uma que é muito utilizada é a entropia cruzada, que é por definição uma função convexa, descrita na [Equação 2.26](#).

$$\mathcal{L}(\theta) = - \sum_{k=1}^K y_k \log(\hat{y}_k) \quad (2.26)$$

Para minimizar a função de custo, o algoritmo *Backpropagation* ([WERBOS, 1974](#)) é amplamente utilizado, já que ele reusa partes da computação dos gradientes de maneira eficaz. O algoritmo segue a direção inversa da usada para obter a saída $h(x)$ para computar os gradientes de atualização dos pesos θ em relação a função de custo $\theta \mapsto \mathcal{L}$. Esse comportamento é conhecido como gradiente descendente ([CAUCHY, 1847](#)). A regra de atualização dos pesos num instante t é dada pela [Equação 2.27](#).

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla \mathcal{L}(\theta_t) \quad (2.27)$$

Onde η é a taxa de aprendizado usada para regular as passadas dos gradientes, e consequentemente tentar controlar a convergência da rede.

2.6.2 Redes Neurais Recorrentes

Redes Neurais Recorrentes (RNNs) são similares a MLP em sua arquitetura, com a adição da possibilidade de conexões recorrentes entre camadas. Muitas variedades de RNNs já foram propostas na literatura, mas focaremos apenas num modelo mais simples, que contém apenas uma camada oculta conectada a si mesmo, conforme ilustração da Figura 14.

Figura 14 – Arquitetura de uma rede neural com uma camada recorrente.

Fonte: Elaborada pelo autor.

Apesar da adição de recorrência na camada oculta parecer trivial, as implicações de aprendizagem sobre sequência de dados são benéficas, isso porque elas podem, em princípio, se basear em informações passadas para gerar uma predição. Em outras palavras, elas são capazes de capturar relações temporais entre uma sequência de dados. Além disso, a sequência de dados pode ter um tamanho variável, simplificando a modelagem de unidades linguísticas como sentenças. A RNN modela o aprendizado para aceitar sequências de tamanho variáveis ao compartilhar os parâmetros entre sucessivos instantes de tempo, ideia apresentada primeiramente por Lang, Waibel e Hinton (1990) e chamada de *time delay neural networks*.

Para simplificar a fundamentação de RNNs, vamos utilizar, praticamente, a mesma terminologia que usada para MLPs, com a exceção de que cada instância da entrada de dados é referida como um elemento temporal. Mais especificamente, a notação mais precisa consiste em relacionar cada elemento a um instante de tempo t , de modo que $0 \leq t < |\mathbf{X}_i|$. Sendo $\mathbf{X}$ a matriz de dados de entrada variavelmente tridimensional, isto é $\mathbf{X} \in \mathbb{R}^{m \times \varphi \times n}$, onde m é o número de instâncias de treinamento, n é o número de *features* e φ é uma variável que corresponde ao tamanho da sequência da i -ésima instância. Ou seja, na Figura 14, a camada de entrada foi ilustrada como um instante único qualquer para facilitar a visualização.

Dessa forma, podemos estender a Equação 2.24 para generalizar os passos da predição

em um instante $\mathbf{u}_t = \mathbf{a}_t$ na ℓ -ésima camada de acordo com [Equação 2.28](#).

$$h_\theta(X) = \mathbf{u}_t^{(\ell)} = f\left(\theta^{(\ell)} \mathbf{u}_t^{(\ell-1)}\right) \quad (2.28)$$

Com a diferença de que quando a ℓ -ésima camada é recorrente, o calculo dos neurônios $\mathbf{u}$ dependem de ativações passadas, conforme mostrado na [Equação 2.29](#).

$$\mathbf{u}_t^{(\ell)} = \begin{cases} r\left(\mathbf{a}_t^{(\ell)}, \mathbf{u}_t^{(\ell)}\right) & \text{se } \ell \text{ é recorrente} \\ \mathbf{a}_t^{(\ell)} & \text{caso contrário} \end{cases} \quad (2.29)$$

Tal que a função r corresponde à recorrência temporal definida na [Equação 2.30](#).

$$r\left(\mathbf{a}_t^{(\ell)}, \mathbf{u}_t^{(\ell)}\right) = f\left(\theta_x^{(\ell)} \mathbf{a}_t^{(\ell)} + \theta_h^{(\ell)} \mathbf{u}_{t-1}^{(\ell)}\right) \quad (2.30)$$

Onde f é uma função de ativação não-linear (e.g. sigmoide), $\theta_x \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ é a matriz de pesos compartilhada para lidar com os elementos de entrada do instante t , e $\theta_h \in \mathbb{R}^{n_\ell \times n_\ell}$ é a matriz de pesos compartilhada para lidar com os estados ocultos da recorrência.

Para obter a saída final para i -ésima instância, basta chegar até o fim da recursão da [Equação 2.28](#) de modo em que cada t seja igual ao tamanho da sequência de dados ϕ . A camada de saída de uma rede recorrente também é temporal, de modo que a probabilidade $P(y|\mathbf{X}_i)$ é uma matriz com dimensão $\phi \times |\mathcal{Y}|$, e a etiqueta mais provável para cada elemento da sequência pode ser obtido com uma operação de *argmax* aplicada elementarmente na [Equação 2.25](#). Para minimizar a função de custo, os pesos da RNN podem ser ajustados usando o algoritmo *Backpropagation Through Time* (BTT) ([WERBOS, 1990](#)), que é uma variação do BP para realizar o ajuste ao longo do tempo, somando-se os gradientes da função de custo em cada instante t .

Algumas vezes é interessante saber informações de um contexto futuro para realizar uma predição. Por exemplo, para a tarefa de SS, uma dica lexical importante é como o falante inicia um enunciado, então é interessante apresentar uma *feature* de informação do contexto à direita (futuro) da palavra que precede um limite de sentença. Entretanto, RNN convencionais processam a sequência de dados na ordem temporal (da esquerda para direita). Uma solução elegante para esse problema foi proposta por [Schuster e Paliwal \(1997\)](#), que consiste, basicamente, em passar uma sequência de dados para uma RNN paralela que processa em ordem invertida (da direita para esquerda); essa RNN é independente de outra RNN que processa a entrada na ordem temporal. E para combinar os dois contextos, a saída de cada neurônio num instante $\mathbf{u}_t$ são combinadas posteriormente, usando por exemplo uma operação de soma:

$$\mathbf{u}_t^{(L)} = \overrightarrow{\mathbf{u}}_t^{(L)} + \overleftarrow{\mathbf{u}}_t^{(L)}$$

A rede neural que usa essa ideia de juntar contexto do passado e do futuro para realizar uma predição é denominada de Rede Neural Recorrente Bidirecional (BRNN), e é bastante utilizada na literatura.

Uma maneira útil de visualizar uma RNN qualquer é desenrolado o grafo formado pela recursão da Equação 2.29. Ilustramos essa visualização para um único neurônio recorrente unidirecional em conjunto com a sentença de entrada “quem foi que tirou o sapato dela” na Figura 15. Desse modo a rede pode ser vista como uma MLP com várias camadas ocultas, o que acaba condizendo com a definição do termo *deep neural networks*.

Figura 15 – Unidade recorrente desenrolada.

Fonte: Elaborada pelo autor.

Apesar das RNNs serem aplicadas em várias tarefas, a sua modelagem básica com a utilização do BTT implica em dois problemas de aprendizado (GOODFELLOW; BENGIO; COURVILLE, 2016): (i) desvanecimento do gradiente (*vanishing gradient*), em que conforme o gradiente é passado para trás com o objetivo de corrigir os pesos para as saídas posteriores, este se torna muito pequeno, perdendo a importância de relações com unidades muito anteriores. (ii) o outro problema é chamado de *exploding gradient*, e nesse caso o gradiente se torna um termo exponencial. O primeiro problema foi ilustrado na Figura 15 em que a saída de “sapato” e “dela” dependem das informações de entrada do primeiro instante de tempo $t = 0$, sendo que um possível desdobramento para esse problema é a utilização de células de memória especiais com unidades de ativação, como as *Long-Short Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997). Para o segundo problema uma solução simples é reescalar os gradientes quando eles passarem de um limiar definido (PASCANU; MIKOLOV; BENGIO, 2013).

Long-Short Term Memory (LSTM)

Embora há outras maneiras de tentar solucionar o problema de desvanecimento do gradiente (GRAVES; MOHAMED; HINTON, 2013). Nesta dissertação, favorecemos o uso de células recorrentes de LSTM, que foram originalmente propostas por Hochreiter e Schmidhuber (1997). Quando células desse tipo são empregadas como unidades de ativação de uma rede neural, estas ficam conhecidas como *LSTM Networks*.

Como muitas variantes de LSTM já foram propostas na literatura, focamos nesta dissertação no uso de uma arquitetura básica apresentada por Greff *et al.* (2016) sem algumas conexões chamadas de *peepholes*, para qual os autores mostraram que desempenha tão bem como outras variantes. A Figura 16 ilustra as conexões de uma unidade de LSTM.

Figura 16 – Unidade de LSTM básica.

Fonte: Elaborada pelo autor.

Conforme ilustrado na Figura 16, a arquitetura básica consiste de um conjunto de conexões recorrentes. Para usar uma célula desse tipo numa rede recorrente, basta trocarmos como é calculado o termo $\mathbf{u}_{t-1}$ da Equação 2.30, de modo que não é necessário redefinir a função r . As atualizações de acordo com as relações dos blocos da figura para um instante t são definidas como:

$$\begin{aligned}
 \mathbf{i}_t &= \sigma(\theta_{ai} \cdot \mathbf{a}_t + \theta_{ui} \cdot \mathbf{u}_{t-1} + \mathbf{b}_i) \\
 \mathbf{f}_t &= \sigma(\theta_{af} \cdot \mathbf{a}_t + \theta_{uf} \cdot \mathbf{u}_{t-1} + \mathbf{b}_f) \\
 \mathbf{o}_t &= \sigma(\theta_{ao} \cdot \mathbf{a}_t + \theta_{uo} \cdot \mathbf{u}_{t-1} + \mathbf{b}_o) \\
 \mathbf{g}_t &= \tanh(\theta_{ac} \cdot \mathbf{a}_t + \theta_{uc} \cdot \mathbf{u}_{t-1} + \mathbf{b}_c) \\
 \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
 \mathbf{u}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
 \end{aligned}$$

Onde $\mathbf{u}_t \in \mathbb{R}^\varphi$ é uma unidade recorrente oculta, $\mathbf{i}_t \in \mathbb{R}^\varphi$ é o *gate* de entrada, $\mathbf{f}_t \in \mathbb{R}^\varphi$ é o *gate* de esquecimento, $\mathbf{o}_t \in \mathbb{R}^\varphi$ é o *gate* de saída, $\mathbf{g}_t \in \mathbb{R}^\varphi$ é o *gate* de modulação da entrada, e $\mathbf{c}_t \in \mathbb{R}^\varphi$ é a célula de memória, que consiste da soma de células de memórias pela multiplicadas elementarmente ($\odot$) por $\mathbf{f}_t$ com estados ocultos anteriores modulados pelo *gate* de entrada $\mathbf{i}_t$. E as matrizes $\theta_{\alpha\beta}$ são pesos das conexões entre os *gates* α e β , e $\mathbf{b}_\alpha$ são vetores de viés para o *gate* α .

Ao contrário de uma RNN básica, a saída de uma unidade LSTM não depende apenas de estados anteriores, mas depende também de informações codificadas internamente em sua célula de memória através dos *gates*. Os *gates* definidos com funções sigmoide (σ) podem ser vistos como reguladores que deixam passar $\sigma(x)$ de informação adiante. Já os *gates* que usam funções $\tanh$, são análogos a funções de recompensa.

Gated Recurrent Unit (GRU)

Com a motivação de simplificar o funcionamento de uma célula de LSTM, [Cho et al. \(2014\)](#) propuseram uma unidade chamada *Gated Recurrent Unit* (GRU), na qual um único *gate* $\tilde{\mathbf{u}}_t$ é responsável por controlar simultaneamente o fator de esquecimento $\mathbf{f}_t$ e a decisão de atualizar a célula interna $\mathbf{c}_t$:

$$\begin{aligned}\mathbf{z}_t &= \sigma(\theta_{az} \cdot \mathbf{a}_t + \theta_{uz} \cdot \mathbf{u}_{t-1} + \mathbf{b}_z) \\ \mathbf{r}_t &= \sigma(\theta_{ar} \cdot \mathbf{a}_t + \theta_{ur} \cdot \mathbf{u}_{t-1} + \mathbf{b}_r) \\ \tilde{\mathbf{u}}_t &= \tanh(\theta_{au} \cdot \mathbf{a}_t + \theta_{uu} \cdot (\mathbf{r}_t \odot \mathbf{u}_{t-1}) + \mathbf{b}_u) \\ \mathbf{u}_t &= (1 - \mathbf{z}_t) \odot \mathbf{u}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{u}}_t\end{aligned}$$

Onde $\theta_{\alpha\beta}$ são pesos das conexões entre os *gates* α e β , e $\mathbf{b}_\alpha$ são vetores de viés para o *gate* $\alpha \in \mathbb{R}^\varphi$. Nessa formulação, a saída $\mathbf{u}_t$ corresponde à interpolação linear ponderada pelo *gate* de atualização $\mathbf{z}_t$ entre o que havia anteriormente armazenado na célula $\mathbf{u}_{t-1}$ e o atual candidato $\tilde{\mathbf{u}}_t$. O novo candidato corresponde a uma regulação do que havia anteriormente $\mathbf{r}_t \odot \mathbf{u}_{t-1}$ com o que há de novo como entrada $\mathbf{a}_t$.

Vale notar que uma unidade GRU tem uma matriz de pesos a menos do que uma unidade LSTM, deixando o treinamento via *Backpropagation* mais rápido.

2.6.3 Redes Neurais Convolucionais

Assim como as RNNs, as Redes Neurais Convolucionais (CNNs) foram bastante utilizadas para lidar com problemas de PLN ([COLLOBERT et al., 2011](#); [FONSECA](#); [ALUISIO et al., 2015b](#); [ZHANG](#); [ZHAO](#); [LECUN, 2015](#)). Entretanto, sua utilização primordial foi para tarefas de visão-computacional, principalmente para reconhecimentos de dígitos escritos manualmente ([LECUN](#); [BENGIO et al., 1995](#)).

As CNNs possuem camadas convolucionais que aprendem filtros sobre informações locais usando pesos compartilhados. As convoluções entre camadas podem variar em suas dimensões. Nesta fundamentação, vamos focar em convoluções que trabalham sobre uma dimensão (CNN 1D) e duas dimensões (CNN 2D). Focamos na convolução 1D pois ela é mais simples e pode ser ilustrada facilmente. A [Figura 17](#) mostra o processo de convolução 1D sobre a sentença “quem foi que tirou o sapato dela”, de modo a gerar uma predição para cada i -ésima palavra $x_i \in \mathbb{R}^n$.

Analogamente, uma CNN 2D aplicada a um texto formaria uma janela deslizante sobre os vetores de sucessivas palavras, de modo que a largura da janela pode não ser igual ao tamanho dos vetores das palavras. Isto é, uma janela com tamanho $w \times h$, onde w pode ser diferente de n . De fato, podemos ver a convolução 1D como um caso específico da CNN 2D em que a largura da janela deslizante é igual ao tamanho dos vetores das palavras $w = n$, deslizando somente no eixo vertical h .

Figura 17 – Arquitetura de uma rede neural com uma camada convolucional 1D.

Fonte: Elaborada pelo autor.

Antes de descrevermos os passos para calcular o *Forward propagation* da CNN 1D ilustrada, vamos destacar uma terminologia baseando-se em parâmetros básicos utilizados numa CNN. Assim como numa RNN, uma camada convolucional trabalha com três dimensões, de modo que temos novamente uma matriz de entrada $\mathbf{X} \in \mathbb{R}^{m \times \varphi \times n}$, onde φ é uma variável que corresponde ao tamanho da i -ésima sentença/linha da matriz. Apesar de ter vários parâmetros para construir uma CNN, destacamos apenas quatro deles:

- Tamanho do *stride* (s): informa quantos passos devem ser dados para conectar sucessivos nós da convolução. Por exemplo, na ilustração acima $s = 1$ pois há um f_i para cada x_i . Mas se por acaso s fosse igual a 2, teríamos apenas 5 nós $f_i = \{f_0, \dots, f_4\}$, sendo $f_1 = [x_1, x_2, x_3]$.
- Tamanho do filtro (w): diz quantos elementos são concatenados para aprender filtros; em outras palavras, é o tamanho da janela deslizante na dimensão temporal. Por exemplo, na ilustração acima $w = 3$, fazendo com que $f_3 = [x_2, x_3, x_4]$. Mas se $w = 5$, então $f_3 = [x_1, x_2, x_3, x_4, x_5]$.
- Tamanho do *padding* (p): diz quantos nós de preenchimento com um valor específico (geralmente zero) são colocados à direita e à esquerda da sentença. De maneira geral, se quisermos classificar exatamente φ elementos com $s = 1$, definimos $p = \lfloor w/2 \rfloor$.
- Número de filtros (n_f): informa quantos filtros devem ser aprendidos após concatenar w entradas, formando um vetor de tamanho $w \cdot n$. A saída de cada f_i é um vetor com dimensão n_f .

De modo geral, a quantidade de filtros f_i (quantidade de neurônios na camada convolucional) é igual a $q_f = (\varphi - w + 2p)/s + 1$. E cada unidade convolucional $f_i \in \{f_0, \dots, f_{q_f}\}$ tem uma matriz de pesos compartilhada $\theta \in \mathbb{R}^{w \cdot n \times n_f}$ associada a ele. Isso significa que todas as unidades f aprendem as mesmas *features*, porém em regiões diferentes da sentença.

Podemos representar o passo da camada convolucional $\ell = c$ com uma breve modificação da Equação 2.24, ao criar uma janela deslizante de tamanho w que se move um passo por vez, conforme Equação 2.31.

$$\mathbf{a}^{(\ell)} = f\left(\boldsymbol{\theta}^{(\ell)} \mathbf{a}_{(t-w+1):t}^{(\ell-1)}\right), \quad w \leq t \leq q_f \quad (2.31)$$

Na Figura 17 após a camada convolucional, temos outra camada bastante utilizada na literatura conhecida como *pooling*. Essa camada serve para dois propósitos; o primeiro é reduzir a dimensionalidade da rede para que a computação dos passos seja mais rápida, e a segunda é para dar outra perspectiva das *features* aprendidas para as camadas seguintes.

Essa perspectiva depende do tipo de operação de *pooling* utilizada. A operação mais conhecida é a *max-pooling*, que tira o máximo das *features* ao longo do eixo temporal usando o mesmo esquema de janela que a camada de convolução. A Equação 2.32 realiza a operação de máximo olhando uma região de w elementos.

$$\mathbf{a}^{(\ell)} = \max_{w \leq t \leq q_f} \left\{ \mathbf{a}_{(t-w+1):t}^{(\ell-1)} \right\} \quad (2.32)$$

A camada de *max-pooling* não possui pesos associados, já que é apenas uma operação. De certo modo, tirar o máximo ao longo do tempo diz para a próxima camada quais são as *features* mais impactantes para a classificação, mas não necessariamente a posição correta delas na entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

A última camada da rede convolucional da Figura 17 é uma camada totalmente conectada com cada região extraída via *max-pooling*. Como na camada convolucional, é possível usar *padding* na camada de *pooling* para garantir que o número de elementos extraídos seja igual ao tamanho da sequência de entrada. Com isso, há exatamente ϕ saídas, sendo possível utilizar a função *softmax* da Equação 2.25 para cada instante de tempo $0 \leq t < \phi$ em ordem de descobrir a classe $\mathcal{Y}_k$ mais provável para uma palavra x_i .

TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os principais trabalhos da literatura para as duas tarefas de pesquisa abordadas neste projeto de mestrado: Segmentação de Sentenças (SS) na [Seção 3.1](#) e Detecção de Disfluências (DD) na [Seção 3.2](#).

Um aspecto importante para trabalhos de SS e DD é o tipo da fala do *corpus* utilizado para a tarefa, seja ele para avaliação ou treinamento. Embora o problema de DD seja, em princípio, definido apenas para *corpus* de fala, o mesmo não acontece para SS. O conceito de detecção de limites de sentença pode ser aplicado para textos escritos, como textos jornalísticos e *reviews* de produtos na Internet (JR; KAESTNER, 2004; BALDWIN; JOSEPH, 2009; READ *et al.*, 2012; LÓPEZ; PARDO, 2015) para facilitar tanto a sua leitura por humanos como o uso posterior de ferramentas de PLN. Entretanto, devido ao cenário no qual este trabalho está inserido, optamos por relatar apenas trabalhos que tratam de *corpus* de fala.

As diferenças entre *corpus* de fala e *corpus* de língua escrita são notáveis, principalmente quando levamos em consideração a quantidade de disfluências e o tamanho médio das frases vindas da fala. A língua falada é também tipicamente menos organizada que a escrita, colocando o desafio de ultrapassar a lacuna entre material falado e escrito (BATISTA; MAMEDE, 2011). As unidades típicas de sentenças para esses dois tipos são diferentes; uma unidade na fala pode ser uma sentença bem formada, um enunciado, ou até mesmo um única palavra (LIU *et al.*, 2005a).

Com relação aos *corpus* de fala, há uma diferença quanto aos tipos de fala: preparada (atuada) ou espontânea. Essa diferença geralmente se reflete na distribuição de limites de sentença no *corpus*, fazendo com que *features* distintas sejam mais discriminativas para um tipo específico de fala (SHRIBERG *et al.*, 2000; LIU *et al.*, 2005b; LIU *et al.*, 2006). Por exemplo, é mostrado em Kolár, Liu e Shriberg (2009) que *features* prosódicas são mais importantes e úteis para segmentar transcrições de programas jornalísticos do que de programas de conversação. Uma das características principais da fala espontânea é o alto número de disfluências, um fenômeno

que traz problemas significativos para sistemas de RAV, diminuindo a inteligibilidade das transcrições (HUANG; ZWEIG, 2002).

Por esses motivos, decidimos agrupar os trabalhos da literatura de SS quanto ao tipo de fala, enfatizando três tipos de fala: (1) fala preparada; (2) fala espontânea; (3) fala comprometida. O último tipo é a fala usada no cenário deste trabalho de pesquisa. A fala comprometida é, além de espontânea, ruidosa. O ruído vem inerentemente do comprometimento neuropsicológico das pessoas que participam dos testes neuropsicológicos que incluem: (i) reconto de histórias largamente conhecidas (como Cinderela, Chapéuzinho Vermelho e Branca de Neve) com base em um livro de figuras ou (ii) reconto imediato de uma história contada por um avaliador e reconto tardio (após 30 minutos do reconto). Quando as pessoas participam desses testes, produzem: sentenças mal formadas sintaticamente; palavras pronunciadas de maneira errada (modificando a morfologia original); prosódia de baixa qualidade (devido a baixa voz dos falantes); grande quantidade e tipos de disfluências.

Como disfluências ocorrem raramente em fala preparada, não agrupamos os trabalhos da literatura de DD quanto ao tipo de fala. Conforme vimos na [Seção 2.2](#), é possível categorizar as disfluências em: (1) preenchimentos, que são uma ou mais palavras classificadas como pausas preenchidas, marcadores discursivos ou termos de edição explícitos; (2) disfluências de edição, nas quais trechos de um enunciado podem ser classificados como repetições, revisões ou recomeços. Desse modo, dividimos os trabalhos relacionados usando essa categorização, pois a modelagem e a avaliação dessas categorias são feitas de formas diferente, e em alguns casos até mesmo diferentes *corpus* são utilizados para cada uma.

É importante mencionar que não há trabalhos relacionados na língua portuguesa brasileira para ambas as tarefas. Portanto, para todos os trabalhos relacionados abaixo assumimos que os experimentos foram feitos para a língua inglesa, mas especificaremos a língua caso os experimentos tenham sido feitos para outra que não seja inglesa. Além disso, decidimos não informar os resultados nas descrições de alguns trabalhos pois cada trabalho usa uma métrica diferente ou um conjunto de teste diferente. Mas de qualquer modo, reportamos os resultados de forma tabulada em termos de F_1 como síntese dos trabalhos nos finais de cada seção, nas quais discutimos um pouco mais sobre o avanço das tarefas em cada cenário. É importante informar que encontramos trabalhos de fala comprometida apenas para SS, criando divisões específicas para cada um desses trabalhos pois são os que mais se aproximam do cenário no qual esta dissertação está incluída. Já para DD não encontramos trabalhos relacionados para fala comprometida.

3.1 Segmentação de Sentenças

A tarefa de segmentação de sentenças a partir de transcrições de fala vem ganhando destaque nas últimas décadas devido ao aumento do uso de *softwares* de Reconhecimento

Automático de Voz (RAV) que geram automaticamente um texto a partir de seu áudio.

Os primeiros trabalhos da área datam do fim da década de 90 (STOLCKE; SHRIBERG, 1996; STOLCKE *et al.*, 1998; SHRIBERG *et al.*, 2000). Seus experimentos eram sobre o *cópus* de transcrições de conversas de telefone chamado Switchboard (GODFREY; HOLLIMAN; MCDANIEL, 1992), disponibilizado e anotado pelo LDC (Linguistic Data Consortium). O Switchboard é transcrito em nível de palavra e possui transcrições feitas por humanos. Um detalhe importante desse *cópus* é que é composto por fala espontânea de várias pessoas diferentes, e em uma mesma conversa diferentes interlocutores pronunciam seus enunciados. Para fala preparada, os primeiros trabalhos focaram em dados de textos escritos simulados para textos de fala (sem pontuação, não sensitivo ao caso, etc.), e também em gravações transcritas de textos lidos (BEEFERMAN; BERGER; LAFFERTY, 1998; CHEN, 1999).

Um evento marcante na área foram as iniciativas do programa EARS (Effective, Affordable Reusable Speech-to-text) para lidar com transcrições de fala, no período de 2002 a 2005. O programa EARS foi patrocinado pela agência americana DARPA¹ com o objetivo de avançar o estado da arte em geração automática de fala (CHEN *et al.*, 2006). Nesse período, foi lançada uma série de avaliações pelo NIST conhecidas como NIST-RT, que tinha como objetivo prover meios para investigar e avaliar tecnologias de Reconhecimento Automático de Voz (RAV) e Extração de Metadados (EMD), promovendo uma integração entre eles (BATISTA; MAMEDE, 2011). A série de avaliações do projeto NIST-RT criaram conjuntos de dados de fala anotados em nível de palavra para Conversações Telefônicas (CTS), Programas de Notícias Jornalísticas (PNJ), e para reuniões (*meetings*) espontâneas. Mais informações sobre o projeto NIST-RT podem ser vistas na Subseção 2.3.6.

A tarefa (ou problema) de segmentação de sentenças está relacionada com a de recuperação de pontuação, em que os limites de sentenças como pontos finais, pontos de interrogação e pontos de exclamação são detectados e os tipos de pontuação para cada limite é distinguido. Um grande número de sinais de pontuação podem ser representados para textos falados, incluindo: ponto final; ponto de exclamação; ponto de interrogação; vírgula; dois pontos; ponto-e-vírgula; travessão e aspas. A maioria dessas pontuações ocorrem raramente quando comparamos com a ocorrência de palavras, e por isso são difíceis de prever ou avaliar (BATISTA; MAMEDE, 2011). Nesta seção, a maioria dos trabalhos que vamos apresentar e discutir tratam da predição de pontos finais, embora comentemos sobre trabalhos que fazem a predição de outros sinais de pontuação devido à relevância do trabalho na área.

Finalmente, conforme vimos na Subseção 2.1.2, a segmentação de sentenças pode ser representada como um problema de classificação binária, em que cada palavra no texto pode ser etiquetada como *B*, significando que há um limite de sentença após essa palavra, ou *NB* caso contrário. O problema foi abordado dessa forma em todos os trabalhos nesta seção, com exceção

¹ A DARPA (Defense Advanced Research Projects Agency, Agência de Projetos de Pesquisa Avançada de Defesa) foi criada em fevereiro de 1958 (como ARPA) por militares e pesquisadores americanos.

do trabalho de Favre *et al.* (2008), o qual será discutido em detalhes.

3.1.1 Fala Preparada

Um dos primeiros trabalhos que realizaram experimentos para segmentar fala preparada foi o de Beeferman, Berger e Lafferty (1998), no qual os autores propõem um método baseado em Modelo de Língua (ML) de trígama o qual possui duas estratégias de decodificação usando o algoritmo de Viterbi: sem e com punição de não prever uma vírgula na correta posição. O método foi avaliado no *cópus* normalizado do Wall Street Journal (WSJ) do Penn Treebank (MARCUS; MARCINKIEWICZ; SANTORINI, 1993), sendo que a normalização consistia em simulara a saída de um RAV tirando sinais de pontuação. Entretanto, não informaram se normalizaram a caixa (baixa/alta) das palavras, e nem se transformaram numerais para palavras. Isso pode facilitar o trabalho de um classificador, pois a capitalização de palavras pode ser um forte indicio de pontuação (GRAVANO; JANSCHKE; BACCHIANI, 2009). Por fim, os autores sugerem que o uso de *features* morfossintáticas ou informações de um árvore sintática provinda de um *parser* estatístico pode resolver os problemas de longa dependência.

É intuitivo que informações como pausas da fala podem ajudar a definir um limite de sentença. O primeiro trabalho a usar informações acústicas para recuperação de sinais de pontuação foi o de Chen (1999), no qual foram realizados experimentos em transcrições de gravações feitas por participantes que leram em voz alta uma carta, caracterizando um *cópus* de fala preparada. O objetivo deste trabalho não foi o de realizar comparações entre métodos, e sim apenas levantar questões relevantes para a área. Descobertas importantes do autor são a maneira de combinar o sistema de RAV com informações lexicais para recuperar pontuações, e também a evidenciação da correlação entre pausas e o local onde ocorre uma sentença. É bastante claro que tanto a avaliação como a métrica utilizada não foram justas: a avaliação ocorreu em um *cópus* composto por apenas três textos idênticos lexicalmente; já quanto à métrica, os próprios autores argumentam que a acurácia não é justa devido ao desbalanceamento das classes.

Gotoh e Renals (2000) perceberam a importância de evidências prosódicas mostradas por Shriberg *et al.* (2000), em especial para informações de duração de pausa. E então propuseram um método que combina informações lexicais e de duração de pausas obtidas a partir da saída de um RAV. O método deles desempenha o papel de segmentar sentenças em textos transcritos automaticamente a partir de gravações de Programas de Notícias Jornalísticas (PNJ). Com os resultados, os autores perceberam que o modelo de pausa sozinho supera o modelo de língua. Além de que a combinação do modelo lexical com o de duração de pausas resultou em melhorias. Christensen, Gotoh e Renals (2006) estendo o trabalho de Gotoh e Renals (2000) através da utilização de outras *features* prosódicas, como a duração de fonemas, de modo a lidar com a recuperação de três sinais de pontuação: ponto final, vírgula e ponto de interrogação. A recuperação foi feita sobre o cenário de transcrições automáticas de PNJs de um subconjunto do *cópus* Hub-4 de 1997 (STERN, 1997). Os resultados provindos da avaliação das MLPs

mostraram que a duração da pausa é muito mais relevante que a duração de fonemas, duração de vogais e valores de pitch para a classificação de pontos finais. Por fim, assim como em outros trabalhos, a junção de evidências lexicais com prosódicas retornou melhores resultados. Esses resultados são piores do que os encontrados por Kim e Woodland (2003), o qual foi avaliado sobre o mesmo corpus.

Assim como Chen (1999), o trabalho de Kim e Woodland (2003) tratou da recuperação de pontuação enquanto o sistema de RAV processa o áudio. Os autores expandiram um trabalho anterior deles (KIM; WOODLAND, 2001), no qual mostraram que recuperar pontuação ao mesmo tempo do reconhecimento de fala melhora os resultados tanto em termos de F_1 para segmentação quanto de WER para os mesmos corpus utilizados neste artigo. Os experimentos são realizados sobre o Hub-4 de 1997 e o Hub-4 de 1998 (STERN, 1997). Os autores observaram melhorias significativas na combinação dos modelos em termos de F_1 e SER , sendo que melhores resultados foram obtidos quando foi dado mais importância para o modelo de prosódia. E como esperado, os resultados para transcrições manuais foram melhores do que para os resultados providas de um sistema de RAV. Entretanto, ao contrário do trabalho de Kim e Woodland (2001), eles perceberam que segmentar o texto sem realizar o reconhecimento automático da fala retornou melhores resultados para a recuperação de pontuação. A maior novidade deste trabalho está na utilização de duas redes Multilayer Perceptron (MLP) com o objeto de qualificar especificamente quais *features* prosódicas conseguem prever um determinado sinal de pontuação com melhor desempenho.

Os métodos anteriores tem sido desenvolvidos para capturar evidências locais (baseando-se em palavras vizinhas), usando informações lexicais e prosódicas independentemente e em conjunto para encontrar limites de sentença. Favre *et al.* (2008) usam informações sintáticas globais a partir de um parser probabilístico, o qual, em princípio, consegue capturar longas dependências. E, embora não tenha sido o primeiro artigo a utilizar informações sintáticas para SS (ROARK *et al.*, 2006), foi o que apresentou melhores resultados. Eles realizaram experimentos sobre o corpus TDT-4, que é composto por transcrições automáticas de PNJs. A grande contribuição deste trabalho são as *features* propostas, as quais eles dividiram em:

- **Features locais:** ML de 4-grama; duração de pausas; duração de fonemas; pitch e energia normalizados por interlocutor.
- **Features sintáticas:** ML sintático baseado em uma gramática livre de contexto probabilística (GLCP) sendo que a probabilidade de uma árvore é igual ao produto das probabilidades das produções usadas para gerá-la (JOHNSON, 1998).
- **Features generativas:** São probabilidades obtidas usando um ML de Evento Oculto (*Hidden Event Language Model*) descrito por Zimmerman *et al.* (2006) para uma dada sequência de palavras.

O método usado por eles difere da usual modelagem usada para problemas de classificação, em que trata-se cada palavra como sendo predecessora de uma pontuação (B) ou não (NB). No caso deles, é dada uma sequência de palavras $w = w_1 \dots w_n$ e a saída do segmentador é uma sequência de índices $b = b_1 \dots b_k$, em que cada $b_i | 0 < i \leq n$ é o índice em que a i -ésima sentença termina ($b_1 = 1$ e $b_k = n$). Foram obtidos resultados estado-da-arte para a época, quando comparado com o método utilizado por (STOLCKE; SHRIBERG, 1996), em termos de F_1 para transcrições manuais e automáticas. Entretanto, obtiveram uma baixa diferença para os resultados que usaram *features* sintáticas ou não: apenas 1% tanto para as transcrições manuais como para as automáticas. Com os resultados, foi possível perceber que a maior contribuição vem de informações locais.

No trabalho de Batista *et al.* (2012) são recuperadas pontuações (ponto final, vírgula e ponto de interrogação) em transcrições manuais e automáticas de PNJs em inglês e português europeu. Este artigo sumariza uma série de artigos lançada pelos autores (BATISTA *et al.*, 2007; BATISTA *et al.*, 2008; BATISTA; TRANCOSO; MAMEDE, 2009; BATISTA *et al.*, 2010), e também a tese de doutorado do primeiro autor (BATISTA; MAMEDE, 2011). O método deles baseia-se na utilização de Regressão Logística (RL), o qual é primeiramente utilizado para recuperar pontos finais e vírgulas simultaneamente, e só depois é reutilizado para classificar pontos de interrogação levando em consideração as pontuações já recuperadas. Com base nos resultados foi possível perceber que a adição de texto escrito no treinamento aumentou apenas 1% em termos de F_1 para as transcrições manuais, mas que para transcrições automáticas acabou piorando o desempenho (dados em inglês) e mantendo o mesmo valor de F_1 para português. Essa descoberta faz surgir a reflexão sobre a necessidade de aumentar o uso de dados lexicais em contrapartida com o tempo de treinamento. Os autores informam que, com os resultados obtidos, eles perceberam que vírgulas são geralmente identificadas por dicas lexicais, enquanto que pontos finais dependem mais de dicas prosódicas. Eles ainda discutem que identificar pontos de interrogação é uma tarefa difícil para transcrições manuais e automáticas devido a particularidades das línguas e da fala, além da baixa representatividade de perguntas nos corpuses de fala.

O atual estado-da-arte para o corpus da avaliação NIST-RT 2004 é o trabalho de Huang *et al.* (2014), no qual são detectados limites de sentença em transcrições manuais e automáticas. O conjunto de *features* utilizado é extenso, e para lidar com elas foi decidido separá-las usando uma estratégia multi-visão (*multi-view learning*). No total, o conjunto consistia em 332 *features*: 54 acústicas; 108 articulatórias; 162 prosódicas; 8 linguísticas. O método é uma adaptação do proposto por Liu *et al.* (2006), mas ao invés de tratar as informações prosódicas usando uma Árvore de Decisão (AD), eles optaram por utilizar uma MLP. Eliminando as informações acústicas e articulatórias como entrada da rede, o método proposto é igual ao de Xu *et al.* (2014), no qual foram feitas avaliações para decidirem a quantidade de camadas e de neurônios na MLP. O método final consiste de uma junção da MLP com um CRF (*Conditional Random Fields*) (SUTTON; MCCALLUM *et al.*, 2012) que lida com as informações linguísticas em conjunto

com as predições da MLP. Os autores mostraram que todas as *features* usadas na estratégia multi-visão desempenharam um papel complementar na predição, e que a combinação entre todas elas resultou em um desempenho superior. Os autores chegam à conclusão de que uma rede neural consegue aprender melhor as relações prosódicas do que uma AD, e com isso conseguiram atingir melhores resultados do que os de Liu *et al.* (2006). Porém, essa afirmação não é razoável, pois deve-se analisar também o uso das informações acústicas e articulatórias na MLP. Além disso, a comparação com os resultados é suscetível a críticas, pois os autores não usaram o mesmo conjunto de teste, em que o número de palavras para o conjunto de teste é cerca de 4 vezes menor do que o descrito por Liu *et al.* (2006).

O trabalho de Che *et al.* (2016) foi um dos trabalhos em que usamos como inspiração para o método proposto nesta dissertação. Ele trata da recuperação de três sinais de pontuação (vírgulas, pontos finais e pontos de interrogação) em transcrições manuais e automáticas de falas de palestras. As palestras correspondem ao conjunto de dados TED Talks de 2011 e 2012, os quais foram distribuídos de maneira gratuita na Web e compilados como recurso de avaliação pelo IWSLT (International Workshop on Spoken Language Translation). Os autores decidiram focar apenas nas *features* lexicais, para as quais seguiram a ideia de trabalhos em outras tarefas de PLN em usar *word embeddings* (COLLOBERT *et al.*, 2011; MAAS *et al.*, 2011) como representação das palavras. Mais especificamente, foram usados vetores do GloVe (PENNINGTON; SOCHER; MANNING, 2014) para representar as palavras dentro de uma janela de palavras vizinhas.

Foram propostas três arquiteturas de redes neurais para recuperar as pontuações. A primeira é simples, e consiste numa MLP com 3 camadas. A segunda proposta, chamada de CNN-1 é uma rede neural convolucional (CNN) que consiste em retirar a primeira camada da MLP e colocar uma camada convolucional seguida pela operação de *max-pooling*. A CNN-1 se move apenas verticalmente dentro da janela de entrada, ou seja, trabalha em apenas uma dimensão. Desse modo, a integridade do vetor da palavra é mantido, conforme ilustrado na Figura 18a. E a terceira proposta, chamada de CNN-2, também é uma CNN com a mesma lógica da CNN-1, porém ela se move em duas dimensões dentro da janela (horizontalmente e verticalmente), desse modo diferentes valores de diferentes dimensões dos vetores das palavras podem ser combinados, conforme ilustrado na Figura 18b.

Figura 18 – Arquiteturas das camadas convolucionais de Che *et al.* (2016).

Fonte: Adaptada de Che *et al.* (2016).

Como a CNN-2 obteve melhores resultados que a CNN-1, os autores sugerem que para predição de sinais de pontuação, os valores escalares de cada posição dos vetores das palavras podem ser consideradas como *features* independentes. Outra informação interessante é que para esse cenário, a CNN conseguiu lidar melhor que a MLP. Além disso, os autores comentam que os modelos não conseguiram lidar bem com pontos de interrogação por dois motivos: (i) a frequência do sinal no corpus é baixa; (ii) o fato de usar uma janela de palavras de tamanho fixo não consegue lidar muito bem com casos em que as evidências lexicais para perguntas estão no início da sentença (*wh-questions*), caracterizando problemas de longa dependência.

3.1.2 Fala Espontânea

A fala espontânea, assim como a fala preparada, praticamente começou a ser estudada no final dos anos 90 por [Stolcke e Shriberg \(1996\)](#), [Stolcke et al. \(1998\)](#). [Stolcke e Shriberg \(1996\)](#) focaram na detecção de limites de sentença para transcrições manuais de fala espontânea a partir de Conversações Telefônicas (CTS) do corpus Switchboard-1. Percebendo a influência de informações prosódicas para DD por [Shriberg, Bates e Stolcke \(1997\)](#), os autores logo expandiram o seu trabalho em segmentação de sentenças para lidar com esse tipo de dado ([STOLCKE et al., 1998](#)), tarefa que será discutida na [Seção 3.2.](#) As diferenças do artigo de 1996 para o de 1998 em relação a SS são as *features* e o método utilizado, porém o corpus de avaliação permanece o mesmo. As *features* utilizadas por [Stolcke e Shriberg \(1996\)](#) foram divididas em: lexicais; morfossintáticas; e relativas a informação de mudança de interlocutor. Já as utilizadas por [Stolcke et al. \(1998\)](#) foram categorizadas em: lexicais, prosódicas e acústicas. Além disso, os autores decidiram testar a eficácia das informações de mudança de interlocutor quando combinadas com as *features* lexicais, e quando combinadas com as *features* prosódicas.

O método proposto consiste num ML simples de trigramas, que retorna a probabilidade de uma sequência de palavras preceder um ponto final • (*B*) ou não (*NB*), descobrindo a melhor sequência de classes usando o algoritmo de Viterbi. Já o método proposto por [Stolcke et al. \(1998\)](#) consiste na utilização do mesmo ML em combinação com as probabilidades retornadas por uma AD do estilo CART. Os resultados dos experimentos mostraram que para o trabalho de [Stolcke e Shriberg \(1996\)](#), o modelo que obteve melhor desempenho foi o que usou informações morfossintáticas em conjunto com lexicais. Os autores perceberam que a maior parte dos erros eram cometidos para segmentação de falsos começos, sugerindo que esse tipo de erro ocorre devido a características da anotação. Já no artigo de [Stolcke et al. \(1998\)](#) os resultados foram reportados em termos de acurácia, então não é possível comparar com os resultados anteriores. Entretanto, ao analisar essa métrica os autores perceberam que os melhores resultados foram obtidos quando as *features* de mudança de interlocutor são combinadas com informações lexicais. E usando essa associação de *features*, melhorias foram encontradas com a combinação entre o modelo lexical e o modelo prosódico usando interpolação linear, com uma diferença de aproximadamente 1% para as outras combinações.

A maior contribuição desses trabalhos vem do pioneirismo de se utilizar MLs de eventos ocultos para SS, uma ideia que vários trabalhos relacionados utilizaram (SHRIBERG *et al.*, 2000; MROZINSKI *et al.*, 2006; FAVRE *et al.*, 2008). Além disso, o trabalho de Stolcke *et al.* (1998) foi o primeiro a mostrar que utilização de informações prosódicas para SS resulta em bons resultados, tanto para fala preparada como espontânea.

Após analisarem os resultados do trabalho de Stolcke *et al.* (1998) que mostrou que informações prosódicas e acústicas são importantes para SS, Shriberg *et al.* (2000) realizaram diversos experimentos para obter um conjunto fixo de *features* prosódicas e acústicas importantes para fala espontânea de CTS e fala preparada de PNJ. Os experimentos foram feitos sobre transcrições manuais e automáticas do Switchboard-1 e do Hub-4 de 1997 para ambos tipos de fala, e destacaram a motivação para o uso de cada *feature* prosódica e acústica, em conjunto com o modo de extraí-las, computá-las e normalizá-las. O método utilizado neste trabalho é o mesmo que o utilizado em trabalhos anteriores, só que o ML de evento oculto utilizado é de 4-grama. Além disso, os autores propuseram uma estratégia para combinar as probabilidades do modelo lexical com o prosódico que consiste em utilizar um HMM sobre as sequências de probabilidades geradas pelos modelos como observações. Os resultados mostraram que o modelo prosódico desempenhou melhor que o modelo lexical para dados de PNJ. Já para dados de CTS o modelo lexical sozinho retornou resultados bem melhores que o prosódico. Assim como na maioria dos trabalhos relacionados, os melhores resultados foram obtidos realizando a junção do modelo lexical com o modelo prosódico. Nessa combinação de modelos, obtiveram melhores resultados para transcrições automáticas usando interpolação linear assim como no trabalho anterior (STOLCKE *et al.*, 1998), e para transcrições automáticas a estratégia de usar HMM foi mais efetiva. A maior contribuição deste trabalho foi a análise minuciosa sobre as ADs de prosódia geradas para a classificação de limites de sentenças em cada cópula. E em decorrência desses experimentos, os autores mostraram que informações de duração de pausa são importantes para ambos os tipos de fala, enquanto valores de *pitch* tem mais importância para fala preparada, e valores relativos ao ML lexical tem mais importância para fala espontânea.

E Liu *et al.* (2005a) segmenta sentenças de transcrições manuais e automáticas em inglês. A motivação principal do trabalho é a proposta de um método baseado em CRF, o qual até então não tinha sido aplicado para essa tarefa. Eles comparam o método CRF proposto com dois outros métodos amplamente usados na literatura: RL (LIU, 2004; HUANG; ZWEIG, 2002; BATISTA *et al.*, 2012) e HMM (SHRIBERG *et al.*, 2000; GOTOH; RENALS, 2000; KIM; WOODLAND, 2001; CHRISTENSEN; GOTOH; RENALS, 2006) modelados usando MLs de evento oculto. Embora um ML de evento oculto seja em síntese diferente de um HMM, no momento da predição o comportamento é o mesmo (STOLCKE *et al.*, 1998). Os cópula usados para a avaliação do método foram os conjuntos de treinamento, validação e teste originais da série NIST-RT de 2003.

Os autores usaram informações lexicais, informações prosódicas extraídas e classificadas usando AD (SHRIBERG *et al.*, 2000), etiquetas morfossintáticas, e mudança de interlocutor

como *features*. O CRF proposto neste trabalho é de cadeia linear, como mostrado na Figura 19.

Figura 19 – Modelo de CRF de cadeia linear proposto por Liu *et al.* (2005a).

Fonte: Adaptada de Liu *et al.* (2005a).

O CRF de cadeia linear apresentado consiste em estados que são eventos $E_i \in \{B, NB\}$, e de observações O_i que são as *features* utilizadas para a i -ésima palavra. A melhor sequência de classes $\hat{E}^*$ para uma sequência de entrada O é definida como:

$$P(\hat{E} | \hat{O}) = \frac{\exp\left(\sum_k \theta_k^\top f_k(\hat{E}, \hat{O})\right)}{Z_\theta(\hat{O})}$$

$$\hat{E}^* = \arg \max_{\hat{E}} P(\hat{E} | \hat{O})$$

sendo que f_k são funções que definem a aplicação dos eventos e das observações em conjunto. Já Z é um termo de normalização, para o qual somam-se todas as possíveis variações dos eventos:

$$Z_\theta(\hat{O}) = \sum_{\hat{E}_i} \exp\left(\sum_k \theta_k^\top f_k(\hat{E}_i, O)\right)$$

O vetor θ_k são pesos a serem aprendidos para cada k -ésima função f durante a etapa de treinamento, que pode ser realizada usando um método de minimização de função convexa como gradiente descendente. A decodificação para encontrar a sequência de evidências $\hat{E}^*$ mais provável é feita utilizando o algoritmo de Viterbi. A principal diferença entre CRF e HMM é: o CRF é um modelo discriminativo (maximiza probabilidade condicional) enquanto o HMM é um modelo generativo (maximiza probabilidade conjunta). Uma descoberta interessante foi que o método conseguiu encontrar melhores resultados para fala espontânea de CTS do que para fala preparada de PNJ, o contrário do que aconteceu no trabalho de Shriberg *et al.* (2000), mas ainda fica uma ressalva pois a métrica utilizada foi diferente.

Já que há mais *NB* do que *B* em um texto, os modelos prosódico e lexical precisam ser construídos de uma maneira eficiente para aprender a partir de uma distribuição desbalanceada dos dados. O trabalho de Liu *et al.* (2005b) explora técnicas para resolver esse problema usando abordagens de reamostragem (em inglês *resampling*) e *bagging* (BREIMAN, 1996) sobre os mesmos corpus (NIST RT-03) que os utilizados no artigo anterior (LIU *et al.*, 2005a). As *features*

utilizadas neste artigo também foram as mesmas do artigo anterior, com exceção que não houve adaptação para os métodos de RL e CRF, pois eles não foram testados, apenas uma combinação de informações lexicais e prosódicas usando HMM de acordo com Shriberg *et al.* (2000). E o método de classificação consiste justamente no uso de HMM para associação de *features* lexicais e prosódicas. Os autores realizaram um estudo piloto para decidir a melhor abordagem para tratar o desbalanceamento, e esse estudo mostrou que usar o conjunto de dados original retorna um melhor desempenho tanto para dados de CTS e PNJ em termos de F_1 . Apesar disso, quando analisa-se a área abaixo da curva como métrica (AUC), a abordagem de *bagging* supera o uso do conjunto de dados original.

As propostas de classificação deste trabalho para o estudo piloto são a principal contribuição para a área, pois foram propostos métodos para balanceamento da classificação *B* vs *NB*, visto que geralmente há muito mais *NB* do que *B* em textos. Os melhores resultados foram obtidos usando o esquema de *bagging* para dados de PNJ e o conjunto de dados originais para CTS. Assim como visto no artigo anterior, os resultados para CTS foram melhores que para PNJ. Assim como em outros trabalhos relacionados, os experimentos mostraram que dicas lexicais e prosódicas provem informações complementares e melhoram o desempenho do modelo quando usadas em conjunto. Os autores concluem que é importante levar em consideração o desbalanceamento dos dados quando trabalha-se com problemas que lidam com *corpus* de fala, pois usualmente eles acabam gerando muitas transcrições, deixando métodos de classificação robustos mais pesados computacionalmente.

O trabalho de Shriberg *et al.* (2000) é expandido em (SHRIBERG *et al.*, 2009), de modo a dividir o texto não apenas em sentenças, mas em trechos de diálogos: declarações, perguntas ou comentários não relacionados ao assunto (*backchannels*). Os diálogos são retirados de transcrições manuais de fala espontânea (Reuniões) e preparada (PNJ) na língua inglesa. Uma diferença para este novo estudo está na qualidade dos *corpus* utilizados. O Switchboard usado anteriormente tinha um grande valor de *WER*, deixando a comparação com o *corpus* de fala preparada um pouco injusta. Porém, nesse estudo são utilizados *corpus* mais atuais disponibilizados para tarefas de avaliação do NIST. Um dos *corpus* é de fala espontânea composto por conversas em reuniões (MRDA), dividido de acordo com o trabalho de Güz *et al.* (2007), e outro de fala preparada é composto por programas de notícias jornalísticas (subconjunto do *corpus* TDT-4). O classificador utilizado é baseado em *boosting*, conhecido como AdaBoost e implementado pela ferramenta BoosTexter (SCHAPIRE; SINGER, 2000); o mesmo classificador usado por Güz *et al.* (2007). Foi feita uma análise do uso de diferentes conjuntos de *features* que desempenham maior contribuição para classificação. Essa análise mostrou que as *features* de duração são menos úteis que as de pitch ou energia e que a informação de pausa é a evidência mais discriminativa. No geral, quando são combinadas as informações lexicais e prosódicas os resultados são melhores. Apesar de ser contra-intuitivo, as *features* lexicais e de duração de pausas apresentam melhores resultados para o *corpus* de reuniões; talvez pelo fato de ter mais falsos começos, marcadores de discursos e pronomes de primeira pessoas nesse estilo de diálogo.

[Khomitsevich et al. \(2015\)](#) propuseram um sistema para pontuação automática de fala transcrita manualmente e automaticamente por especialistas para o língua russa, em que o sistema recupera três sinais de pontuação: pontos finais, vírgulas e pontos de interrogação. Eles avaliaram o método usando três corpú: entrevistas em rádio e televisão (ENT), conversação telefônicas (CTS), e obras literárias lidas em voz alta (TFV). As *features* utilizadas pelo método foram divididas em lexicais e prosódicas. As lexicais consistiam da própria identidade da palavra, a etiqueta morfossintática, etiquetas gramaticais (ex: caso e gênero) e informações semânticas relacionadas a Entidades Nomeadas (ex: nome de pessoa ou localização). No total haviam 887 etiquetas gramaticais em conjunto com as semânticas. As prosódicas foram retiradas do intervalo de cada palavra, e consistiam em 60 *features* distribuídas em duração de pausa e de fonemas, pitch e energia. Eles usaram dois classificadores para diferentes *features*: um CRF de cadeia linear para informações lexicais; e uma Máquina de Vetores de Suporte (SVM - *Support Vector Machines*) para lidar com informações prosódicas. As probabilidades dos dois classificadores são combinadas usando RL para a detecção de pontos finais e vírgulas, e dadas essas pontuações, é feito mais uma passada para a detecção de pontos de interrogação, sendo que os classificadores são os mesmos mas treinados sobre dados e *features* diferentes. Com os resultados eles perceberam que, em geral, o modelo lexical superou o modelo prosódico, exceto para as transcrições de TFV, provavelmente porque o modelo prosódico foi treinado justamente sobre esse tipo de fala, o que deixou os resultados um pouco enviesados. Todavia, a combinação entre os dois modelos melhorou o F_1 da detecção de pontuação para fala de ENT e CTS. Além disso, foi possível ver diferenças nos resultados para esses dois tipos de fala: pontos finais foram mais bem detectados do que vírgulas em CTS, sendo que ocorreu o inverso para fala de ENT.

Este trabalho é interessante porque usa dois classificadores, CRF e SVM, que são bastante utilizados na literatura, para tratar as respectivas *features* prosódicas ([SHRIBERG et al., 2005](#); [LUENGO et al., 2005](#); [RAMÍREZ et al., 2006](#); [AVANÇO; BRUM; NUNES, 2016](#)) e lexicais ([SHA; PEREIRA, 2003](#); [LIU et al., 2006](#); [HUANG et al., 2014](#)) em diferentes tarefas de PLN. De certa forma, este artigo realizou uma avaliação de robustez, pois treinou os classificadores com dicas lexicais e prosódicas de obras literárias e testou em fala de CTS e ENT. Entretanto deve-se levar em consideração que o RL final usado para combinar os dois classificadores foi treinado sobre dados de CTS, fazendo com que robustez com a combinação entre os dois métodos não seja de fato verificada para esse tipo de fala.

Recentemente, [Tilk e Alumäe \(2015\)](#) propuseram um método baseado em RNN que possui uma célula especial capaz de aprender longas dependências, conhecida como LSTM (do inglês *Long short-term memory*) ([HOCHREITER; SCHMIDHUBER, 1997](#)). O objetivo é recuperar vírgulas e pontos finais de transcrições manuais e automáticas para o idioma estoniano.

O corpus utilizado para treinamento consistia de diferentes gêneros linguísticos, incluindo: conversações em programas de rádio (CPRS) e em ligações telefônicas (CTS); programas de notícias jornalísticas; palestras; e textos escritos de jornais, internet, revistas, entre outros,

totalizando aproximadamente 386M palavras e 145h de áudio. Para validação foram usadas 2h de CPRS e 4.4h de CTS, e para teste foram usados 2 de CPRS e 5.6h de CTS. Isso constitui uma avaliação em fala espontânea, mas treinando com dados misturados, sendo que textos escritos correspondem à maior parte do treinamento. As *features* utilizadas por eles foram simples, e o argumento apresentado para a escolha foi que o método deve ser transportável para outras línguas. Em suma, foi utilizada só a identidade da palavra codificada em um vetor *one-hot* como *feature* lexical, e só duração de pausa como *feature* prosódica.

O método consiste em realizar duas passadas sobre as transcrições, ilustradas na [Figura 20](#):

1. São aprendidas informações textuais sobre a palavra sendo vista num momento t , com o objetivo de prever o ponto final em cada momento. Para cada momento t , as informações são aprendidas através de uma rede neural com uma camada totalmente conectada (100 neurônios) seguida por uma camada recorrente com LSTM (100 neurônios), para depois fazer a predição usando uma camada final com ativação *softmax*. A camada recorrente é a responsável por aprender as informações com base nos vetores das palavras vizinhas.
2. Após realizar o treinamento em todo o corpus de treinamento, as informações textuais aprendidas com a LSTM são concatenadas com informações de duração de pausa em um vetor, que é passado para uma nova rede recorrente com LSTM (100 neurônios) seguida com uma camada final com ativação *softmax*, a qual é responsável por detectar qual sinal de pontuação é mais provável.

Figura 20 – Classificador de dois passos proposto por [Tilk e Alumäe \(2015\)](#).

Fonte: [Tilk e Alumäe \(2015\)](#).

Os autores comparam o método com um ML de 4-grama para tratar informações lexicais em conjunto com uma AD para duração de pausa, método amplamente utilizado na literatura para SS ([STOLCKE et al., 1998](#); [SHRIBERG et al., 2000](#); [KIM](#); [WOODLAND, 2003](#); [CHRISTENSEN](#); [GOTOH](#); [RENALS, 2006](#); [KOLÁR](#); [LIU](#); [SHRIBERG, 2009](#)). Quando são comparados os

métodos usando ambas as *features*, a RNN retorna melhores resultados tanto para as transcrições manuais quanto para as automáticas. A abordagem de ajustar uma rede neural usando apenas informações textuais é interessante, pois cria a possibilidade de usar para a fase de treinamento mais dados do que os disponíveis das transcrições. Os autores também foram os primeiros a utilizarem redes neurais recorrentes (especialmente com LSTM) para recuperar vírgulas e pontos finais. O uso desse tipo de classificador é atraente porque tem a capacidade de aprender evidências sobre observações dependentes a uma longa distância (SUTSKEVER; VINYALS; LE, 2014).

Tilk e Alumäe (2016) propuseram a restauração de vírgulas e pontos finais em dois passos, expandido o seu trabalho anterior (TILK; ALUMÄE, 2015), com o uso de redes neurais recorrentes bidirecionais (BRNN) para realizar também a predição de pontos de interrogação para a língua estoniana e inglesa. A recuperação é feita sobre transcrições manuais e automáticas para o mesmo corpus do trabalho anterior para o estoniano. Já para o inglês é utilizado um corpus de palestras, mais especificamente do TED *Talks* de 2012 para treinamento e uma versão de 2011 para teste; a mesmo corpus com a mesma divisão foi utilizado nos experimentos de Che *et al.* (2016). As *features* utilizadas foram as mesmas que no trabalho anterior, com exceção de que aqui eles representaram as identidades das palavras; foram usadas *word embeddings* inicializadas aleatoriamente. Para o inglês as *word embeddings* foram inicializadas com os mesmos vetores de Che *et al.* (2016), além de que não foram utilizadas informações prosódicas. Desse modo, os resultados podem ser comparados em nível de método com os de Che *et al.* (2016).

O método proposto também consiste de duas passadas, como anteriormente, mas difere na arquitetura da rede neural:

1. No primeiro passo são aprendidas informações textuais sobre as *embeddings* usando uma camada recorrente bidirecional (BRNN), a qual é composta por células de GRU (do inglês *Gated Recurrent Units*) (CHO *et al.*, 2015) em conjunto com um mecanismo de atenção (BAHDANAU; CHO; BENGIO, 2015). A saída dessa camada é passada para uma RNN que utiliza as informações de atenção para aprender relações de alinhamento que são passadas para uma camada de saída totalmente conectada com a ativação *softmax*.
2. Após as informações textuais serem aprendidas, elas são passadas em conjunto com informações de duração de pausa para uma nova rede, que é composta por uma camada de RNN com células de GRU e uma camada de saída com ativação *softmax*.

Os resultados do método proposto foram comparados com os resultados publicados em Che *et al.* (2016), obtendo melhores resultados quando são usados os mesmos vetores como representações das palavras. O novo método também foi melhor do que o proposto no artigo anterior. Os autores ainda verificam a contribuição individual da expansão do modelo, apontando que a visão bidirecional da rede recorrente é um fator importante para a detecção de pontos finais

e pontos de interrogação, entretanto o mecanismo de atenção teve um efeito muito mais leve na recuperação de pontuação.

3.1.3 Fala Comprometida

3.1.3.1 *Fraser et al. (2015)*

O trabalho que mais compartilha características com o proposto nesta dissertação e que não foi realizado anteriormente por nós é o de *Fraser et al. (2015)*. Nele é proposto um método para segmentar sentenças de transcrições manuais de fala comprometida e preparada para o inglês. A fala comprometida nesse caso também vem de pacientes com doenças neuropsicológicas.

O corpus composto por transcrições de fala comprometida contém 28 (11SD + 17PNFA) narrativas recontadas por pacientes com subtipos diferentes de afasia progressiva primária, uma síndrome caracterizada pela dissolução isolada e gradual da função da linguagem (*MESULAM et al., 2001*). Para avaliar o comportamento da fala não comprometida no mesmo cenário, foram transcritas manualmente 23 narrativas recontadas por pacientes saudáveis (Controle). Para obter as narrativas, era solicitado aos participantes contarem a história da Cinderela, a qual havia sido mostrada previamente em um livro contendo apenas figuras (*SAFFRAN; BERNDT; SCHWARTZ, 1989*). As transcrições foram segmentadas por um único pesquisador seguindo dicas sintáticas, semânticas e prosódicas.

Além disso, os autores realizaram experimentos com um subconjunto de histórias jornalísticas do corpus de PNJ TDT-4 (chamado aqui de TDT-4L), o qual resultou em cerca de 804K palavras. Para ambos os corpus, as transcrições foram transformadas para caixa baixa e sinais de pontuações foram removidos para simular a saída de um sistema de RAV. Para fazer uma comparação justa entre os dois corpus, foram selecionadas 20 programas do TDT-4 (chamado aqui de TDT-4S), sendo possível determinar as diferenças em relação ao tamanho ou gêneros dos corpus.

As *features* utilizadas neste trabalho foram bem simples e divididas em lexicais e prosódicas. As lexicais eram formadas pela própria palavra e pelas classes morfossintáticas da própria palavra e das palavras da esquerda e da direita da palavra alvo. Para obter as informações prosódicas, primeiramente foi realizado um processo de alinhamento forçado entre as transcrições manuais e o áudio, provendo transcrições em nível de fonemas. Usando essas informações, eles extraíram as seguintes *features*:

- **Pausas:** calculadas entre palavras, foram categorizadas de acordo com a sua duração em nenhuma pausa (0ms), pausa curta ($< 400ms$) e pausa longa ($\geq 400ms$);
- **Pitch, energia e duração:** calculadas para a última vogal da própria palavras e das suas duas vizinhas mais próximas. Foram binarizadas de acordo com a comparação se eram maior do que a respectiva média para o falante correspondente.

O método consistia no classificador CRF proposto por [Liu et al. \(2005a\)](#), sendo que foi atribuído o corte mínimo de frequência para as *features* como 20, para tentar amenizar o efeito de sobreajuste. Para avaliar o modelo no cópua de fala comprometida e o TDT-4S, foi realizado um esquema de validação cruzada deixando um exemplo para teste e o resto para treinamento, até que todos os exemplos passassem como teste pelo menos uma vez (conhecido como *leave-one-out CV*). Esse esquema foi realizado para exemplos de cada subtipo de afasia (SD e PNFA) e para exemplos de controle (CTL). Já no TDT-4L original foram selecionadas aleatoriamente 90% das histórias para treinamento e o resto para teste.

Os experimentos foram medidos em termos de F_1 , e os resultados correspondentes ao uso de todas as *features* podem ser vistos na [Tabela 4](#).

Tabela 4 – Resultados para SS por [Fraser et al. \(2015\)](#).

Cópus	Subtipo	F_1
FCOM	SD	0.43
	PNFA	0.47
	CTL	0.51
TDT-4	S	0.59
	L	0.61

Fonte: Adaptada de [Fraser et al. \(2015\)](#).

FCOM corresponde ao cópus composto por transcrições de fala comprometida.

Os resultados mostraram que as informações que mais ajudam na segmentação são as morfofossintáticas e prosódicas, mas que a combinação de todas as *features* retornou melhores resultados. Como esperado pelos autores, os valores dos resultados para os subtipos de afasia foram menores do que para os de controle. O resultado para o TDT-4S, que tem praticamente a mesma quantidade de dados, foi provavelmente melhor por dois motivos: a qualidade do áudio é melhor e o gênero linguístico contém mais dicas prosódicas para a segmentação.

O método também foi analisado extrinsecamente para verificar como o ruído inserido pelo segmentador automático influenciou métricas de complexidade sintática. E com essa análise foi verificada que algumas métricas são robustas à segmentação, enquanto outras não. Os pesquisadores que pretender usar futuras tarefas a serem inseridas no *pipeline* de PLN após a segmentação devem estar cientes dos efeitos impostos pela segmentação, principalmente se a tarefa for a distinção direta dos subtipos de afasia.

3.1.3.2 [Treviso, Shulby e Aluísio \(2017\)](#)

Neste artigo, os autores propõem um método automático para segmentação de sentenças para fala comprometida, contando com uma avaliação neuropsicológica baseada na análise do discurso narrativo. O método usa Redes Neurais Recorrentes Convolucionais (RCNN) que trata

independentemente informações prosódicas e textuais. O método atingiu o estado-da-arte para a segmentação de sentenças de fala comprometida.

Em um trabalho conjunto com a Faculdade de Medicina da USP e o Núcleo Interinstitucional de Linguística Computacional (NILC), foram realizados experimentos da tarefa de segmentação de sentenças, para o Português Brasileiro (PTBR), em transcrições manuais de narrativas de testes neuropsicológicos. Para utilizar informações prosódicas, foi aplicado alinhamento forçado entre a transcrição manual e o áudio da narrativa para o *cópus* da Cinderela e BALE, estória do Carrocho, apresentados na [Subseção 4.1.1](#) e [Subseção 4.1.2](#).

Na primeira tentativa de trabalhar com o *cópus* anotado manualmente, o alinhamento forçado retornou erros, pois em algumas transcrições manuais havia mais palavras que as pronunciadas no áudio, e em outras menos. Assim, as transcrições manuais foram reanotadas por um único pesquisador (autor deste trabalho) para permitir o sucesso do alinhamento forçado. Esse *cópus* foi chamado de Cinderela v1.

Os *cópus* utilizados na avaliação do método são praticamente os mesmos que os descritos na [Seção 4.1](#). Eles apresentam as mesmas estatísticas em relação ao número de palavras, número de sentenças e duração, assim como os dados relativos aos interlocutores. Mais especificamente, foram utilizados dois *cópus* de fala comprometida: Cinderela versão 1 e Bale, estória do Cachorro; e dois de fala preparada: Constituição L(arge) e S(mall). Replicamos as propriedades de cada *cópus* na [Tabela 5](#).

Tabela 5 – Estatísticas dos *cópus* utilizados por [Treviso, Shulby e Aluísio \(2017\)](#)

Cópus	Tipo de Fala	# Palavras	# Sentenças	# Textos	Duração
Cinderela v1	Comprometida	23807	1843	60	4h 11m
Constituição L	Preparada	63275	2698	357	7h 39m
Constituição S	Preparada	30521	1409	60	3h 43m
Cachorro	Comprometida	1096	166	10	0h 09m

Fonte: Elaborada pelo autor.

Como a quantidade de palavras dos *cópus* de fala comprometida é pequena, realizou-se um 5-fold CV para medir o desempenho do modelo, que foi focado em um *cópus* por vez entre Cinderela e Constituição. Ou seja, em um determinado ciclo o modelo é treinado em um pedaço de 80% de um *cópus* e testado nos outros 20% do mesmo *cópus*. Além disso, para o *cópus* da Cinderela, foi avaliado o modelo para um grupo de pacientes específico (Controle ou CCL), já que cada grupo apresenta características neuropsicológicas diferentes.

As *features* utilizadas foram categorizadas em:

- **Lexicais:** *embedding* de 50 dimensões para cada palavra treinada através de um modelo de língua neural ([FONSECA; ROSA; ALUÍSIO, 2015a](#));

- **Morfossintáticas:** vetor aleatório de 10 dimensões para a classe morfossintática retornada pelo etiquetador *nlpnet*²;
- **Prosódicas:** duração de pausa; angulação do *pitch*, energia e duração do primeiro, último, penúltimo e antepenúltimo fonema (retornando $4 \cdot 3 + 1 = 13$ *features*).

O método de segmentação consiste na combinação linear de um modelo de RCNN que lida com as *features* lexicais e morfossintáticas, e outro modelo de RCNN que lida apenas com as informações prosódicas. A combinação linear entre os dois modelos é igual a usada por [Shriberg et al. \(2000\)](#): usa-se um peso complementar entre os dois modelos. As RCNNs possuem a mesma arquitetura: a primeira camada é uma camada convolucional (CNN) que se move em apenas uma dimensão, seguida por uma operação de *max-pooling*; a segunda é uma camada recorrente (RNN), em que cada neurônio é uma célula LSTM; a última é uma camada totalmente conectada distribuída no tempo com a função de ativação *softmax*, retornando assim a probabilidade de uma palavra anteceder um ponto final. Mais detalhes sobre a arquitetura podem ser vistos na [Seção 4.2](#).

Os experimentos foram avaliados de maneira intrínseca e extrínseca usando a métrica F_1 e usando a RCNN como classificador. Os resultados da avaliação intrínseca para o corpus da Cinderela foram de 0.74 para pacientes de controle (CTL) e 0.70 para pacientes com a fala comprometida (CCL), já para a Constituição foram de 0.79 e 0.80, para o conjunto L e S, respectivamente. Na avaliação extrínseca, os melhores resultados foram obtidos quando treinado com o corpus da Cinderela e testado no do Cachorro, com os valores de F_1 de 0.66 para Controle e 0.64 para CCL.

Esse trabalho mostrou que é possível alcançar bons resultados com fala comprometida quando comparando-os com fala preparada, até mesmo quando é usado praticamente a mesma quantidade de dados para treinamento. Outra evidência interessante foi que o uso de *word embeddings*, sem as etiquetas morfossintáticas, conseguem retornar quase o mesmo resultado que usando-as, o que indica que as *embeddings* já carregam conhecimento morfossintático suficiente. Foi mostrado também que o método bateu o estado-da-arte ao superar o método usado por [Fraser et al. \(2015\)](#) com uma grande diferença para os corpus de fala comprometida e preparada (diferença absoluta de ~0.20 e ~0.30). Além disso, os resultados mostraram que o método consegue um bom desempenho quando a história dos testes neuropsicológicos é diferente da usada no treinamento.

² <nilc.icmc.usp.br/nlpnet/>

3.1.4 Síntese

Fala Preparada

A [Tabela 6](#) sumariza os trabalhos de fala preparada, trazendo propriedades de cada trabalho. A vasta maioria de trabalhos relacionados discutidos tratam de fala preparada para o idioma inglês (EN); no melhor de nosso conhecimento, temos apenas um para o russo (RU), português europeu (PT), e português brasileiro (BR).

Tabela 6 – Resumo das características dos trabalhos de fala preparada para SS.

Trabalhos	Córpus	Cenário	Método	Features					F_1	F_1 (RAV)
				L	P	M	A	S		
(CHEN, 1999)	Próprio	TFV - EN	HMM	x	x		x		-	-
(GOTOH; RENALS, 2000)	Próprio	PNJ - EN	ML	x	x				-	0.70
(SHRIBERG et al., 2000)	Hub-4 1997	PNJ - EN	ML + AD	x	x		x		-	-
(KIM; WOODLAND, 2003)	Próprio	PNJ - EN	ML + AD	x	x				0.78	0.48
(LIU et al., 2005a)	NIST RT-03	PNJ - EN	Votação	x	x	x	x		-	-
(LIU et al., 2005b)	NIST RT-03	PNJ - EN	ML + AD	x	x	x	x		-	-
(CHRISTENSEN; GOTOH; RENALS, 2006)	Hub-4 1997	PNJ - EN	ML + AD	x	x				-	0.70
(FAVRE et al., 2008)	TDT-4	PNJ - EN	ML + GLCP	x	x			x	0.80	0.67
(SHRIBERG et al., 2009)	TDT-4	PNJ - EN	AdaBoost	x	x		x		0.77	-
(BATISTA et al., 2012)	NIST RT-03	PNJ - EN	RL	x	x	x	x		0.77	0.67
	Próprio	PNJ - PT	RL	x	x	x	x		0.80	0.70
(HUANG et al., 2014)	NIST RT-04	PNJ - EN	MLP + CRF	x	x	x	x	x	0.84	0.65
(FRASER et al., 2015)	TDT-4	PNJ - EN	CRF	x	x	x			0.61	-
(KHOMITSEVICH et al., 2015)	Próprio	TFV - RU	CRF + SVM	x	x	x			0.71	0.66
(CHE et al., 2016)	TED-T 2012	PAL - EN	CNN	x					0.65	0.60
(TILK; ALUMÃE, 2016)	TED-T 2012	PAL - EN	RNN	x	x				0.73	0.71
(TREVISIO; SHULBY; ALUÍSIO, 2017)	Constituição	TFV - BR	RCNN	x	x	x			0.80	-

Fonte: Elaborada pelo autor.

O cenário foi dividido no tipo de fala seguido pelo idioma: Programas de Notícias Jornalísticas (PNJ); Texto Falado em Voz Alta (TFV); Palestras (PAL). As *features* foram categorizadas em: (L) lexicais; (P) prosódicas; (M) morfossintáticas; (S) sintáticas. Resultados cujo F_1 não foi reportado pelos autores foram deixados em branco.

Não incluímos os dois trabalhos pioneiros da área feitos por [Beeferman, Berger e Lafferty \(1998\)](#) porque além de usarem textos escritos normalizados para simular a fala, eles não apresentam resultados em termos de F_1 para o sinal de ponto final. Os artigos de [Shriberg et al. \(2000\)](#), [Liu et al. \(2005a\)](#) e [Liu et al. \(2005b\)](#) tiveram um grande impacto na área, sendo que o primeiro estudou o impacto de diferentes *features* prosódicas para SS, o segundo estudou diferentes classificadores combinados num esquema de votação (CRF, RL e HMM), e o terceiro estudou a tarefa no ponto de vista de um problema desbalanceado de aprendizado de máquina. Entretanto, os resultados desses trabalhos não foram publicados em termos de F_1 . Uma possível

explicação para os trabalhos não terem reportados os resultados usando a métrica F_1 é que, de 2001 até 2005, a série de avaliações do NIST propôs a métrica *NIST SU Error* (equivalente ao *SER*), portanto a maioria dos trabalhos da época eram derivados dessas avaliações e reportaram usando a métrica proposta nas avaliações.

Além disso, é necessário fazer uma nota ao trabalho de [Kim e Woodland \(2003\)](#) devido ao fato de eles reportarem as avaliações sem separar os sinais de pontuação e, portanto, o F_1 apresentado é a média para predição de vírgulas, pontos finais e pontos de interrogação.

Antes de analisar os trabalhos, é importante salientar que apesar de tabelar os resultados para os trabalhos relacionados, não é justo compará-los em termos de F_1 por três motivos: (i) apenas alguns trabalhos reportam os resultados usando essa métrica; (ii) os corpú de teste são diferentes; (iii) apesar de alguns casos os corpú de teste serem o mesmo, a comparação entre os resultados ainda seria injusta, pois geralmente os autores dividiram o conjunto de treino e teste de maneiras diferentes - sem realizar uma validação cruzada para avaliar o modelo.

Quando analisamos os corpú de teste dos trabalhos, podemos perceber que a maioria trata de PNJ, sendo esses obtidos por meios próprios, pelas série de avaliações NIST-RT, ou usando o corpú do projeto TDT-4. Os quatro últimos trabalhos analisados mudaram de cenário. [Che et al. \(2016\)](#) e [Tilk e Alumäe \(2016\)](#) reportaram resultados para o mesmo conjunto de teste do TED Talks de 2012, sendo possível compará-los diretamente. Os trabalhos de [Khomitsevich et al. \(2015\)](#) e [Treviso, Shulby e Aluísio \(2017\)](#) realizaram experimentos sobre dados de textos falados em voz alta, categorizando uma fala bem preparada e de alta qualidade de gravação.

Muitos adotaram o Modelo de Língua (ML) de eventos ocultos do trabalho de [Stolcke e Shriberg \(1996\)](#) para tratar as informações lexicais. Árvores de Decisão (AD) foram bastante usadas para trabalhos devido a importância das *features* prosódicas feitas por [Shriberg et al. \(2000\)](#). Já os trabalho mais recentes focaram no uso de CRFs, um método que [Liu et al. \(2005a\)](#) mostrou ser bastante eficaz em termos de *SER*, tanto para fala preparada de PNJ como para fala espontânea de CTS. Os últimos trabalhos seguiram a estratégia de usar Redes Neurais, sendo essas Convolucionais (CNN), Recorrentes (RNN) ou a concatenação das duas (RCNN).

Quando olhamos para as *features* no decorrer do tempo, percebemos que desde o início os autores perceberam a importância de se utilizar informações prosódicas, com exceção do trabalho de [Che et al. \(2016\)](#), que optou por utilizar apenas informações lexicais usando *word embeddings*. Além disso, podemos perceber que os últimos trabalhos utilizaram, além da prosódia, informações linguísticas de morfossintaxe, obtendo bons resultados.

O que podemos concluir com os resultados apresentados na tabela é que, para o cenário de fala preparada de PNJ apresentam-se bons resultados, com um F_1 consideravelmente alto (média de 0.78) e um *SER* baixo (média de 0.40) quando são usadas *features* prosódicas e lexicais combinadas para as transcrições manuais. Outra observação interessante é em relação a diferença das transcrições manuais com as geradas por RAV para os trabalhos que reportam

resultados para esses dois casos: o F_1 é em média cerca de 12% mais baixo para as geradas por sistemas de RAV. Porém, quando observamos os resultados a partir de 2014, que foi quando Huang *et al.* (2014) usou CRF e MLP combinadas como classificador, vemos em média um F_1 de 0.66 para transcrições de RAV, o que acarreta numa diferença de apenas 7%. Isso nos mostra que nos últimos anos a distância entre os dois tipos de transcrições de fala preparada vem diminuindo.

Fala Espontânea

A Tabela 7 resume os trabalhos apresentados na subseção de fala espontânea com suas principais propriedades. Assim como para fala preparada, a maioria dos trabalhos relacionados realizaram experimentos para o idioma inglês (EN), provavelmente devido ao esforço do programa EARS do DARPA, entretanto foi possível encontrar um trabalho para o russo (RU) e dois trabalhos recentes para o estoniano (ET).

Tabela 7 – Resumo das características dos trabalhos de fala espontânea para SS.

Trabalhos	Córpus	Cenário	Método	Features				F_1	F_1 (RAV)
				L	P	M	A		
(STOLCKE; SHRIBERG, 1996)	SWB-1	CTS - EN	ML	x		x	x	0.76	-
(SHRIBERG <i>et al.</i> , 2000)	SWB-1	CTS - EN	ML + AD	x	x		x	-	-
(LIU <i>et al.</i> , 2005a)	NIST RT-03	CTS - EN	Votação	x	x	x	x	-	-
(LIU <i>et al.</i> , 2005b)	NIST RT-03	CTS - EN	ML + AD	x	x	x	x	-	-
(GÜZ <i>et al.</i> , 2007)	MRDA	REU - EN	AdaBoost	x	x		x	0.71	-
(SHRIBERG <i>et al.</i> , 2009)	MRDA	REU - EN	AdaBoost	x	x		x	0.82	-
(KHOMITSEVICH <i>et al.</i> , 2015)	Próprio	ENT - RU	CRF + SVM	x	x	x		0.50	0.46
	Próprio	CTS - RU	CRF + SVM	x	x	x		0.62	0.58
(TILK; ALUMÄE, 2015)	Próprio	CTS - ET	RNN	x	x			0.72	0.58
(TILK; ALUMÄE, 2016)	Próprio	CTS - ET	RNN	x	x			0.76	0.60

Fonte: Elaborada pelo autor.

O cenário foi dividido no tipo de fala seguido pelo idioma: Conversas Telefônicas (CTS); Reuniões (REU); Entrevistas (ENT). As *features* foram categorizadas em: (L) lexicais; (P) prosódicas; (M) morfossintáticas; (A) acústicas. Resultados cujo F_1 não foi reportado pelos autores foram deixados em branco.

A maior parte dos autores focou no cenário de conversações telefônicas, mas teve alguns que lidaram com reuniões e entrevistas. Também é perceptível que há mais trabalhos para fala preparada do que para espontânea, isso deve-se principalmente por dois motivos: (i) trabalhar com fala preparada é mais fácil, geralmente não são necessárias informações para tratar a mudança de interlocutor no meio de uma conversa; (ii) é mais fácil encontrar ou construir um córpus para esse tipo de fala. Entretanto, quando olhamos para questões financeiras para não membros da LDC, o segundo motivo não é totalmente verdade. O córpus TDT-4³ tem um preço

³ <<https://catalog.ldc.upenn.edu/LDC2005S11>>

de cerca 17,000.00\$ mais caro que o MRDA⁴, apesar do MRDA ter uma duração razoável de reuniões (72h).

Os métodos variaram bastante para esse tipo de fala, mas devemos notar que muitos autores realizaram experimentos conjuntos para fala espontânea e preparada usando o mesmo método, o que traz um pouco de consistência em relação aos anos. Assim como para fala preparada, os primeiros métodos consistiam em usar métodos de ML de eventos ocultos em combinação com árvores de decisão, principalmente devido aos esforços positivos demonstrados por Stolcke e Shriberg (1996) e Shriberg *et al.* (2000). O AdaBoost, que é um *ensemble* de ADs simples, foi usado mais especificamente para o cenário de reuniões, conseguindo bons resultados para as transcrições manuais. Refletindo o que aconteceu também para fala preparada, os últimos trabalhos optaram pelo uso de CRFs e RNNs, provavelmente influenciados pelo bom desempenho de trabalhos de outras áreas de PLN.

Não conseguimos apontar um padrão entre as *features* utilizadas pelos trabalhos relacionados. Cada autor resolveu usar um conjunto de *features* diferentes para o cenário específico de seu trabalho. Por exemplo, Tilk e Alumäe (2015) resolveu não fazer o uso de informações morfo-sintáticas para que o seu método pudesse ser usado em qualquer língua, até mesmo para as que não possuem um etiquetador morfo-sintático disponível.

Os resultados em termos de F_1 também não pode ser diretamente comparados. Entretanto, eles são comparáveis dado o corpus e cenário, por exemplo, a comparação dos trabalhos que utilizaram o MRDA como corpus e os dois últimos trabalhos para o estoniano. Os trabalhos de Liu *et al.* (2005a) e Liu *et al.* (2005b) já foram comparados na Subseção 3.1.2 usando a métrica *SER*.

O que podemos concluir com os resultados é que, em média, o F_1 para fala espontânea está em cerca de 0.70, uma diferença de 5% mais baixa do que para fala preparada. Quando analisamos apenas os trabalhos que possuem resultados para saídas geradas por sistemas de RAV, a média da diferença entre os dois tipos de transcrições é de 10%, o que é 3% maior do que a média da diferença para fala preparada a partir de 2014. Em outras palavras, podemos dizer que estatisticamente a fala espontânea é mais difícil de ser segmentada, resultado também encontrado para informações prosódicas nos experimentos de Kolár, Liu e Shriberg (2009). Todavia, devemos considerar que o tamanho da amostragem para a fala espontânea é bem menor do que para fala preparada.

Fala Comprometida

A Tabela 8 sumariza os trabalhos apresentados na subseção de fala comprometida com suas principais propriedades. Apenas dois trabalhos compõem a tabela, um para o inglês e outro para o português brasileiro.

⁴ <<https://catalog.ldc.upenn.edu/LDC2004S02>>

Tabela 8 – Resumo das características dos trabalhos de fala comprometida para SS.

Trabalhos	Cópus		Cenário	Método	Features			F ₁
	Nome	Subtipo			L	P	M	
(FRASER <i>et al.</i> , 2015)	Próprio	SD	NAR - EN	CRF	x	x	x	0.43
	Próprio	PNFA	NAR - EN	CRF	x	x	x	0.47
	Próprio	CTL	NAR - EN	CRF	x	x	x	0.51
(TREVISIO; SHULBY; ALUÍSIO, 2017)	Cinderela v1	CCL	NAR - BR	RCNN	x	x	x	0.70
	Cinderela v1	CTL	NAR - BR	RCNN	x	x	x	0.74

Fonte: Elaborada pelo autor.

O cenário foi dividido no tipo de fala - Reconto de Narrativas (NAR) - seguido pelo idioma. As *features* foram categorizadas em: (L) lexicais; (P) prosódicas; (M) morfossintáticas. O subtipo depende de cada cópus utilizado.

O cópus utilizado por Fraser *et al.* (2015) também possui transcrições de fala comprometida, nas quais pacientes recontam uma narrativa. Além disso, a história da narrativa (Cinderela) é a mesma nos dois trabalhos. Entretanto, não é totalmente justo comparar os resultados para os subtipos diferentes de controle (CTL), isso porque cada um corresponde a um comprometimento neuropsicológico diferente. Além de que comparar os resultados diretamente para CTL também seria injusto, pois os conjuntos de dados diferem até mesmo na língua usada.

Para que fosse possível comparar os dois trabalhos, Treviso, Shulby e Aluísio (2017) replicaram o método de CRF usado por Fraser *et al.* (2015) e testaram no cópus da Cinderela v1. Quando são consideradas *word embeddings* como *features* para representação lexicais das palavras, a RCNN superou o CRF com uma margem de 19% para transcrições de CTL e 17% para transcrições de CCL.

3.2 Detecção de Disfluências

Assim como para segmentação de sentenças (SS), a tarefa de detecção de disfluências (DD) a partir de transcrições de fala também vem ganhando destaque devido a transcrições automáticas através de sistemas de RAV.

Os trabalhos de detecção de disfluências da fala começaram antes dos de segmentação de sentenças (SHRIBERG, 1994), influenciados pelo lançamento do cópus de fala espontânea de conversações telefônicas Switchboard (GODFREY; HOLLIMAN; MCDANIEL, 1992). Para cópus de fala preparada há menos trabalhos, visto que geralmente não encontram-se disfluências em enunciados planejados. Na verdade, a fala preparada contém menos (ou nenhuma) disfluências do que a fala espontânea. Isso reflete a natureza dos problemas aos quais os interlocutores encontram enquanto planejam um enunciado espontâneo. Por exemplo, em uma fala em que o interlocutor lê em voz alta um texto escrito, dificilmente haverá disfluências porque geralmente o próprio texto escrito não as contém. Todavia, em cenários de Programas de Notícias Jornalísticas (PNJ) de rádio ou televisão, os interlocutores podem realizar enunciados com disfluências, apesar

da fala ser preparada previamente através de *scripts* (LIU, 2004).

A série de avaliações RT do NIST também ajudou a área de detecção de disfluências a ganhar destaque, já que DD faz parte da grande área de extração de metadados (EMD) a partir de transcrições produzidas por sistemas de RAV. Mais informação sobre o projeto NIST-RT pode ser vista na [Subseção 2.3.6](#).

A área de DD é difícil de ser estudada por dois motivos. O primeiro é a falta de disponibilização de dados anotados, pois anotar diferentes tipos de disfluências é uma tarefa lenta e cansativa. O segundo é a falta de padronização em conjuntos de dados, métricas e estilos de anotação usados por pesquisadores, por exemplo, a palavra *repair* foi empregada por muito tempo como sinônimo de disfluência de edição (LIU, 2004), porém nas novas anotações do Switchboard ela é empregada como a região de disfluência correspondente ao enunciado corrigido (HOUGH; SCHLANGEN, 2015) e também como um tipo de disfluência de edição (WANG *et al.*, 2017). Essas discrepâncias fazem com que seja ainda mais difícil comparar trabalhos de DD do que de SS e assim propor avanços para a área.

O estilo de anotação dos corpús, geralmente, refletem a modelagem do problema. Muitos trabalhos tentam detectar o tipo específico de disfluência (problema multi-classe) descrita na anotação, enquanto outros tentam apenas detectar se uma palavra está numa região de disfluência ou não (problema binário). A maioria dos trabalhos que lidaram com tipos de disfluências anotaram os dados de acordo com o trabalho de Shriberg (1994): categorizando as disfluências em preenchimentos e disfluências de edição. Além disso, conforme visto por Moniz *et al.* (2014), os reparos realizados em disfluências de edição variam entre diferentes tipos de fala, sendo que embora a fala espontânea apresente mais disfluências, ela também fornece mais evidências prosódicas para a detecção dessas regiões. Portanto, dividimos os trabalhos relacionados em trabalhos que procuraram detectar preenchimentos e trabalhos que se propuseram a detectar disfluências de edição, não seguindo necessariamente a anotação proposta por Shriberg (1994).

3.2.1 Disfluências de Edição

Antes de trazer os trabalhos relacionados, lembramos que disfluências de edição, cuja definição segue o trabalho de Shriberg (1994), são divididas em: repetições (*repetitions*), revisões (*revisions*), recomeços (*restarts* ou *false stats*), e disfluências complexas, que constituem uma série das disfluências anteriores seguidas ou recursivas. É importante notar que a nomenclatura usada pelos trabalhos relacionados varia bastante, portanto focamos mais nos trabalhos que seguem a nomenclatura de Shriberg (1994):

$$[(\text{enunciado original}) * \langle \text{termo de edição} \rangle \text{ correção}]$$

Lembramos também que a avaliação da tarefa de detecção de disfluências de edição depende fortemente de como é modelado o problema. Uma parte dos trabalhos preocupam-se

em detectar o ponto de interrupção (*) e o começo do enunciado original. Outros detectam de forma sequencial todas as palavras que fazem parte do enunciado original, seja de forma binária (é ou não disfluência) ou de forma estruturada (início/meio/fim da região de disfluência). Além disso, a maioria dos trabalhos não se preocupa em remover preenchimentos antes da detecção de disfluências de edição, alegando que informações como o termo de edição explícito acabam ajudando lexicalmente a tarefa. Isso tudo nos mostra que a comparação de métodos deve ser feita com cuidado, ainda que seja sobre o mesmo *cópus* e com a mesma anotação.

O trabalho de [Bear, Dowding e Shriberg \(1992\)](#) propôs um método para detectar reparos⁵ baseado em duas etapas. A primeira etapa consiste em um algoritmo simples de casamento de padrões, o qual usa regras para recuperar enunciados candidatos de revisões. A segunda etapa consiste em aplicar informações sintáticas, semânticas, e análises acústicas para distinguir revisões verdadeiras de falsos positivos. As avaliações foram realizadas sobre 500 sentenças do *cópus* ATIS, o qual contém padrões lexicais mais estáticos do que o Switchboard e, portanto, abordagens de casamento de padrões conseguem ser mais aplicáveis. Os resultados obtidos na época não puderam ser comparados, mas a maior contribuição é a alegação dos autores de que informações acústicas podem ser bastante eficazes quando combinadas com outras fontes de informação, enfatizando diferenças acústicas entre revisões verdadeiras e falsos positivos.

Como informações sintáticas e semânticas são muitas vezes difíceis de serem obtidas, requerendo o processamento de *parsers* que geralmente foram treinados em textos escritos, o trabalho de [Heeman e Allen \(1994\)](#) apresenta um algoritmo que corrige reparos sem esses processamentos. O algoritmo determina a região que faz parte do enunciado original ao construir um padrão de reparo, baseando-se em termos de edição, fragmentos de palavras e na relação entre o enunciado original e a correção. As regiões candidatas são então passadas para um modelo estatístico baseado em modelos de língua (ML) que julga os candidatos como fluentes ou não. Os autores realizam experimentos sobre um *cópus* próprio de fala espontânea, anotado de acordo com o trabalho de [Bear, Dowding e Shriberg \(1992\)](#). Os resultados dos experimentos mostraram que uma grande quantidade de reparos podem ser detectados sem precisar de informações sintáticas e semânticas. Os autores ainda comentam que a incorporação de métricas prosódicas na detecção de reparos pode melhorar significativamente o desempenho do método.

O trabalho de [Stolcke et al. \(1998\)](#) combina os trabalhos de [Shriberg, Bates e Stolcke \(1997\)](#), [Stolcke e Shriberg \(1996\)](#) para criar um método que segmenta sentenças e detecta disfluências simultaneamente. O método consiste em combinar informações prosódicas e informações lexicais, modeladas por árvores de decisão e modelos de língua baseados em *n*-gramas, respectivamente. Os autores mostraram que uma segmentação de sentenças apropriada e uma remoção de disfluências consistente é capaz de melhorar o desempenho de um reconhecedor automático de voz. Eles apresentam os resultados finais em termos de acurácia para transcrições manuais e automáticas do Switchboard-1. Eles concluem que os resultados devem ser tratados

⁵ Reparos (*repairs*) era um termo empregado antigamente para denotar disfluências de edição.

como *baselines* para trabalhos futuros, pois o modelo pode ser beneficiado com informações morfofossintáticas e sintáticas.

Essa abordagem de combinar as tarefas de SS e DD geralmente retorna em bons resultados (LIU *et al.*, 2006; HASSAN *et al.*, 2014; CHO *et al.*, 2015), já que limites de sentenças e disfluências possuem dependências entre si. Por exemplo, se sabemos que estamos no meio de uma repetição sabemos também que a palavra não precede um limite de sentença. Entretanto, o modo como o problema é modelado para lidar com ambas as categorias difere bastante na literatura. Stolcke *et al.* (1998) e Liu *et al.* (2006) limitaram o potencial do classificador ao dar etiquetas de disfluências e limites de sentença de modo exclusivo, por exemplo, um termo de edição explícito deve ser marcado obrigatoriamente como uma disfluência ou como fim de sentença, mas não ambos.

Hassan *et al.* (2014) propõe uma solução para esse problema ao usar uma abordagem em etapas. Sendo a primeira etapa responsável por segmentar as transcrições em sentenças, e a segunda etapa realiza a detecção/remoção de disfluências sobre essas transcrições segmentadas. Ou seja, a tarefa de SS fica independente da tarefa de DD, entretanto, a tarefa de DD depende da tarefa de SS. A tarefa de DD não foi avaliada intrinsecamente, mas sim extrinsecamente ao verificar o impacto na tarefa de tradução ao variar as estratégias para remoção das disfluências. Uma abordagem mais interessante é apresentada por Cho *et al.* (2015), em que os autores modelam as tarefas de SS e DD em conjunto numa MLP através de um problema de classificação multi-saída (MAURER; PONTIL; ROMERA-PAREDES, 2016), para qual a rede possui duas camadas de saídas distintas e independentes uma da outra, porém os pesos de camadas anteriores são compartilhados para serem usados pelas camadas específicas de cada tarefa. A Figura 21 mostra a arquitetura proposta pelos autores.

Figura 21 – Arquitetura da MLP para tratar SS e DD em conjunto proposta por Cho *et al.* (2015).

Fonte: Cho *et al.* (2015).

As camadas de saída da Figura 21 são passadas diretamente para um CRF de cadeia linear, de modo que eles são combinados através de uma combinação log-linear com as probabilidades das etiquetas retornada pela operação *softmax* e os pesos do CRF. Eles dividiram as disfluências

em cinco classes, incluindo preenchimentos e disfluências de edição, de modo a marcar uma palavra com uma das cinco etiquetas. Com essa modelagem, os autores conseguiram bons resultados sem utilizar informações prosódicas para um *cópus* de palestras do TED-*talks* de 2013. Os autores não reportaram medidas de F_1 para cada classe, portanto os resultados correspondem a média de todos os tipos de disfluências ($F_1 = 0.62$). Os resultados mostraram que, modelando as tarefas em conjunto, a combinação dos classificadores desempenhou melhor que classificadores independentes. Além disso, ao analisar extrinsecamente as duas tarefas em um sistema de tradução automática, os autores mostraram que realizar as duas tarefas simultaneamente retorna resultados melhores do que independentemente.

Modelos de CRF começaram a ser empregados por [Liu et al. \(2005\)](#), quando os autores realizaram experimentos de DD sobre o Switchboard-2 usando três classificadores diferentes: HMM, CRF e Regressão Logística (RL). Sendo que os modelos de HMM e RL têm como objetivo detectar o ponto de interrupção, e então um algoritmo baseado em regras pode ser usado para retornar na sequência e encontrar o ponto de início da disfluência de edição ([HEEMAN; ALLEN, 1994](#); [CORE; SCHUBERT, 1999](#); [LIU; SHRIBERG; STOLKE, 2003](#); [WANG et al., 2010](#)). Já o CRF é modelado usando classes para indicar início (B), meio (I) e fora (O) de uma região de disfluência, sendo que mais duas classes são adicionadas para modelar o fato de que um ponto de interrupção pode ocorrer no início (B+IP) e no meio da região (I+IP). Eles realizaram os experimentos sobre os *cópus* de conversações telefônicas e de programas de notícias jornalísticas das avaliações do NIST-RT 04. Com base na comparação dos classificadores, eles verificaram que os modelos de CRF e RL desempenharam melhor que o HMM, enfatizando que a abordagem de CRF consegue detectar a região do enunciado original e o ponto de interrupção sem a necessidade de regras manuais.

A maioria dos trabalhos de DD foram feitos para a língua inglesa, que possui mais *cópus* anotados. Porém, devemos mencionar que há vários trabalhos que estudam o fenômeno de disfluências e sua correlação com informações prosódicas para língua portuguesa europeia ([MONIZ et al., 2010](#); [MONIZ; ISABEL; TRANCOSO, 2011](#); [MONIZ et al., 2014](#); [MONIZ et al., 2015](#)), e também trabalhos do mesmo projeto que procuram detectá-las, como o de [Medeiros et al. \(2013\)](#). Neste trabalho, os autores avaliaram o desempenho de um modelo de árvore de decisão CART em um *cópus* de aulas universitárias, nomeado de Lectra. Realizaram experimentos focando na identificação automática de regiões de disfluências de modo binário, e distinguindo entre seus elementos estruturais. Concluíram que informações como pausas preenchidas e fragmentos de palavras tem um forte impacto no desempenho. Os experimentos para detectar o enunciado original, o ponto de interrupção e o enunciado corrigido mostraram que *features* como a duração de palavras, *score* de confiança da palavra, e o fato de saber se duas palavras são iguais, têm em geral um alto impacto no desempenho.

Conforme visto até aqui, os autores não seguiram uma consistência nas avaliações para que fosse possível comparar os resultados com outros trabalhos, dado que usaram diferentes

cópus e anotações, realizando até mesmo diferentes tipos de detecções, como detecção de reparos, detecção do enunciado original, detecção de pontos de interrupção e detecção de preenchimentos em conjunto com os anteriores. Foi a partir do trabalho de [Qian e Liu \(2013\)](#) que começou a existir uma consistência na avaliação da tarefa de detecção de disfluências de edição, utilizando o cópus Switchboard-2 para treinamento e teste, com a divisão desses conjuntos feita por [Johnson e Charniak \(2004\)](#). Ainda assim, os trabalhos descritos a seguir devem ser comparados com cautela, pois uma grande parcela deles utilizaram diferentes divisões do cópus para treinamento e teste.

[Qian e Liu \(2013\)](#) propõem um método baseado em três passos usando *n*-gramas de *features* lexicais e lógicas, baseando-se na identidade e nas classes morfosintáticas das palavras. O modelo preditivo empregado por eles é conhecido como Max-Margin Markov Networks (M3Ns) ([TASKAR; GUESTRIN; KOLLER, 2004](#)) com a função de custo de Hamming ponderada para balancear precisão e cobertura. O modelo prevê se uma palavra faz parte de uma das seguintes classes: BE (começo da região de disfluência), IE (dentro da região), EE (fim da região), SE (única palavra da região), O (fora da região). Esse esquema é conhecido como BIES (*Begin-Inside-End-Single*). O primeiro passo é responsável pela detecção de preenchimentos. O segundo detecta palavras que fazem parte de uma região qualquer de disfluência a partir do texto original e do texto filtrado no passo anterior. No terceiro passo, *n*-gramas de *features* adicionais são extraídos do texto filtrado na etapa anterior para melhorar ainda mais a detecção das regiões de disfluências. Após o último passo os autores conseguiram um $F_1 = 0.841$, assumindo o estado-da-arte na época.

O trabalho anterior é melhorado por [Wang, Ng e Sim \(2014\)](#) ao incluir uma estratégia de decodificação usando *beam-search* com algumas *features* adicionais, como duração de pausas, aumento do tamanho dos *n*-gramas e modelos de língua de fluência e de disfluência. Com isso os autores conseguiram melhorar o F_1 para 0.861. A ideia aplicada neste trabalho é de que a sequência de etiquetas previstas pelo classificador pode não ser logicamente válida, e portanto um decodificador ajuda a arrumar possíveis erros. Por exemplo, se o classificador retornar a seguinte sequência de etiquetas:

que já tinha uma filha duas filhas na casa
ND IE ND BE IE IE FE ND ND

em que BE, IE e FE significam respectivamente início, meio e fim da região de disfluência e ND significa que a palavra não faz parte da disfluência, a palavra “já” possui uma classificação que, embora seja possível para um modelo preditivo classificá-la dessa maneira, essa predição não faz sentido, pois a palavra “já” está fora de uma região de disfluência. Dessa forma, um decodificador ou um classificador extra (como um modelo de fluência por exemplo) pode ajudar a deixar na ordem certa de etiquetagem.

[Ferguson, Durrett e Klein \(2015\)](#) também melhoraram o trabalho de [Qian e Liu \(2013\)](#) ao empregar um modelo Semi-CRF ([SARAWAGI; COHEN, 2005](#)) com várias *features* acústicas

e prosódicas. Os autores mostram que ao adotar esse modelo com *features* prosódicas é possível melhorar os resultados para $F_1 = 0.854$, um resultado um pouco abaixo do trabalho de Wang, Ng e Sim (2014).

Redes Neurais Recorrentes (RNNs) foram primeiramente aplicadas para DD por Hough e Schlangen (2015) de maneira incremental, de modo a etiquetar uma palavra por vez, da esquerda para direita, ou seja, para classificar uma palavra olha-se apenas para o contexto à sua esquerda. A rede recorrente possui células recorrentes simples e é alimentada por uma camada de *word embeddings*. Na saída da rede neural é aplicada a função *softmax* para deixar a saída de cada classe em função de probabilidade. As etiquetas propostas diferem dos trabalhos anteriores, pois os autores se preocupam em identificar o trecho do enunciado original e do corrigido, e também preenchimentos. A saída da rede ainda é passada por um decodificador baseado em cadeias de Markov que procura arrumar possíveis erros na detecção do trecho do enunciado corrigido. Sem o uso de informações prosódicas e sem realizar engenharia de *features*, os autores conseguiram bons resultados para as detecções de cada classe, com um $F_1 = 0.711$ para o enunciado original, $F_1 = 0.689$ para a correção, e $F_1 = 0.902$ para os preenchimentos. Hough e Schlangen (2017) aprimoram o trabalho anterior ao introduzir células de LSTM na rede recorrente, durações de palavras como *features*, e realizando as tarefas de SS e DD ao mesmo tempo. Com isso eles conseguiram melhorar os resultados com um $F_1 = 0.720$ para a correção, e $F_1 = 0.918$ para os preenchimentos, porém para a detecção do enunciado original houve uma degradação no desempenho. Por fim, com a análise de erros foi mostrado que quanto maior o tamanho do enunciado original, pior é o desempenho da RNN, sendo que as que usam células de LSTM conseguem lidar um pouco melhor com esse aumento de tamanho.

Redes LSTMs foram primeiramente empregadas por Zayats, Ostendorf e Hajishirzi (2016). O modelo empregado por eles consiste numa RNN bidirecional com células de LSTM que recebe de entrada *word embeddings* e emite a saída para uma camada totalmente conectada com predições em *softmax*. A saída é modelada de acordo com o esquema BIES e, portanto, aplicaram um decodificador baseado em programação linear para incorporar restrições na estrutura da sequência de etiquetas de disfluências. Neste trabalho eles buscaram detectar o enunciado original e o corrigido. Sem usar informações prosódicas, os autores conseguiram um $F_1 = 0.859$. Eles concluem mostrando que disfluências que não são repetições são mais difíceis de serem reconhecidas do que as de outro tipo, como revisões e recomeços.

Wang, Che e Liu (2016) propõem um método conhecido na literatura como Encoder-Decoder (CHO *et al.*, 2014) com mecanismo de atenção. Neste trabalho, eles usam uma rede LSTM para codificar a entrada composto de *word embeddings* em um vetor de tamanho fixo, para então usar outra rede LSTM para decodificar esse vetor em cada instante de tempo. Esse esquema de codificar uma sentença e transformá-la em outra é conhecido como *Sequence to Sequence - Seq2Seq* (SUTSKEVER; VINYALS; LE, 2014). O mecanismo de atenção é responsável por dizer ao decodificador qual é a parte da entrada mais relevante para a classificação em um

determinado instante de tempo. Usando o esquema de predição BIES e com essa abordagem de *Seq2Seq* eles conseguiram um $F_1 = 0.867$. Os autores concluem que uma abordagem que integra informações sintáticas pode atingir um melhor desempenho. Em Wang *et al.* (2017) os autores levam essa ideia adiante e constroem um método neural baseado em transições, e com isso melhoram o resultado para $F_1 = 87.5$.

3.2.2 Preenchimentos

Antes de trazer os trabalhos relacionados, lembramos que nossa definição de preenchimentos, que segue o trabalho de Shriberg (1994), é dividida em: pausas preenchidas (e.g. *uh*, *ah*, marcadores discursivos (e.g. *né*, *então*, *bom*) e termos de edição explícitos (e.g. *dig*, *não*).

Geralmente, a detecção de preenchimentos é vista como uma etapa de pré-processamento para a detecção de disfluências de edição. Isso porque a retirada de preenchimentos pode facilitar a segunda etapa. Por exemplo, na sentença: “foi pra casa *né não uh* foi pra festa”, ao retirar os preenchimentos ela ficaria: “foi pra casa foi pra festa”. Desse modo, o trecho do enunciado original fica mais próximo do trecho corrigido, facilitando as relações de longa dependência para abordagens baseadas em palavras vizinhas. Por isso, vários trabalhos realizam a detecção de preenchimentos mas não reportam os resultados obtidos nessa etapa (LIU *et al.*, 2005; QIAN; LIU, 2013). Outros trabalhos ainda detectam os preenchimentos em conjunto com as disfluências de edição, reportando apenas uma métrica para a detecção final e assim dificultando a análise separada dos resultados (STOLCKE *et al.*, 1998; CHO *et al.*, 2015).

O trabalho de Shriberg, Bates e Stolcke (1997) propôs um método baseado em árvores de decisão que usa apenas informações prosódicas para detectar apenas pausas preenchidas. Os experimentos foram realizados sobre o corpus Switchboard-1. Usando esse método, os autores conseguiram uma cobertura de 92.3% e uma acurácia de 89.7%. Apesar de não reportarem precisão, os resultados mostrados são consideravelmente bons, pois o classificador não baseou-se na identidade das palavras, mas apenas em informações prosódicas. Eles mostraram que informações de silêncio, duração de pausa e pitch são tipicamente boas para essa detecção. Stolcke *et al.* (1998) aprimoraram o trabalho anterior ao detectar disfluências em conjunto com limites de sentenças, e embora fizeram a análise conjunta das tarefas de detecção de preenchimentos e disfluência de edição, eles reportaram uma matriz de confusão na qual foi possível ver que o melhor método (modelo de língua lexical) conseguiu uma acurácia de 100% para a detecção de preenchimentos.

Em contraste com o trabalho de Shriberg, Bates e Stolcke (1997), o trabalho de Snover, Dorr e Schwartz (2004) usa apenas informações lexicais (identidade e classe morfosintática) para detectar preenchimentos em corpus de programas de notícias jornalísticas e de conversações telefônicas do NIST-RT 04. O método usado é conhecido como *Transformation-Based Learning* (TBL), que aprende um conjunto de regras que transformam hipóteses iniciais com o objetivo de reduzir a taxa de erro das hipóteses realizadas. Os resultados indicam uma grande degradação de

desempenho de transcrições manuais para automáticas e também de fala espontânea para outra fala mais preparada. Analisando as principais regras aprendidas, é perceptível que a técnica usada baseia-se fortemente em evidências lexicais, de modo que o classificador chegou a memorizar que expressões como “*you know*” (“você sabe”) e “*I mean*” (“digo”) como sendo preenchimentos em todos os casos. Os autores informam que informações de um *parser* sintático pode ajudar a melhorar casos mais ambíguos e deixar as regras mais generalizáveis. Wang *et al.* (2010) também utilizam regras baseadas em identidades e classes morfossintáticas das palavras para detectar preenchimentos, mostrando que removendo disfluências de edição e mantendo preenchimentos acarreta num melhor desempenho para um método de tradução automática baseado em CRF.

Modelos baseados em redes neurais e CRFs são avaliados por Bertero *et al.* (2015). Os autores propõem quatro abordagens para a detecção simultânea de preenchimentos e disfluências de edição. O primeiro método consiste na aplicação de uma rede neural que recebe uma janela de *word embeddings* como entrada e gera uma saída via *softmax*, conforme (COLLOBERT *et al.*, 2011). A segunda abordagem é a utilização de um CRF de cadeia linear básico, conforme (LIU *et al.*, 2005). A terceira é a combinação dos dois sistemas anteriores através de um algoritmo de votação. A última abordagem consiste em realizar dois passos: o primeiro é aplicado um classificador treinado com todos os dados do conjunto de treinamento, e o segundo é remover as disfluências reconhecidas na primeira etapa, exceto recomeços, e então aplicar um segundo classificador. Eles usaram apenas informações de identidade e classes morfossintáticas para uma janela de palavras como *features*. Avaliaram essas quatro abordagens em um corpus de conversações telefônicas conhecido como Fisher (CIERI *et al.*, 2004), o qual possui transcrições automáticas anotadas manualmente com etiquetas de disfluência. Os melhores resultados foram obtidos ao combinar a saída da rede neural e do CRF usados na quarta abordagem, que consiste nas duas passadas sobre os dados. Mais especificamente, obtiveram um F_1 de 0.908 para a detecção de preenchimentos.

Os trabalhos de Hough e Schlangen (2015) e Hough e Schlangen (2017) descritos na subseção anterior detectam disfluências de edição e também preenchimentos de modo incremental usando redes neurais recorrentes: o primeiro com células simples e o segundo com células de LSTM. Os autores mencionam que detectar o termo de edição explícito dentro da estrutura de disfluências de edição não é o suficiente para o método incremental e, portanto, há a necessidade de detectar e remover pausas preenchidas e marcadores discursivos também. O primeiro trabalho consegue um F_1 de 0.902 em transcrições automáticas usando *word embeddings* e etiquetas morfossintáticas. Já o segundo consegue um F_1 de 0.918 em transcrições automáticas usando também informações prosódicas de duração das palavras.

3.2.3 Síntese

Disfluências de Edição

A Tabela 9 sumariza os trabalhos apresentados na subseção sobre disfluências de edição, apresentando as principais propriedades dos trabalhos. A maioria dos trabalhos relacionados realizaram experimentos para o idioma inglês, exceto o trabalho de Medeiros *et al.* (2013), que foi feito para o português europeu.

Tabela 9 – Resumo das características dos trabalhos de detecção de disfluências de edição.

Trabalhos	Corpus	Estrutura	Método	Features			F ₁
				L	P	M	
(BEAR; DOWDING; SHRI-BERG, 1992)	Próprio	Binário	Regras	x	x		0.462
(HEEMAN; ALLEN, 1994)	Próprio	Binário	ML + Regras	x		x	0.829
(STOLCKE <i>et al.</i> , 1998)	SWB-1	Específica	ML + AD	x	x		-
(LIU <i>et al.</i> , 2005)	SWB-2*	En. Original e PI	CRF	x	x	x	-
(MEDEIROS <i>et al.</i> , 2013)	LECTRA	Binário	AD (CART)		x		0.525
	LECTRA	En. Original	AD (CART)		x		0.412
(QIAN; LIU, 2013)	SWB-2	En. Original	M3N	x		x	0.841
(WANG; NG; SIM, 2014)	SWB-2	En. Original	M3N + <i>beam-search</i>	x		x	0.861
(CHO <i>et al.</i> , 2015)	TED-T 2013 [†]	En. Original	CRF + MLP	x		x	0.617
(FERGUSON; DURRETT; KLEIN, 2015)	SWB-2	En. Original	Semi-CRF	x	x	x	0.854
(HOUGH; SCHLANGEN, 2015)	SWB-2*	En. Original	RNN + HMM	x		x	0.711
	SWB-2*	En. Corrigido	RNN + HMM	x		x	0.689
(ZAYATS; OSTENDORF; HAJISHIRZI, 2016)	SWB-2	En. Original	BiLSTM + Prog. Linear	x		x	0.859
(WANG; CHE; LIU, 2016)	SWB-2	En. Original	Seq2Seq + <i>Attention</i>	x		x	0.867
(HOUGH; SCHLANGEN, 2017)	SWB-2*	En. Original	BiLSTM + HMM	x	x	x	0.601
	SWB-2*	En. Corrigido	BiLSTM + HMM	x	x	x	0.720
	SWB-2*	En. Corrigido	BiLSTM + HMM	x	x	x	0.557 [†]
(WANG <i>et al.</i> , 2017)	SWB-2	En. Original	BiLSTM + Transições	x		x	0.875

Fonte: Elaborada pelo autor.

As *features* foram categorizadas em: (L) lexicais; (P) prosódicas; (M) morfossintáticas. Resultados cujo F₁ não foi reportado pelos autores foram deixados em branco. * Os arquivos do Switchboard-2 usados para teste e treinamento desses trabalhos são diferentes dos usados pelos outros trabalhos relacionados nesta tabela. † Resultados para transcrições automáticas (RAV). ‡ O trabalho de Cho *et al.* (2015) utilizou também um corpus de sentenças paralelas do inglês para o francês providas de reuniões do Parlamento Europeu.

Alguns trabalhos tentaram fazer apenas uso de informações lexicais para tratar disfluências (CLARK; WASOW, 1998; SNOVER; DORR; SCHWARTZ, 2004), concluindo que esse tipo de informação ajuda bastante na tarefa, mas dicas prosódicas ainda são fontes importantes para se lidar com alguns tipos de reparos específicos, como a detecção de ponto de interrupção em disfluências de edição. Trabalhos mais antigos focaram no uso de modelos de língua estatísticos

(HEEMAN; ALLEN, 1994) em conjunto com sistemas de RAV, de modo que a saída do RAV já viesse sem disfluências, fazendo com que a ideia de lidar com as tarefas em conjunto fosse incentivada, por ser mais vantajosa do que tratadas separadamente.

O trabalho de Stolcke *et al.* (1998) não possui resultados reportados em F_1 , mas têm a acurácia geral levando em consideração preenchimentos e todos os tipos específicos de disfluências de edição: repetições, revisões e recomeços. Com o melhor método avaliado, a acurácia reportada é de 92.7% para transcrições manuais e 73.8% para transcrições automáticas. Já o trabalho de Liu *et al.* (2005) reportou os resultados usando a média da métrica *SER* para o enunciado original e para o ponto de interrupção (PI). Isso pois procuraram detectar o ponto de interrupção e o enunciado original simultaneamente. Portanto, não sabe-se realmente qual é o desempenho do CRF para cada estrutura.

Com base na Tabela 9 é possível perceber que a maioria dos trabalhos buscou detectar a estrutura do enunciado original das disfluências, sem diferenciar o seu tipo. Dessa forma, puderam avaliar usando a métrica F_1 de forma binária para essa região de disfluência. Os trabalhos de Hough e Schlangen (2015), Hough e Schlangen (2017) detectaram a região de correção também devido a abordagem incremental utilizada, a qual é necessária para aplicação em tempo real modelada por eles.

Trabalhos mais recentes focam no uso de redes neurais, e também na combinação delas com modelos de CRF e HMM. A maioria desses trabalhos realiza a detecção de disfluências com uma decodificação em cima da saída da classificação anterior, tentando manter a integridade lógica da sequência de etiquetas de disfluências (WANG; NG; SIM, 2014; ZAYATS; OSTENDORF; HAJISHIRZI, 2016; HOUGH; SCHLANGEN, 2017). Essa tendência nos indica possíveis modelos que podem ser explorados nesta dissertação.

É bastante evidente que informações lexicais são importantes para a detecção de disfluências de edição. O único trabalho relacionado que não as usou foi o de Medeiros *et al.* (2013), o qual o fez propositalmente para levantar um estudo de informações prosódicas para a tarefa. Quanto a informações prosódicas, elas foram bastante utilizadas pelos primeiros trabalhos da área. Bear, Dowding e Shriberg (1992) descreve padrões prosódicos que podem ser utilizados para detectar reparos na fala, focando principalmente em padrões de *pitch* e duração de fonemas. Shriberg, Bates e Stolcke (1997) realizaram vários experimentos quantitativos usando árvores de decisão para verificar quais são as informações prosódicas que mais evidenciam diferentes tipos de disfluência, mostrando a importância desse tipo de informação para a tarefa. E, então, vários trabalhos perceberam isso e adotaram quando possível informações prosódicas em seus modelos preditivos (LIU *et al.*, 2005; FERGUSON; DURRETT; KLEIN, 2015; HOUGH; SCHLANGEN, 2017).

É possível ver também que os resultados para detecção do enunciado original no Switchboard-2 melhoraram bastante nos últimos 4 anos apenas trocando o método de classificação. Além disso, é possível ver com os trabalhos de Hough e Schlangen (2015) e Hough e

Schlangen (2017) que o desempenho para transcrições automáticas é consideravelmente pior, até mesmo quando se usa informações prosódicas. Por fim, enfatizamos mais uma vez que é necessário comparar com bastante cautela os trabalhos relacionados, pois além de haver diferentes partições do *cópus*, diferentes modelagens de predição, e diferentes métricas de avaliação, muitos fazem o uso de várias informações extras como modelos de língua de fluência (WANG; NG; SIM, 2014), ou realizam diferentes pré-processamentos que impactam na distribuição das classes, como a eliminação de preenchimentos antes da detecção dos enunciados originais (WANG; CHE; LIU, 2016; WANG *et al.*, 2017).

Preenchimentos

Há poucos trabalhos que detectam preenchimentos, e dentre esses há ainda menos trabalhos que reportam os resultados para essa detecção, portanto não sumarizamos os trabalhos em uma tabela os trabalhos descritos na Subseção 3.2.2.

Pausas preenchidas e marcadores discursivos são fortemente dependentes na identidade das palavras (SHRIBERG; BATES; STOLCKE, 1997). Uma abordagem simples para detecção de preenchimentos é utilizar uma lista de palavras e regras manuais com o auxílio de classes morfossintáticas. Entretanto, classificar corretamente se uma palavra é um preenchimento não é uma tarefa trivial, principalmente no caso da ambiguidade de marcadores discursivos. Por exemplo, é comum falantes da língua portuguesa brasileira usarem a expressão “é” ou “eh” quando estão indecisos na comunicação - caracterizando uma disfluência, porém claramente a palavra “é” também pode ser empregada como um verbo auxiliar. Além disso, erros na saída de sistemas de RAV (especialmente com pausas preenchidas) podem dificultar a detecção dos mesmos (LIU, 2004).

PROJETO DEEPBONDD

Neste capítulo, é apresentada a proposta de pesquisa desta dissertação de mestrado. O projeto recebeu o nome de DeepBonDD (a **Deep** neural approach to **B**oundary and **D**isfluency **D**etection).

Na [Seção 4.1](#) são descritos os dois corpúscos da área clínica utilizados em nossas avaliações, sendo eles compostos de narrativas de pacientes que contaram histórias com apoio de um livro de figuras, com cenas em sequência, sem palavras: o Corpus Cinderela ([Subseção 4.1.1](#)) e o corpus da bateria BALE ([Subseção 4.1.2](#)). Por fim, na [Subseção 4.1.3](#), descrevemos o corpus da Constituição Brasileira de 1988 do Projeto FalaBrasil ([Subseção 2.3.1](#)), que é um corpus de fala preparada modificado para ser usado em nossos experimentos.

O método desenvolvido para as tarefas de Segmentação de Sentenças (SS) e Detecção de Disfluências (DD) é apresentado na [Seção 4.2](#), que inclui: (i) a especificação de uma modelagem generalizável para problemas de classificação sequências ([Subseção 4.2.1](#)), (ii) a descrição do modelo neural camada à camada ([Subseção 4.2.2](#)), e (iii) a metodologia de avaliação utilizada nas avaliações ([Subseção 4.2.3](#)).

Todos os passos de processamento da abordagem DeepBonDD para segmentar uma transcrição em sentenças e remover as disfluências podem ser vistos na [Figura 22](#).

Mais especificamente, após receber uma transcrição limpa (sem sinais de pontuação e com letras em caixa baixa), os passos de processamento do DeepBonDD são:

1. Passar a transcrição paralelamente para um segmentador de sentenças e um removedor de preenchimentos (pausas preenchidas e marcadores discursivos);
2. Combinar a saída desses dois processos, ou seja, são introduzidos os sinais de limites de sentença em seus devidos lugares e as pausas preenchidas que foram classificadas como preenchimentos são removidas;

Figura 22 – Passos de processamento do DeepBonDD

Fonte: Elaborada pelo autor.

3. Aplicar o removedor de disfluências de edição (repetições e revisões), gerando uma transcrição final segmentada e livre de disfluências.

4.1 Córpus

Na área clínica, a obtenção de narrativas de pacientes com apoio de um livro de figuras sem palavras é preferível ao invés da contação de uma estória pelo examinador, para evitar que os pacientes repitam as mesmas palavras usadas pelo examinador. Entretanto, essa forma de obtenção da narrativa não fornece uma estória única a ser usada como fonte do reconto. E embora a estória possa ser conhecida dos pacientes, eles podem ter sido expostos a versões diferentes dela, ou até mesmo terem ouvido a narrativa com ênfases diferentes em certas partes da estória.

4.1.1 Córpus Cinderela

O primeiro córpus usado nesta dissertação é um conjunto de narrativas recontadas e transcritas de fala comprometida e normal, sendo que as narrativas são baseadas em um livro que contém uma sequência de figuras (sem palavras) sobre a estória da Cinderela. A sequência de 22 figuras do livro e a estória prototípica estão ilustradas na [Figura 23](#). Um total de 28 proposições ajudam o anotador a computar o escore do paciente no teste, as quais são apresentadas no [Apêndice A](#).

Na avaliação, o examinador pergunta ao paciente se este se lembra da estória da Cinderela. Em caso positivo, o examinador apresenta ao participante da pesquisa o livro, com as cenas em sequência, sem palavras, contando a estória. Quando o participante termina de ver todas as

Figura 23 – Sequência de figuras e estória prototípica da Cinderela.

Fonte: Adaptada de Toledo (2017).

cenar, o examinador solicita que o participante conte a história com suas próprias palavras. Não é estipulado um tempo limite para a narração, durante a qual o examinador não faz nenhum comentário, a menos que o participante tenha dificuldades em iniciar ou continuar a estória, fazendo uma pausa longa; nesse caso, o examinador intervém com questões genéricas, para encorajar o participante a continuar a narração. Caso ele não consiga, o examinador pergunta se ele se lembra das cenas da estória, caso não, é permitido que o participante veja o livro novamente. Todo o teste é registrado com câmara de vídeo e com gravador de áudio. As narrativas foram coletadas pela Faculdade de Medicina da Universidade de São Paulo (FMUSP). Os áudios das narrativas foram transcritos manualmente por um anotador treinado, que avaliou as transcrições contando as proposições recontadas.

Como o processo de reconto é falado em voz alta pelo paciente de maneira espontânea, nas transcrições são encontrados comentários fora do assunto da estória, disfluências e trechos agramaticais, os quais foram marcados pelo anotador de acordo com o manual apresentado no Apêndice A. Para exemplificar, a Figura 24a traz uma narrativa transcrita, sem marcações de limites de sentença, mas com as marcações de algumas disfluências e de trechos agramaticais feitas pelo anotador. Essas transcrições foram previamente usadas em trabalhos de detecção automática da progressão da DA Aluísio, Cunha e Scarton (2016), e em análises linguísticas dos três grupos de pacientes (com Doença de Alzheimer (DA), com Comprometimento Cognitivo Leve (CCL), e Controle (CTL)) em Toledo (2017).

Já a Figura 24b traz uma transcrição limpa, sem as marcações do anotador. Esse tipo de transcrição é o que espera-se como saída de um sistema de RAV. E como o propósito do projeto no qual esta pesquisa está incluída é automatizar o processo de transcrição, retiramos

Figura 24 – Exemplos de transcrições com e sem marcações da estória da Cinderela.

(a) Transcrição original

ela morava com a madrasta as irmã <empty>né</empty> e ela era diferenciada das três era maltratada ela tinha que fazer limpeza na casa toda no castelo alias e as irmãs não faziam nada...((pausa 3 segundos)) até que um dia chegou um convite do rei ele ia fazer um baile...((pausa 3 segundos)) e a madrasta então <empty>é::: </empty> colocou que todas as filhas elas iam menos a cinderela... ((pausa 6 segundos)) bom como ela não tinha o vestido sapato as coisas tudo então ela mesmo teve que fazer a roupa dela começou a fazer as irmãs não queriam que ela fosse de jeito nenhum... <empty>eu sou muito resumista nas histórias ((risos))</empty>

(b) Transcrição limpa

ela morava com a madrasta as irmã né e ela era diferenciada das três era maltratada . ela tinha que fazer limpeza na casa toda no castelo alias e as irmãs não faziam nada até que um dia chegou um convite do rei ele ia fazer um baile e a madrasta então é colocou que todas as filhas elas iam menos a cinderela bom como ela não tinha o vestido sapato as coisas tudo então ela mesmo teve que fazer a roupa dela começou a fazer as irmãs não queriam que ela fosse de jeito nenhum eu sou muito resumista nas histórias

Fonte: Elaborada pelo autor.

essas marcações para deixá-las mais próximas das saídas de RAV.

Os dados demográficos dos 60 pacientes submetidos ao processo descrito são apresentados na [Tabela 10](#), para os quais 20 são pacientes com DA, 20 com CCL, e 20 CTL.

Tabela 10 – Informações demográficas dos participantes na avaliação com a estória da Cinderela.

Informação	CTL	CCL	DA
Idade média	74.8	73.3	78.2
Escolaridade média (anos)	11.4	10.8	8.6
Num. de homens/mulheres	4/16	6/14	10/10

Fonte: Dados da pesquisa.

Para a tarefa de SS, o corpus com as transcrições limpas foi anotado com informações de limites de sentença por três anotadores, ao longo do primeiro ano de duração da pesquisa. Como a anotação pelos três anotadores foram feitas em períodos diferentes, alguns experimentos foram realizados usando anotações diferentes. Definimos como **Cinderela versão 1** o corpus anotado por somente um anotador, o qual foi usado nas avaliações publicadas em [Treviso, Shulby e Aluísio \(2017\)](#). E definimos como **Cinderela versão 2** o corpus anotado usando informações de concordância de três anotadores, o qual foi usado para os experimentos reportados nesta dissertação.

4.1.1.1 Anotação de Limites de Sentença: Cinderela versão 1

Essa versão do corpus foi anotada por um único anotador, que usou apenas dicas lexicais, sintáticas e semânticas para anotar os limites de sentença, com o objetivo de garantir uma sentença bem formada para o sucesso de um *parser* sintático ([CUNHA et al., 2015](#)), garantindo que cada sentença tenha um verbo principal. E apesar dessa estratégia ter a tendência de formar sentenças linguisticamente bem formadas, não foram utilizadas dicas prosódicas como sugerido por [Saffran, Berndt e Schwartz \(1989\)](#).

Os áudios relativos às transcrições também precisaram de ajustes. Mais especificamente, algumas transcrições possuíam palavras a mais ou a menos do que as reproduzidas pelos pacientes

durante o reconto. Além disso, as transcrições também possuíam trechos relativos a falas do examinador, que não é algo desejado para a automação do processo, já que espera-se que um paciente, sozinho, reconte a história em voz alta e um sistema de RAV faça a transcrição automaticamente. Ou seja, em nível de palavra, as transcrições não estavam totalmente fiéis ao conteúdo do áudio.

Para resolver esse problema, realizamos quatro ajustes. O primeiro ajuste foi eliminar os trechos do áudio em que o examinador faz comentários em voz alta. A partir disso, realizamos os outros três ajustes, retornando mais duas versões para cada transcrição: com e sem as marcações dos ajustes. Os três ajustes realizados e suas respectivas marcações foram as seguintes:

- **Inclusão [x]:** Incluímos e marcamos com [x] os trechos x que estavam no áudio mas não estavam na transcrição;
- **Remoção {x}:** Removemos e marcamos com {x} os trechos x que estavam na transcrição mas não estavam no áudio;
- **Substituição [x | y]:** Substituímos x por y e marcamos com [x | y] os trechos x que estavam na transcrição mas que eram pronunciados como y no áudio.

No total, o conjunto de 60 transcrições anotadas retornou as estatísticas apresentadas na [Tabela 11](#). A quantidade de instâncias (palavras) para usar em algoritmos de aprendizado de máquina não é muito grande, o que nos fez pensar em estratégias de validação cruzada para avaliação dos métodos. Além de que para um problema de classificação binária de SS, temos um conjunto de dados desbalanceado com apenas 7.74% de instâncias que correspondem a limites de sentença.

Tabela 11 – Estatísticas do cópus Cinderela v1.

Propriedade	CTL	CCL	DA	Total
# Transcrições	20	20	20	60
# Palavras	8096	7905	7806	23807
# Sentenças	621	552	670	1843
# Sent. / # Transc.	31.05	27.60	33.50	30.72
# Palav. / # Sent.	13.04	14.32	11.65	12.92
Duração	1h 09m	1h 12m	1h 50m	4h 11m

Fonte: Dados da pesquisa.

Podemos ver também que pacientes com DA produziram, em média, mais sentenças em suas narrativas, porém essas sentenças são menores que as de CTL e CCL. Isso acontece, principalmente, devido ao fato de que pacientes com DA tem uma forte tendência a incluir disfluências em seus recontos.

4.1.1.2 Anotação de Limites de Sentença: Cinderela versão 2

Na versão 2, o corpus de narrativas transcritas de recontos da estória da Cinderela foi anotado por mais dois pesquisadores. Assim como na versão 1, a anotação foi feita para marcar limites de sentença nas transcrições. Entretanto, dessa vez foram usadas pistas prosódicas para realizar a marcação pelos dois anotadores. A anotação foi feita sobre as transcrições arrumadas pelos três ajustes realizados para a versão 1¹.

Dessa maneira, foi possível alinhar três versões de cada transcrição, cada uma de um anotador diferente: Anotador 1, com as mesmas transcrições ajustadas pelo anotador da versão 1; Anotador 2 e 3, que analisaram as transcrições ajustadas. Vale notar que foram removidos os sinais de limites de sentenças dessas transcrições ajustadas para não enviesar os novos anotadores. Mostramos na [Tabela 12](#) as estatísticas referentes a cada anotador, sendo que não mostramos informações sobre o número de transcrições, número de palavras e duração das gravações pois são exatamente as mesmas da versão 1.

Tabela 12 – Estatísticas do corpus Cinderela para cada anotador.

Propriedade	Anotador 1			Anotador 2			Anotador 3		
	CTL	CCL	DA	CTL	CCL	DA	CTL	CCL	DA
# Sentenças	621	552	670	616	598	864	654	618	794
# Sent. / # Transc.	31.05	27.60	33.50	30.80	29.90	43.20	32.70	30.90	39.70
# Palav. / # Sent.	13.04	14.32	11.65	13.14	13.22	9.03	12.38	12.79	9.83

Fonte: Dados da pesquisa.

A anotação final utilizada para os experimentos foi a realizada pelo anotador 3, no papel de adjudicador, já que esse anotador simplesmente revisou as discordâncias do anotador 1 e 2 levando em consideração dicas prosódicas. Essa anotação resultou num total de 2066 sentenças, o que representa 8.67% de instâncias correspondentes a limites de sentença.

Quando comparamos o tamanho de sentenças entre anotadores, podemos ver que para as transcrições de narrativas de pacientes com DA houve mais diferença, já que as transcrições produzidas por esses pacientes têm muito mais disfluências e comentários não relacionados a tarefa, dificultando a análise semântica e sintática. Comparando os anotadores 1 e 3, é possível verificar que o último fez mais marcações de limites de sentenças, o que consequentemente acarretou em mais sentenças por transcrições e em sentenças menores. Em princípio, sentenças menores são mais fáceis de serem modeladas para a tarefa de SS, pois os limites de sentença estarão associados a dependências próximas. O desvio padrão para o tamanho das sentenças pelo anotador 3 foi de 7.21, já para o primeiro anotador foi de 7.91, o que indica que além das sentenças produzidas pelo terceiro anotador serem menores, elas também são mais estáveis em

¹ Todas as versões com as devidas anotações estão disponíveis a outros pesquisadores, após pedidos por e-mail.

seu tamanho. Uma possível explicação para essas diferenças está no auxílio de informações prosódicas usadas pelo terceiro anotador. Por exemplo, o anotador 1 fez a seguinte marcação:

o amigo achou pra ela esse aqui . eu não lembro

Enquanto o anotador 3 fez a seguinte marcação escutando o áudio para desambiguar o trecho:

o amigo achou pra ela . esse aqui eu não lembro

Em suma, com a versão 2 do *córpus Cinderela* espera-se que os resultados sejam melhores dos que os da versão 1 publicados em [Treviso, Shulby e Aluísio \(2017\)](#), pois a contribuição das *features* prosódicas podem ser maiores.

As informações de concordância seguindo a medida Kappa ([CARLETTA, 1996](#)), F_1 e SER são apresentadas na [Tabela 13](#).

Tabela 13 – Concordância entre anotadores para a Cinderela versão 2.

Anotadores		Kappa			F_1			SER^*		
Ref.	Hip.	CTL	CCL	DA	CTL	CCL	DA	CTL	CCL	DA
A1	A2	0.82	0.80	0.74	0.83	0.82	0.77	0.34	0.38	0.54
A3	A1	0.90	0.88	0.81	0.91	0.89	0.83	0.19	0.22	0.31
A3	A2	0.89	0.89	0.89	0.90	0.90	0.90	0.20	0.20	0.21

Fonte: Dados da pesquisa.

* A métrica SER não é simétrica.

Em geral, bons valores de concordância entre anotadores usando a métrica Kappa dependem da tarefa sendo tratada. Na [Tabela 13](#) podemos perceber a dificuldade da tarefa para os textos menos bem formados (DA). Mas, baseando-se em trabalhos relacionados, valores de Kappa para marcações de limites de sentença estão entre 0.8 e 0.9 ([STEVENSON et al., 1999](#); [BATISTA et al., 2012](#)). Sendo assim, a concordância do anotador 3 com os outros anotadores está consideravelmente boa.

As outras métricas foram usadas para fins de ilustração, já que reportaremos os resultados dos experimentos pilotos usando elas, e consequentemente podemos comparar o desempenho do método com o desempenho humano.

4.1.1.3 Anotação de Preenchimentos

Para anotar os preenchimentos, usamos o *córpus Cinderela v2*, anotado conforme mostrado na subseção 4.1.1.2. Isso significa que informações de limites de sentença estavam disponíveis no momento da anotação.

Conforme vimos na [Seção 2.2](#), os preenchimentos podem ser divididos em pausas preenchidas (por exemplo, *ahn*, *eh*), marcadores discursivos (por exemplo, *né*, *bom*), e termos de edição explícitos (por exemplo, *não*, *digo*). Os anotadores foram instruídos para marcar palavras com essas três categorias, de acordo com o manual apresentado no [Apêndice B](#). Para casos duvidosos, foi sugerido aos anotadores que escutassem o áudio da transcrição sendo analisada. Para se familiarizar com a tarefa, os anotadores primeiramente anotaram 6 transcrições, sendo 2 transcrições de cada grupo paciente. Após esse passo os anotadores discutiram sobre os erros e os acertos, e então o manual de anotação foi modificado.

Embora seja fácil identificar pausas preenchidas, o mesmo não pode ser dito sobre marcadores discursivos e termos de edição explícitos. O manual usado nas avaliações do NIST-RT ([STRASSEL, 2003](#)) limitou os marcadores discursivos a um conjunto fixo pré-determinado. Como não temos uma grande quantidade de dados, não seguimos essa estratégia e deixamos os anotadores marcarem quaisquer marcadores discursivos. Termos de edição explícitos são mais raros que os outros dois grupos e, portanto, também não foram limitados a um conjunto fixo.

Portanto, nessa etapa dois anotadores marcaram as palavras com etiquetas das três categorias de preenchimento, levando em consideração informações de limites de sentenças e informações do áudio para decidir casos difíceis. Mostramos na [Tabela 14](#) as estatísticas referentes aos dois anotadores.

Tabela 14 – Estatísticas de preenchimentos do corpus Cinderela v2 para cada anotador.

Propriedade	Anotador 1				Anotador 2			
	CTL	CCL	DA	Total	CTL	CCL	DA	Total
# Pausas Preenchidas	38	79	50	167 (0.26)	24	64	13	101 (0.19)
# Marcadores Discursivos	139	149	183	471 (0.72)	145	147	145	437 (0.80)
# Termos de Ed. Explícitos	1	8	3	12 (0.02)	1	3	3	7 (0.01)
Total	178	236	236	650	170	214	161	545

Fonte: Dados da pesquisa.

A anotação final utilizada para os experimentos foi a realizada pelo anotador 1, por dois motivos: (i) esse anotador conhecia melhor a teoria envolvida pois esteve envolvido com a criação do manual de anotação; (ii) identificou mais palavras como sendo disfluências (650 vs 545). Portanto, todas as análises apresentadas a seguir foram realizadas sobre os dados anotados pelo anotador 1.

É possível ver na tabela que marcadores discursivos foram anotados mais vezes, seguido de pausas preenchidas e termos de edição explícitos, sendo que o último constitui apenas 2% dos preenchimentos anotados. Isso constitui uma distribuição de categorias bastante desbalanceada (72% MD vs 26% PP vs 2% TEE). Também é possível observar que há mais preenchimentos para pacientes de CCL e DA, provavelmente porque a fala desses pacientes é comprometida inerentemente devido as suas condições neuropsicológicas.

Embora não conseguimos mais que dois anotadores para verificar qual anotador possui mais concordância com outros, é possível utilizar a anotação do anotador 2 como predições para verificar a taxa de acerto de acordo com observações humanas. Isso é mostrado em termos de F_1 em conjunto com informações de concordância seguindo a medida Kappa na [Tabela 15](#).

Tabela 15 – Concordância entre anotadores para anotação de preenchimentos.

Categoria	Kappa			F ₁		
	CTL	CCL	DA	CTL	CCL	DA
Pausas Preenchidas	0.64	0.81	0.35	0.65	0.81	0.35
Marcadores Discursivos	0.87	0.88	0.85	0.87	0.88	0.85
Termos de Ed. Explícitos	1.00	0.18	0.67	1.00	0.18	0.67
Todas Categorias	0.83	0.84	0.67	0.83	0.85	0.67

Fonte: Dados da pesquisa.

Na [Tabela 15](#) podemos perceber a dificuldade da tarefa para os textos menos bem formados (DA), da mesma forma como aconteceu para a anotação de limites de sentença. Além disso, houve mais concordância para anotar marcadores discursivos do que as outras duas categorias. Na verdade, a categoria de termos de edição explícitos foi adicionada na tabela apenas para curiosidade, pois ela acontece muito raramente (e.g. para CTL tem apenas 1) e, portanto, foi desconsiderada.

Identificar preenchimentos é considerada uma tarefa fácil na literatura, vários trabalhos os detectam usando listas de palavras e regras manuais ([STOLCKE et al., 1998](#); [LIU et al., 2005](#); [WANG et al., 2010](#)), por isso acreditamos que a concordância para a anotação de preenchimentos ficou dentro de uma margem razoavelmente boa (0.67 - 0.84). Os valores de F_1 só foram reportados para que em nossos experimentos o desempenho humano possa ser comparado com o desempenho do método automático.

Para verificar se os pacientes tem uma tendência em cometer disfluências logo no início do conto, com relação aos preenchimentos, calculamos os a quantidade média de preenchimentos cometidos em cada $[x\%, x + 10\%]$ sentenças da transcrição, de modo que incrementamos 10% ao valor de x até chegar a 100%. Isso é demonstrado na [Figura 25](#) para o cópus composto pelas transcrições de todos os pacientes.

Podemos perceber na [Figura 25](#) que a maioria dos preenchimentos são cometidos nas 30% sentenças iniciais. Além disso, é possível ver que os pacientes introduzem preenchimentos novamente aos chegar nas últimas 90% sentenças das transcrições. A proporção de cada tipo de preenchimento se manteve do primeiro ao último trecho da transcrição.

Em termos de estrutura narrativa, as 28 proposições da estória da Cinderela pode ser agrupada em quatro componentes:

- Orientação: 1 a 7.

Figura 25 – Quantidade de preenchimentos por pedaço das transcrições.

Fonte: Elaborada pelo autor.

- Problema: 9, 12, 18, 22 e 24.
- Desenvolvimento: 8,10,11,13,14,15,16,17,19,20,21,23,25,26.
- Desfecho: 27,28.

Com relação aos componentes da macroestrutura, os preenchimentos ocorrem com mais frequência nos componentes de orientação e desfecho, preservando mais o núcleo da estrutura (problema/desenvolvimento).

Já foi mostrado por [Shriberg \(1994\)](#) que, para fala espontânea de conversações telefônicas, quanto maior o tamanho do enunciado, maior a chance de ocorrer disfluências, de modo que pelo menos uma disfluência ocorre a cada 20 palavras. Decidimos verificar isso para o nosso cenário de fala comprometida ao plotar a quantidade média de preenchimentos por tamanho da sentença/número de palavras na [Figura 26](#). Para remover *outliers* decidimos incluir no gráfico apenas os resultados para tamanhos de sentenças que ocorrem pelo menos 3 vezes no córpus.

Na [Figura 26](#) é possível ver que conforme o tamanho da sentença aumenta, maior é a quantidade média de preenchimentos, e que só a partir de sentenças com 30 palavras é que a taxa de uma palavra é ultrapassada. Além disso, é possível ver também que, em geral, a proporção dos tipos de preenchimentos se mantém. Vale observar que não há disfluências para sentenças de tamanho 32, mas há mais de 3 sentenças com esse tamanho no córpus e por isso ela foi adicionada ao gráfico.

4.1.1.4 Anotação de Disfluências de Edição

A anotação de disfluências de edição foi realizada logo após a anotação de preenchimentos. Nessa etapa havia informações de limites de sentença e também as marcações de preenchimentos.

Figura 26 – Quantidade média de preenchimentos por tamanho de sentença.

Fonte: Elaborada pelo autor.

Conforme vimos na [Seção 2.2](#), as disfluências de edição possuem uma estrutura na qual um enunciado original vem seguido de um ponto de interrupção onde o interlocutor pode explicitar um termo de edição para a correção que vem em seguida. As disfluências de edição podem ser divididas de acordo com seu tipo em quatro categorias:

1. repetições (<pra ela> pra ela ir ao castela);
2. revisões (<uma filha> duas filhas);
3. recomeços (<ela tá> aqui é uma noiva); e
4. disfluências complexas, que abrangem as três anteriores contidas em outras.

O anotador foi instruído a marcar apenas as palavras do enunciado original com essas as três primeiras categorias, de acordo com o manual do [Apêndice B](#). Para casos duvidosos, foi sugerido ao anotador que escutasse o áudio da transcrição sendo analisada.

A estratégia de marcar as palavras do enunciado original com os três tipos de disfluências de edição é interessante pois é possível deixar a anotação binária, futuramente, indicando apenas se um dado trecho é ou não uma região de disfluência. Além disso, por este trabalho estar contido em um cenário de pesquisa clínica, pode ser interessante no futuro distinguir cada tipo de disfluência. A desvantagem dessa anotação é quanto ao poder de identificar a quantidade exata de regiões de disfluências. Por exemplo, o trecho abaixo:

ele que que que deu a carta
ND REP REP ND ND ND ND

possui 2 repetições em seguida (*que que*). Porém, o exemplo a seguir possui uma repetição apenas, mas duas marcações de repetições:

pra ela pra ela ir ao castelo
 REP REP ND ND ND ND ND

Esse problema pode ser corrigido ao anotar as disfluências usando o esquema BIES, no qual são dadas etiquetas para limitar o início e o fim de uma região de disfluência. Porém, neste trabalho, devido a baixa quantidade de dados e ao objetivo de discernir entre tipos de disfluência, optamos por apenas anotar o tipo de disfluência. Mas é possível converter a anotação utilizada aqui no esquema BIES, pois as regras de anotação usados no manual indicavam que a anotação deveria ser feita por região. Desse modo, é possível obter a palavra de início e de fim da região de disfluência. Ou seja, o que fizemos na verdade foi distribuir a etiqueta da região para cada palavra contida nela.

Uma limitação da nossa anotação é a não anotação de disfluências complexas; nos limitamos a anotar apenas as disfluências mais internas, deixando a mais abrangente intacta. Para resolver esse problema é possível aplicar o método de remoção de disfluências repetidamente, eliminando a disfluência mais interna e detectando a mais abrangente em seguida. Por exemplo, o seguinte trecho com disfluências possui uma repetição contida em outra repetição:

foram com a a com a cinderela
 ND ND REP ND ND ND ND

No primeiro passo, o enunciado original da repetição interna seria removido, formando uma nova repetição localmente conectada:

foram com a com a cinderela
 ND ND ND ND ND ND

Embora essa nova repetição (*com a com a*) não esteja anotada, o método preditivo treinado sobre o corpus anotado ainda é capaz de eliminá-la, pois já aprendeu os padrões de disfluência de edição em uma fase anterior.

Em relação a cada tipo de disfluência de edição, já foi mostrado na literatura que as repetições são mais fáceis de serem identificadas do que outros tipos, principalmente recomeços, por serem mais raros (LIU, 2004). Quanto aos recomeços, eles são frequentemente confundidos com sentenças incompletas (STRASSEL, 2003), algo que não anotamos no nosso dataset. Mostramos na Tabela 16 as estatísticas referentes a anotação do único anotador.

Tabela 16 – Estatísticas de disfluências de edição do corpus Cinderela v2.

Propriedade	CTL	CCL	DA	Total
# Repetições	349	267	322	938 (0.53)
# Revisões	179	260	252	691 (0.39)
# Recomeços	51	33	65	149 (0.08)
Total	579	560	639	1778

Fonte: Dados da pesquisa.

É possível ver na [Tabela 16](#) que repetições foram anotadas mais vezes, seguidas de revisões e recomeços, sendo que o último constitui apenas 8% das disfluências anotadas. Isso constitui uma distribuição de categorias bastante balanceada entre repetições e revisões (53% vs 39%), porém recomeços ainda são raros em nosso cópus. Também é possível observar que há mais disfluências de edição para pacientes de DA do que para CTL e CCL.

Apresentamos na [Figura 27](#) a quantidade média de disfluências por trecho da transcrição, apresentado anteriormente para os preenchimentos, porém agora com disfluências de edição.

Figura 27 – Quantidade de disfluências de edição por pedaço das transcrições.

Fonte: Elaborada pelo autor.

É possível perceber na [Figura 27](#), como visto anteriormente na [Figura 25](#), que a maioria das disfluências são cometidas no início e no fim dos recontos, com alguns picos no meio das transcrições.

Plotamos a quantidade média de disfluências de edição por tamanho da sentença/número de palavras na [Figura 28](#). Para remover *outliers* decidimos incluir no gráfico apenas os resultados para tamanhos de sentenças que ocorrem pelo menos 3 vezes no cópus.

Na [Figura 28](#) é possível ver que conforme o tamanho da sentença aumenta, maior é a quantidade média de disfluências de edição, e que a partir de sentenças com 14 palavras a taxa de uma disfluência por sentença é ultrapassada. Além disso, é possível ver também que, em geral, a proporção dos tipos de disfluências de edição se mantém para cada tamanho de sentença: mais repetições que revisões e recomeços. Outra observação interessante é que sentenças com mais de 20 palavras apresentam pelo menos duas disfluências, o dobro do que o encontrado por [Shriberg \(1994\)](#) em fala espontânea de conversações telefônica. Isso caracteriza um aspecto importante da fala comprometida: o grande número de disfluências.

Figura 28 – Quantidade média de disfluências de edição por tamanho de sentença.

Fonte: Elaborada pelo autor.

4.1.2 *Cópus BALE*

Este cópus faz parte da Bateria de Avaliação da Linguagem no Envelhecimento (BALE - *Battery of Language Assessment in Aging*), que está em processo de validação e produção por pesquisadores da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) (JERÔNIMO *et al.*, 2016).

Um dos testes dessa bateria consiste na apresentação de um conjunto de sete figuras, que contam a história de um garoto que esconde um cachorro que ele encontrou na rua, conhecida na literatura como *The dog story* (BOEUF, 1971). Os participantes são solicitados a observar as figuras dispostas em sequência e, assim que eles se sentem confiantes para contar a história, o relato é gravado. Assim como na produção do cópus da Cinderela, o examinador do teste tenta não interferir. A avaliação feita pelo examinador inclui a quantidade e qualidade de proposições recontadas. Avaliação complementares incluem aspectos semânticos e sintáticos, quantitativos e qualitativos.

Como este cópus também é composto de narrativas de pacientes, podemos avaliar o desempenho de nossos métodos em dados do mesmo domínio que da Cinderela (fala comprometida), mas com a diferença de história e vocabulário das narrativas. Ou seja, podemos avaliar a robustez do nosso método.

Para realizar a avaliação de robustez, os pesquisadores da PUCRS selecionaram e anotaram um conjunto de 10 narrativas com informações de limites de sentença. Informações demográficas e propriedades das narrativas são mostradas na Tabela 17.

As marcações de limites de sentença deste cópus foram feitas por um único anotador da PUCRS, usando evidências sintáticas, semânticas e prosódicas. Em uma comparação com o cópus da Cinderela, este cópus apresenta menos sentenças e palavras no total, menos sentenças por transcrição, e as sentenças possuem, em média, menos palavras. Essas propriedades refletem

Tabela 17 – Informações demográficas e propriedades do *córpus* BALE.

Propriedade	CTL	CCL	Total
Idade média	73.3	67.2	70.8
Escolaridade média (anos)	6.8	4.0	5.7
Num. de homens/mulheres	4/2	0/4	4/6
# Transcrições	6	4	10
# Palavras	611	482	1093
# Sentenças	98	68	166
# Sent. / # Transc.	16.3	17.0	16.6
# Palav. / # Sent.	6.2	7.1	6.6
Duração	4m	4m	8m

Fonte: Dados da pesquisa.

o tamanho da estória do cachorro, que é muito mais curta do que a da Cinderela.

4.1.3 *Córpus Constituição*

O *córpus* Constituição foi disponibilizado pelo projeto FalaBrasil ([Subseção 2.3.1](#)) e seu conteúdo é estruturado de acordo com a Constituição Brasileira de 1988 ([BATISTA *et al.*, 2013](#)). A fala deste *córpus* pode ser categorizada como fala preparada, mais especificamente com textos falados em voz alta. Este *córpus* foi utilizado apenas para a tarefa de SS, pois não possui anotações de DD. Entretanto, para utilizar ele para SS foi necessário aplicar vários passos de pré-processamento descritos abaixo, já que o *córpus* original também não possui anotações de limites de sentença.

As transcrições manuais deste *córpus* não foram feitas em nível de palavra e, consequentemente, havia trechos pronunciados que não condiziam foneticamente com as palavras transcritas. Por exemplo, números eram colocados através de algarismos: “Cap. V”, “100,000 reais”, “1º inciso”, etc. ao invés de “Cap. cinco”, “cem mil reais”, “primeiro inciso”. Apesar de um modelo lexical e um modelo prosódico conseguirem tratar essas informações independentemente, quando tenta-se usar ambos simultaneamente em nosso método é necessário alinhar essas informações, de modo que a *i*-ésima palavra da transcrição corresponda com a *i*-ésima palavra do áudio. Para resolver esse problema, convertemos todos os algarismos de numerais e cardinais de sistemas numéricos romanos e arábicos para a respectiva forma escrita por extenso.

Apesar dessa conversação resolver o problema de alinhamento, cada arquivo de transcrição possuía apenas 30s de fala transcrita, então nenhum deles tinha mais de uma sentença. Portanto, se os pares de transcrições e áudios fossem passados para um modelo preditivo que aceitasse essa estrutura como uma instância, bastaria o modelo prever um limite de sentença para a última palavra de cada instância. Para resolver esse problema, decidimos agrupar as transcrições de acordo com a estrutura da constituição mostrada na [Figura 8](#). Mais especificamente, agrupamos os arquivos por artigos, de modo que ignoramos os cabeçalhos de subestruturas pais

de artigos. A [Figura 29](#) mostra o resultado desse processo para um conjunto de artigos.

Figura 29 – Exemplo de um artigo da constituição após agrupar arquivos por artigos.

artigo terceiro constituem objetivos fundamentais da república federativa do brasil dois pontos **inciso um** construir uma sociedade livre justa e solidária **inciso dois** garantir o desenvolvimento nacional **inciso três** erradicar a pobreza e a marginalização e reduzir as desigualdades sociais e regionais **inciso quatro** promover o bem de todos sem preconceitos de origem raça sexo cor idade e quaisquer outras formas de discriminação

Fonte: Elaborada pelo autor.

Desse modo, bastaria colocar um limite de sentença no começo de cada subestrutura de um artigo, como incisos, alíneas e parágrafos. Entretanto, as evidências lexicais seriam fortes para um dado modelo de predição; bastaria notar que elas iniciam cada subestrutura (marcadas em negrito na figura) para classificar corretamente todas as instâncias de treinamento. Não obstante, resolvemos facilmente esse problema com a remoção dessas evidências lexicais usando expressões regulares com *negative look-behind* de preposições. O resultado para o exemplo anterior está ilustrado na [Figura 30](#).

Figura 30 – Exemplo de um artigo pré-processado da constituição após retirar pistas lexicais.

constituem objetivos fundamentais da república federativa do brasil dois pontos construir uma sociedade livre justa e solidária garantir o desenvolvimento nacional erradicar a pobreza e a marginalização e reduzir as desigualdades sociais e regionais promover o bem de todos sem preconceitos de origem raça sexo cor idade e quaisquer outras formas de discriminação

Fonte: Elaborada pelo autor.

Apesar de realizar o pré-processamento usando expressões regulares para remover dicas lexicais, algumas dicas poderiam ter sido passadas despercebidas, pois uma estrutura poderia ter terminado erroneamente com uma preposição ambígua. Isso porque não queríamos remover casos em que a pista lexical faz parte do texto, como “no artigo segundo é descrito”², ou seja, desejamos manter anáforas e catáforas. Portanto, para garantir o sucesso do pré-processamento, realizamos uma inspeção manual nos arquivos originais para casos em que as pistas lexicais foram eliminadas e a sentença produzida tinha tamanho menor que 5.

Após todos esses passos de processamento, o corpus resultante é o que chamamos em nossos experimentos de Constituição. Esse corpus resultante tem 357 transcrições, e é cerca de três vezes maior que o corpus Cinderela. Como desejamos avaliar se o tamanho do corpus influencia nos resultados, selecionamos 60 transcrições aleatoriamente, com a única condição

² Observe que “no” é uma contração de uma preposição mais um artigo (em + o); a preposição olhada negativamente pela expressão regular.

de que o tamanho médio de sentenças nas transcrições deveriam ser maiores que 12. O cópus composto por todas as transcrições foi denominado de Constituição L, enquanto o cópus menor foi denominado de Constituição S. As estatísticas de ambos são apresentadas na [Tabela 18](#).

Tabela 18 – Estatísticas dos cópus Constituição L e Constituição S.

Propriedade	Const. L	Const. S
# Transcrições	357	60
# Palavras	63275	30521
# Sentenças	2698	1409
# Sent. / # Transc.	7.6	23.5
# Palav. / # Sent.	23.4	21.7
Duração	7h 39m	3h 43m

Fonte: Dados da pesquisa.

Vale notar que este cópus não possui informações demográficas, pois foi lido em voz alta por apenas uma pessoa, da qual não encontramos informações.

Comparando com o Cópus Cinderela v2, podemos ver que o cópus Constituição L tem cerca de 3 vezes mais palavras, mas tem apenas 30% sentenças a mais distribuídas em ~7.5h de gravação. Isso significa que no cópus da Constituição as sentenças são mais longas e o falante produz enunciados demorados, de maneira cautelosa/planejada. Comparando as duas versões da Constituição, podemos ver que as transcrições do conjunto grande são muito mais esparsas, contendo apenas cerca de 8 sentenças, enquanto a Constituição S contém cerca de 24 sentenças por transcrição. Desse modo, podemos concluir que o cópus Constituição S é mais semelhante ao cópus da Cinderela que o Constituição L em relação à distribuição de limites de sentença.

Sumário dos Cópus

Apresentamos três cópus nesta seção: Cinderela, BALE, e Constituição. Apenas o cópus Cinderela v2 possui anotações de disfluências. Para simplificar a comparação, resumizamos as propriedades de cada cópus na [Tabela 19](#).

Tabela 19 – Sumário dos Cópus utilizados nos experimentos.

Propriedade	Cinderela v2	Constituição S	Constituição L	BALE
# Transcrições	60	357	60	10
# Palavras	23807	63275	30521	1093
# Sentenças	2066	2698	1409	166
# Sent. / # Transc.	34.4	7.6	23.5	16.6
# Palav. / # Sent.	11.6	23.4	21.7	6.6
# Preenchimentos	545	-	-	-
# Disfs. de Edição	1778	-	-	-
Duração	4h 11m	7h 39m	3h 43m	8m

Fonte: Dados da pesquisa.

4.2 Abordagem RCNN para SS e DD

Na seção anterior apresentamos os corpúis utilizados nesta pesquisa. Nesta seção, mostramos como utilizamos esses recursos para ambas as tarefas: Segmentação de Sentenças (SS) e Detecção de Disfluências (DD). Antes de descrever detalhadamente a Rede Neural Recorrente Convolutiva (RCNN) proposta, definimos uma terminologia generalizável da modelagem sequencial para segmentar sentenças e detectar disfluências.

4.2.1 Problemas de Classificação Sequenciais

Modelamos nossas tarefas como problemas de classificação para os quais temos uma entrada de dados representada por uma matriz $\mathbf{X} \in \mathbb{R}^{m \times \varphi \times d}$, em que m é o número de instâncias de treinamento (transcrições), φ é o tamanho relativo a j -ésima instância de treinamento, que é igual ao número de *tokens* da j -ésima transcrição do corpúis, e d corresponde à dimensão do vetor de cada uma das palavras $w \in \mathbf{X}_j$ que são representadas por vetores de *features* lexicais (*embeddings*) ou prosódicas (duração, pausa, *pitch*, energia).

Além da entrada, temos uma saída esperada representada por uma matriz $\mathbf{Y} \in \mathbb{R}^{m \times \varphi \times K}$, sendo K o número de elementos do conjunto de etiquetas $\mathcal{Y} = \{\mathcal{Y}_1, \dots, \mathcal{Y}_K\}$. Por exemplo, para o problema de classificação binária ($K = 2$) de SS, esse conjunto pode ser $\mathcal{Y} = \{NB, B\}$, em que B significa um limite de sentença e NB não. O objetivo de um classificador é aprender um modelo $h_\theta : \mathbf{X} \mapsto \mathbf{Y}$ que mapeia as sequências de instâncias $\mathbf{x} \in \mathbf{X}$ para suas respectivas etiquetas $\mathbf{y} \in \mathbf{Y}$. Para isso acontecer, o modelo h precisa aprender um conjunto de pesos θ usados para minimizar uma função de custo $\mathcal{L}(\theta)$, que mensura o quão errado o modelo classifica as instâncias de entrada. O objetivo do classificador é descobrir qual é a etiqueta $\mathcal{Y}_k$ mais provável para cada instância $\mathbf{x}$:

$$h_\theta(\mathbf{x}) = \left\{ \arg \max_{\mathcal{Y}_k} P(\mathcal{Y}_k | w) \right\}, \quad \forall w \in \mathbf{x} \quad (4.1)$$

Como $\mathbf{x}$ é uma sequência de palavras, a saída da [Equação 4.1](#) é uma sequência de etiquetas previstas $\mathbf{p} \in \mathcal{Y}^\varphi$. Em termos linguísticos, o modelo deve classificar cada palavra $w_{j,i}$ pertencente a uma transcrição $s_j \in \mathcal{D}$ para sua respectiva marcação $y_{j,i} \in \mathcal{Y}$ definida previamente pela anotação do corpúis.

O aprendizado dos pesos θ não é feito sobre o corpúis inteiro $\mathcal{D}$, mas sim sobre uma partição dele, chamada de conjunto de treinamento $\mathcal{T} \subset \mathcal{D}$, sendo a outra partição reservada para o teste $\mathcal{S} \subset \mathcal{D}$, de modo que $\mathcal{T} \cap \mathcal{S} = \emptyset$ e $\mathcal{T} \cup \mathcal{S} = \mathcal{D}$. Portanto, o modelo tenta minimizar a função de custo $\mathcal{L}$ olhando apenas as instâncias de $\mathcal{T}$ para aprender os pesos ideais θ^* :

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta; \mathcal{T}) \quad (4.2)$$

Dessa forma, novas instâncias do teste podem ser previstas usando a [Equação 4.1](#) com os pesos ideais aprendidos de acordo com a [Equação 4.2](#). Uma consideração importante é que, ao

contrário da modelagem normalmente utilizada em problemas de classificação para aprendizado de máquina, esta modelagem trabalha sobre o eixo temporal, isto é, sobre sequências de dados. E, conforme vimos na [Subseção 2.6.2](#), trabalhar com sequências de dados pode trazer benefícios para problemas de PLN.

4.2.2 Modelo Neural Recorrente Convolutacional

O nosso modelo de Redes Neurais Recorrentes Convolucionais (RCNN) utiliza dois tipos de *features*: textuais e prosódicas. As *features* textuais são divididas em etiquetas morfossintáticas e *word embeddings*. Já as *features* prosódicas são divididas em duração, *pitch*, energia e pausa. Praticamente a mesma arquitetura da RCNN pode ser utilizada para SS como para DD, com a exceção do número de classes ($|\mathcal{Y}|$) definidas para cada tarefa, o qual corresponde ao número de neurônios na camada de saída.

As *word embeddings* são treinadas de modo não supervisionado para gerar uma representação na forma de um vetor real multidimensional para cada palavra w no vocabulário $w \in \mathcal{V}$. Podemos ver uma *embedding* como uma função $e_w : \mathcal{V} \mapsto \mathcal{E}$, em que $\mathcal{E}$ é o vocabulário de treinamento das *word embeddings*. De maneira análoga, a palavra w sofre o mesmo comportamento para as etiquetas morfossintáticas, porém ao invés de mapear as palavras para o vocabulário $\mathcal{E}$, elas são mapeadas para o conjunto de etiquetas $\mathcal{C}$ morfossintáticas definidas pelo etiquetador escolhido.

As *features* prosódicas de duração, *pitch*, energia são extraídas para o primeiro, antepenúltimo, penúltimo e último fonema de cada palavra. Já a *feature* de pausa corresponde à duração extraída para o espaço entre duas *palavras*, sendo que quando não há pausa, a duração é igual a zero. Como essas informações prosódicas variam para cada falante, elas são devidamente normalizadas antes de serem utilizadas pelo modelo preditivo.

Para extrair automaticamente novas *features* a partir da entrada e ao mesmo tempo lidar com problemas de longa dependência entre palavras, adotamos um método baseado em RCNN, o qual foi inspirado nos trabalhos de segmentação de sentenças de [Tilk e Alumäe \(2015\)](#) e [Che et al. \(2016\)](#), e também no trabalho de classificação de textos usando RCNN por [Lai et al. \(2015\)](#), para o qual remodelamos de forma que a unidade básica de classificação fosse uma sequência de palavras. A arquitetura da nossa RCNN pode ser vista na [Figura 31](#).

Nosso modelo final consiste na combinação de dois modelos. O primeiro é responsável por tratar apenas informações textuais, enquanto o segundo trata apenas informações prosódicas. Ambos os modelos têm a mesma arquitetura mostrada na [Figura 31](#). Essa estratégia é baseada na ideia de que podemos treinar o modelo textual com ainda mais dados, já que informações textuais são facilmente encontradas na Web. Para obter a classe mais provável $\mathcal{Y}_k$ para uma palavra $w \in \mathbf{x}$, os modelos são combinados usando uma interpolação linear entre eles, sendo que

Figura 31 – Arquitetura da RCNN para o modelo textual e o prosódico.

Fonte: Elaborada pelo autor.

um recebe o complemento ponderado do outro:

$$P(\mathcal{Y}_k | w) = \alpha \cdot P_{text}(\mathcal{Y}_k | w) + (1 - \alpha) \cdot P_{pros}(\mathcal{Y}_k | w) \quad (4.3)$$

Desse modo, basta substituírmos a probabilidade condicional da [Equação 4.1](#) pela mesma probabilidade definida na [Equação 4.3](#) através da interpolação linear entre os modelos que tratam informações textuais P_{text} e prosódicas P_{pros} , os quais serão definidos camada-a-camada a seguir. A interpolação linear é uma abordagem interessante, pois além ser computacionalmente rápida, o valor do α é interpretável, informando a importância do modelo textual em contraste com o prosódico.

4.2.2.1 Camada de Entrada

A entrada de ambos os modelos é uma matriz de tamanho $\varphi \times d$. Mas o conteúdo e o valor de d variam do modelo textual para o prosódico.

No modelo textual, criamos uma matriz de *word embeddings* $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d_e}$, sendo d_e o tamanho dos vetores treinados de modo não-supervisionado, e concatenamos esses vetores com os vetores de *embeddings* de etiquetas morfossintáticas que correspondem a linhas da matriz $\mathbf{C} \in \mathbb{R}^{|\mathcal{C}| \times d_c}$, a qual é inicializada aleatoriamente assumindo valores de uma distribuição gaussiana escalada pela dimensões da matriz de entrada ([GLOROT; BENGIO, 2010](#)), obtendo assim um novo vetor com tamanho $d = d_e + d_c$. Palavras do vocabulário do corpus que não estão no vocabulário de *embeddings* $w_i \in \mathcal{V} \wedge w_i \notin \mathcal{E}$ compartilham um único vetor gerado aleatoriamente, representando uma palavra desconhecida.

O respectivo vetor de *embedding* e_i para a palavra w_i pode ser facilmente buscado na matriz de *embedding* $\mathbf{E}$, realizando a operação da [Equação 4.4](#):

$$E_e(w_i) = e_i = \text{one_hot}(w_i)^\top \cdot \mathbf{E} \quad (4.4)$$

sendo que a função `one_hot` retorna um vetor preenchido por zeros em todas as dimensões exceto na i -ésima, que é preenchido com o escalar 1. É importante notar que a matriz de *word*

embeddings $\mathbf{E}$ é dimensionada pelo tamanho do vocabulário do *córpus* e não do treinamento de *embeddings*³.

Para o modelo prosódico, passamos diretamente as informações de duração, *pitch* e energia dos fones de cada palavra, e também a duração da pausa entre duas palavras. Sendo assim, para o modelo prosódico, cada palavra possui um vetor de tamanho $d = 4 \cdot 3 + 1 = 13$.

4.2.2.2 Camada Convolutacional

A partir desta camada, ambos os modelos comportam-se da mesma maneira. E uma vez que temos a matriz de entrada composta pelas *features* das palavras, a camada convolutacional a recebe, a qual é responsável por extrair n_f *features* novas a partir de uma janela deslizante de tamanho h_c , que corresponde ao tamanho do filtro aplicado ao vetor concatenado $[e_1, \dots, e_{h_c}]$ correspondente a uma região de h_c palavras vizinhas (KIM, 2014). A camada convolutacional produz novas *features* $\mathbf{u}_t$ para cada t -ésima palavra ao aplicar um filtro $\theta^{(c)} \in \mathbb{R}^{n_f \times h_c \cdot d}$ compartilhado para uma janela de h_c *embeddings* $\mathbf{e}_{t-h_c+1:t}$ em uma sentença de tamanho φ :

$$\mathbf{u}_t = \text{ReLU} \left(\theta^{(c)} \mathbf{e}_{t-h_c+1:t} + \mathbf{b}^{(c)} \right), \quad h_c \leq t \leq q_f \quad (4.5)$$

em que $\mathbf{b}^{(c)}$ é o vetor de *bias*, ReLU é uma função de ativação não linear definida na Equação 2.23, e q_f representa a quantidade de filtros.

Nossa camada convolutacional se move em apenas uma dimensão (CNN 1D) verticalmente, um passo por vez, o que resulta numa quantidade de filtros q_f igual a $\varphi - h_c + 2 * p + 1$. E como queremos classificar exatamente φ elementos, adicionamos $p = \lfloor h_c/2 \rfloor$ elementos de *padding* em ambos os lados da sentença.

Em adição, aplicamos uma operação de *max-pooling* no eixo temporal focando numa região de h_m palavras, com o intuito de passar para a próxima camada apenas as *features* mais importantes:

$$\hat{\mathbf{u}}_t = \max \left\{ \mathbf{u}_{t-h_m+1:t} \right\}, \quad h_m \leq t \leq q_f \quad (4.6)$$

sendo que aplicamos o mesmo conceito de *padding* da camada convolutacional para a operação da Equação 4.6.

4.2.2.3 Camada Recorrente Bidirecional

As novas *features* extraídas são passadas para uma camada recorrente bidirecional com n_r unidades. Como vimos na Seção 2.6.2, uma camada recorrente é capaz de armazenar informações passadas ao conectar estados ocultos anteriores com o estado oculto atual em um instante t .

³ É natural usar esse procedimento para a avaliação do modelo, pois o treinamento fica mais rápido, mas quando deseja-se publicar o método como ferramenta, deve-se dimensionar pelo vocabulário de *word embeddings*.

Numa camada recorrente básica, os valores das unidades ocultas $\mathbf{a}_t$ e de saída $\mathbf{u}_t$ são computadas seguindo a Equação 4.8:

$$\mathbf{a}_t = f\left(\theta_u^{(r)} \hat{\mathbf{u}}_t + \theta_a^{(r)} \mathbf{a}_{t-1} + \mathbf{b}_a^{(r)}\right) \quad (4.7)$$

$$\mathbf{u}_t = g\left(\theta_u^{(r)} \mathbf{a}_t + \mathbf{b}_u^{(r)}\right) \quad (4.8)$$

em que f e g são funções de ativação não-lineares, sendo que em nosso método $g = \sigma$ e $f = \tanh$. Entretanto, ao invés de utilizarmos uma camada recorrente com unidades convencionais, utilizamos células de LSTM definidas na Seção 2.6.2 como unidades de ativação, de modo que a saída da camada recorrente torna-se:

$$\mathbf{a}_t = \tanh\left(LSTM(\mathbf{a}_t)\right) \quad (4.9)$$

$$\mathbf{u}_t = \sigma\left(\theta_u^{(r)} \mathbf{a}_t + \mathbf{b}_u^{(r)}\right) \quad (4.10)$$

Assim como em Tilk e Alumäe (2016), a nossa camada recorrente baseia-se em estados anteriores e em estados futuros, mecanismo conhecido como recorrente bidirecional (BRNN). Para implementar o mecanismo bidirecional, realizamos o truque de inverter as sequências antes de passá-las para uma nova camada recorrente LSTM básica. Esse procedimento dobrou a quantidade de pesos, já que a nova camada é independente da outra. A saída desta camada num instante t é a soma entre as saídas da camada recorrente que trata as sequências da esquerda-para-direita $\vec{\mathbf{u}}_t$ com a que trata as sequências da direita-para-esquerda $\overleftarrow{\mathbf{u}}_t$:

$$\mathbf{u}_t = \vec{\mathbf{u}}_t + \overleftarrow{\mathbf{u}}_t \quad (4.11)$$

Com o uso de uma camada bidirecional que trata *features* convolucionadas, a rede é apta a explorar o princípio que afirma que informações de palavras próximas tem uma grande influência, enquanto considera que palavras longes à esquerda e à direita também podem ter algum impacto na classificação. Isso acontece frequentemente para a tarefa de SS, por exemplo, para o caso de sentenças interrogativas definidas pelos pronomes “quem”, “qual”, “quando”.

4.2.2.4 Camada Totalmente Conectada

Após a camada BRNN, usamos uma estratégia de regularização conhecida como *Dropout*, que tenta prevenir a co-adaptação entre unidades ocultas durante o *forward-backpropagation*, sendo que ignoram-se neurônios com o propósito de reduzir a chance de superajustar o modelo (SRIVASTAVA *et al.*, 2014). Levamos em consideração a necessidade de não afetar as conexões recorrentes, preservando a capacidade de modelar sequências eficientemente (PHAM *et al.*, 2014).

Na prática, quando aplicamos a estratégia de *dropout* com probabilidade $p = 0.5$ na BRNN, algumas unidades $\mathbf{u}$ são ignoradas: $\mathbf{u} \in \mathcal{T} = \mathbf{m} \odot \mathbf{u}$, sendo $\odot$ uma operação de multiplicação elementar, e $\mathbf{m}$ é um vetor de tamanho n_r que corresponde a uma máscara binária para cada

unidade de ativação retirada independente de uma distribuição de Bernoulli $\mathbf{m}_i \sim \text{Bernoulli}(p)$. Durante a fase de teste, todas as unidades são ativadas com uma ponderação igual a p : $\mathbf{u} \in \mathcal{S} = p\mathbf{u}$.

A última camada, então, recebe a saída da BRNN a cada instante t , e a passa por uma camada totalmente conectada básica, de modo a produzir previsões usando a função de ativação *softmax* (Equação 2.25), que nos dá a probabilidade final de uma palavra preceder um limite de sentença (B) ou não (NB):

$$\mathbf{p}_w = P(\mathcal{Y}_k | w) = \text{softmax}\left(\boldsymbol{\theta}^{(f)}\mathbf{u}_t + \mathbf{b}^{(f)}\right) \quad (4.12)$$

em que $\boldsymbol{\theta}^{(f)} \in \mathbb{R}^{n_r \times K}$ é a matriz de pesos compartilhada em cada instante t , e $\mathbf{b}^{(f)} \in \mathbb{R}^{n_r}$ é o vetor compartilhado para os temas de *bias*.

4.2.2.5 Treinamento

Definimos todos os parâmetros a serem treinados como o conjunto de pesos Θ :

$$\Theta = \{\mathbf{E}, \mathbf{C}, \boldsymbol{\theta}^{(c)}, \mathbf{b}^{(c)}, \overrightarrow{\boldsymbol{\theta}}^{(r)}, \overrightarrow{\mathbf{b}}^{(r)}, \overleftarrow{\boldsymbol{\theta}}^{(r)}, \overleftarrow{\mathbf{b}}^{(r)}, \boldsymbol{\theta}^{(f)}, \mathbf{b}^{(f)}\} \quad (4.13)$$

em que $\mathbf{E}$ é a matriz de *embeddings* para as palavras, $\mathbf{C}$ é a matriz de *embeddings* para as etiquetas morfossintáticas. Já $\boldsymbol{\theta}^{(l)}$ e $\mathbf{b}^{(l)}$ são, respectivamente, a matriz de pesos e vetor de *bias* da camada $l \in \{c, r, f\}$.

Dividimos os pesos da BRNN para a camada recorrente esquerda-para-direita $\overrightarrow{\boldsymbol{\theta}}^{(r)}, \overrightarrow{\mathbf{b}}^{(r)}$, e para a camada recorrente direita-para-esquerda $\overleftarrow{\boldsymbol{\theta}}^{(r)}, \overleftarrow{\mathbf{b}}^{(r)}$. Mais especificamente, o conjunto de pesos dessas duas camadas incluem os seguintes conjuntos de pesos definidos nas células de LSTM: os pesos $\boldsymbol{\theta}_{\hat{\mathbf{u}}} \in \mathbb{R}^{n_r \times n_f}$ e vetor de *bias* $\mathbf{b}_{\hat{\mathbf{u}}} \in \mathbb{R}^{n_r}$ da entrada $\hat{\mathbf{u}}$; os pesos $\boldsymbol{\theta}_a \in \mathbb{R}^{n_r \times n_r}$ e *bias* $\mathbf{b}_a \in \mathbb{R}^{n_r}$ dos estados ocultos recorrentes $\mathbf{a}$ para todos os *gates* (i, f, o, g). Além disso, temos a matriz de pesos para a ativação de saída $\mathbf{u}$ de cada camada recorrente $\boldsymbol{\theta}_u \in \mathbb{R}^{n_r \times n_r}$ e seus *bias* $\mathbf{b}_u \in \mathbb{R}^{n_r}$.

Definimos a função de custo $\mathcal{L}$ como sendo a entropia cruzada entre a distribuição da probabilidade da sequência de etiquetas previstas $\mathbf{p}$, obtidas para todas as palavras da sentença de acordo com a Equação 4.12, e a sequência de etiquetas originais $\mathbf{y}$:

$$\mathcal{L}(\mathbf{y}, \mathbf{p} | \boldsymbol{\theta}) = - \sum_i \mathbf{y}_i \cdot \log(\mathbf{p}_i) \cdot \omega_{\mathbf{y}_i} \quad (4.14)$$

Como SS e DD são problemas de classificação desbalanceados (e.g. $\#NB \gg \#B$), atribuímos diferentes pesos para cada etiqueta ω_ℓ na Equação 4.14, de modo que os pesos das classes minoritárias ($\ell = B$) sejam maiores que os das majoritárias ($\ell = NB$), calculados de acordo com a Equação 4.15:

$$\omega_\ell = \frac{|\mathbf{Y}|}{2 \cdot |\mathbf{Y} = \ell|} \quad (4.15)$$

Minimizamos a função de custo em relação a todos os pesos $\Theta \mapsto \mathcal{L}$ usando o procedimento RMSProp (TIELEMAN; HINTON, 2012) com *backpropagation* para computar os gradientes $\nabla \mathcal{L}$, no qual a rotina de atualização dos pesos Θ da Equação 2.27 é modificada, de modo que para cada instante t os gradientes são normalizados por uma média móvel exponencial r_t da magnitude do gradiente:

$$r_t = \gamma \cdot r_{t-1} + (1 - \gamma) \cdot \nabla \mathcal{L}(\Theta_t)^2 \quad (4.16)$$

$$\Theta_{t+1} = \Theta_t - \eta \frac{\nabla \mathcal{L}(\Theta_t)}{\sqrt{r_t} + \varepsilon} \quad (4.17)$$

em que η é a taxa de aprendizado, $0 < \gamma < 1$ é o fator de esquecimento, e $\varepsilon = 10^{-6}$ é um valor pequeno para estabilidade numérica.

4.2.3 Metodologia de Avaliação

Como pré-processamento, separamos os *tokens* delimitados por espaços e não removemos *stopwords* das transcrições, já que elas podem ser *features* importantes em nosso domínio para todas as tarefas.

As etiquetas morfossintáticas foram extraídas com a ferramenta *nlpnet*⁴ treinada numa versão revisada do corpus Mac-Morpho (FONSECA; ROSA; ALUÍSIO, 2015a), o qual contém um conjunto de 25 etiquetas. Já as *word embeddings* foram primeiramente avaliadas de acordo com o número de dimensões e com a técnica usada para induzi-las, e então a melhor configuração foi mantida para todas as avaliações seguintes de cada tarefa.

As informações prosódicas (*pitch*, energia e duração) foram extraídas em nível de fones a partir das gravações usando o PRAAT (BOERSMA *et al.*, 2002) para cada tarefa. Como as transcrições foram feitas manualmente, precisamos realizar um procedimento de alinhamento forçado entre o áudio e o texto para conseguir extrair essas informações. Esse alinhamento foi feito usando o HTL Toolkit (YOUNG *et al.*, 2002) com um corpus de fala limpa e um dicionário de pronúncia transcrito foneticamente pelo sistema Petrus (SERRANI, 2015), aumentado via regras manuais de modo a inserir algumas pronúncias, resultando num modelo apropriado para reconhecimento automático de voz.

Como nossas tarefas são bastante desbalanceadas, se o método classificasse todas as palavras como sendo da classe majoritária, obteríamos uma acurácia superior a 90%. Por essa razão, usamos as métricas de F_1 e SER para SS (Subseção 2.1.4), e F_1 para DD (Subseção 2.2.2).

Avaliamos nossas tarefas de maneira intrínseca para cada tarefa, com a inclusão de uma avaliação cruzada de benchmarks (*cross-benchmark evaluation*) para a tarefa de SS. Como temos poucos dados, para avaliar intrinsecamente adotamos uma estratégia de validação cruzada para o

⁴ <<http://nilc.icmc.usp.br/nlpnet/>>

grupo sendo analisado (CTL, CCL ou DA). Verificamos com base na baixa variância ($\pm 1\%$) entre as medidas dos *folds* que dividir o *cópus* em $k = 5$ partições é uma escolha aceitável. Conforme discutido na Subseção 2.1.4, usamos a média micro entre os *folds*, ou seja, acumulamos os erros e acertos durante os *folds* e calculamos a métrica no final. Para avaliação cruzada de benchmarks, realizamos testes de robustez com outros *cópus* de fala transcritos.

Já para a análise extrínseca, avaliamos o DeepBonDD através de métricas sintáticas extraídas pelo Coh-Metrix-Dementia⁵. A avaliação extrínseca é importante, pois com ela podemos avaliar diretamente o impacto das tarefas abordadas nesta dissertação na próxima tarefa subsequente do projeto, a detecção automática de demências, especificamente, a extração de features para a classificação automática das narrativas nas classes CCL, DA e Controle.

⁵ <<http://nilc.icmc.usp.br/coh-metrix-dementia/>>

AVALIAÇÕES PARA SEGMENTAÇÃO DE SENTENÇAS

Realizamos avaliações para a tarefa de SS usando nosso modelo de RCNN em três corpú: Cinderela v2, Constituição e, usando avaliação cruzada de benchmarks (ou testes de robustez) no corpú BALE.

Para as *word embeddings*, primeiramente realizamos uma análise dos algoritmos de treinamento não supervisionado e do número de dimensões das *embeddings* na [Seção 5.2](#), e então usamos a RCNN com essas *embeddings* para realizar outras avaliações, como em relação a importância das *features* ([Seção 5.3](#)), comparação de classificadores ([Seção 5.4](#)), e avaliação cruzada de benchmarks ([Seção 5.5](#)).

Avaliamos também nosso conjunto de dados com algumas *baselines* na [Seção 5.1](#). E, finalmente, na [Seção 5.6](#), realizamos uma análise dos erros cometidos pela RCNN com o propósito de verificar se o modelo é capaz de aprender alguma estrutura semântica, sintática e morfológica.

Em geral, todos os experimentos usando a RCNN foram realizados com os hiperparâmetros mostrados na [Tabela 20](#), mas para experimentos específicos, variamos alguns hiperparâmetros com o objetivo de analisar o impacto deles.

5.1 *Baselines*

Os *baselines* computados nessa pesquisa são mais fortes do que o *baseline* que simplesmente classifica todas as instâncias como a classe majoritária (o qual reportamos como *chance baseline*). Mais especificamente, os *baselines* reportados são:

- **Dicas de pausas:** É bastante consolidado na literatura que a pausa na fala geralmente

Tabela 20 – Hiper-parâmetros da RCNN.

Var.	Hiper-parâmetro	Textual	Prosódico
d_e	Tam. <i>emb.</i> de palavras	600	-
d_c	Tam. <i>emb.</i> de etiquetas	10	-
n_f	Número de filtros	100	32
h_c	Tamanho do filtro	7	5
h_m	Tamanho do <i>max-pooling</i>	3	3
n_r	Unidades recorrentes	100	100
p	Taxa de <i>dropout</i>	0.5	0.5
γ	Fator de esquecimento	0.9	0.9
η	Taxa de aprendizado	0.001	0.001

Fonte: Elaborada pelo autor.

indica o fim de um enunciado. Sendo assim, resolvemos verificar os resultados obtidos prevendo limites de sentença em locais cujas durações das pausas (períodos de silêncio no áudio) eram maiores que um limiar de p segundos.

- **Dicas lexicais:** Verificando as transcrições do corpus desta pesquisa (Cinderela v2), percebemos que a maioria das sentenças começam com a palavra “e”, e terminam com a palavra “né”. Então resolvemos verificar os resultados obtidos prevendo limites de sentença em locais em que essas dicas lexicais ocorreram exclusivamente.

Os resultados dessas *baselines* são apresentados na [Tabela 21](#) em termos de F_1 .

Tabela 21 – Resultados das *baselines* em termos de F_1 .

<i>Baseline</i>	Cinderela v2			BALE		Constituição	
	CTL	CCL	DA	CTL	CCL	L	S
Chance	0.15	0.15	0.18	0.28	0.25	0.08	0.09
Pausa ($p = 0.5s$)	0.22	0.26	-	0.33	0.30	0.19	0.07
Pausa ($p = 1s$)	0.10	0.16	-	0.27	0.22	0.22	0.08
Pausa ($p = 2s$)	0.06	0.07	-	0.17	0.18	0.23	0.08
Lexical (“né”)	0.17	0.21	0.25	0.20	0.08	0.00	0.00
Lexical (“e”)	0.38	0.38	0.27	0.53	0.20	0.01	0.02

Fonte: Elaborada pelo autor.

E os resultados das *baselines* em termos de *SER* são apresentados na [Tabela 22](#).

Deixamos os melhores resultados para cada corpus em negrito. É importante salientar que ao contrário do F_1 , quanto menor o valor de *SER*, melhor é o resultado.

Primeiramente, podemos ver na [Tabela 21](#) que as informações lexicais são mais úteis para os corpus Cinderela e BALE, e informações de pausa são mais úteis para o corpus Constituição. A Constituição, composta de fala preparada, teve resultados muito ruins para as informações lexicais porque, de fato, a dica lexical “né” não ocorreu no corpus, e a dica “e” ocorreu algumas

Tabela 22 – Resultados dos *baselines* em termos de *SER*.

<i>Baseline</i>	<i>Cinderela v2</i>			<i>BALE</i>		<i>Constituição</i>	
	CTL	CCL	DA	CTL	CCL	L	S
Chance	11.38	11.79	8.83	5.27	6.09	22.45	20.66
Pausa ($p = 0.5s$)	1.13	1.29	-	0.87	1.16	1.14	1.27
Pausa ($p = 1s$)	1.01	1.03	-	0.88	1.03	0.93	1.03
Pausa ($p = 2s$)	0.98	0.98	-	0.92	0.94	0.87	0.96
Lexical (“né”)	0.96	0.93	0.87	0.89	0.99	1.00	1.00
Lexical (“e”)	1.02	1.00	1.01	0.67	0.93	2.41	2.34

Fonte: Elaborada pelo autor.

vezes, mas em poucos casos veio depois de um limite de sentença. Para o BALE os resultados usando a dica lexical “e” foi alta comparada aos outros, e isso deve-se principalmente por ser um corpus muito pequeno e essa dica ser produzida com frequência por pacientes de controle. Entretanto, os resultados para todos os corpus são baixos quando comparamos com os resultados de trabalhos relacionados.

A análise por *SER* é muito mais punitiva, pois não leva em consideração os acertos, apenas os erros cometidos. Desse modo, podemos ver uma discrepância entre o resultado das dicas lexicais entre as duas métricas. Em termos de *SER*, obteve-se melhores resultados com a dica lexical “né” precedendo o ponto final da sentença, enquanto analisando via F_1 melhores resultados foram para a dica “e”. Isso ocorreu pelo fato da palavra “e” ocorrer frequentemente no corpus como evidências falseáveis, enquanto o “né” ocorreu muito pouco como falso positivo/negativo.

Em geral, como esperado, os corpus de fala comprometida obtiveram melhores resultados com as *baselines* de dicas lexicais, enquanto o corpus Constituição, de fala preparada, obteve melhores resultados usando informações prosódicas. Entretanto, devemos ressaltar que analisar a Constituição com as dicas lexicais “né” e “e” não é muito justo, pois essas dicas não são frequentes neste corpus. De qualquer modo, quando testamos as mesmos *baselines* com os respectivos verdadeiros negativos da Constituição na própria Constituição, os resultados lexicais melhoraram um pouco apenas, sendo que as informações prosódicas continuaram sendo mais discriminativas.

Nas próximas tabelas de resultados, apresentamos os melhores resultados em negrito, independente da *baseline*, para cada corpus e para ambas as métricas.

5.2 Avaliação de Word Embeddings

Após verificarmos com as *baselines* que as informações lexicais são importantes para segmentar fala comprometida, resolvemos avaliar diferentes *word embeddings*, que é a principal

*feature lexical*¹ usada em nosso modelo da RCNN.

A avaliação das *embeddings* foi realizada com as transcrições do corpus Cinderela v2, por meio da variação de técnicas/algoritmos de indução das *embeddings* das palavras e pela variação da dimensão dessas *embeddings* induzidas. Descrevemos cada uma das técnicas de indução de *embeddings* na Subseção 2.4.1.

O corpus utilizado para a indução desses vetores é composto por textos da Wikipédia de 2016, de um *crawling* feito sobre artigos de notícias do portal G1² e do corpus PLN-BR (BRUCKSCHEN *et al.*, 2008).

Realizamos alguns procedimentos de pré-processamento simples nesse corpus, sendo que deixamos todas as palavras em caixa baixa e separamos os *tokens* de sinais de pontuações de modo a deixar um espaço em branco entre cada *token*. Após esses procedimentos, o corpus resultante para a indução das *embeddings* retornou ~356M *tokens* dentre os quais ~1.7M eram distintos um do outro.

Os resultados em termos de F_1 do desempenho da RCNN sobre cada tipo de paciente são mostradas na Figura 32, para as quais variamos a dimensão das *embeddings* entre os valores $d_e \in \{50, 100, 300, 600\}$.

Na maioria dos casos, o *word2vec* desempenhou melhor que as outras técnicas. Mais especificamente, para CTLs o melhor resultado foi do *word2vec* com Skip-gram e do *fastText* com CBOW, com um F_1 de 0.77. Já para os pacientes com CCL, o *wang2vec* com Skip-gram foi o melhor com um F_1 de 0.74. E para as transcrições de DA a melhor técnica foi o *word2vec* com o algoritmo CBOW, retornando um F_1 de 0.66.

Como a métrica *SER* é muito mais punitiva, decidimos plotar o desempenho da RCNN usando essa métrica sobre cada tipo de paciente na Figura 33, sendo que variamos a dimensão das *embeddings* entre os valores $d_e \in \{50, 100, 300, 600\}$.

Tanto para CTLs quanto para CCLs os melhores resultados foram obtidos com a técnica *wang2vec*, com um $SER = 0.49$ para CTL, e $SER = 0.49$ para CCL usando o algoritmo Skip-gram, sendo que os melhores resultados para CTLs e DAs ($SER = 0.57$) foram obtidos com 300 dimensões do vetor, mas para CCLs o melhor resultado foi usando vetores de 600 dimensões.

Esses resultados nos mostram que somente usando *embeddings*, nossa arquitetura RCNN proposta conseguiu melhores resultados do que o estado-da-arte em termos de F_1 , publicados em Treviso, Shulby e Aluísio (2017) para ambas as classes: CTL e CCL usando *embeddings* com 50 dimensões. Provavelmente devido ao tamanho das *embeddings* usadas, visto que nas figuras é possível observar, na maioria dos casos, um crescimento de F_1 e uma diminuição de *SER* até 300 dimensões.

¹ Embora uma *embedding* tenha capacidade de capturar informações semânticas, sintáticas e morfossintáticas, ela é usada como uma representação do *token lexical*.

² <<http://g1.globo.com/>>

Figura 32 – F_1 para variação de *embeddings* no corpus Cinderela v2.

Fonte: Elaborada pelo autor.

Nosso objetivo com esse experimento foi o de verificar qual tipo de *embeddings* desempenha melhor para a tarefa de SS: as que capturam similaridades semânticas (*word2vec*); sintáticas (*wang2vec*); ou morfológicas (*fastText*). Entretanto, não conseguimos discernir com precisão qual é o tipo mais discriminativo de modo geral, pois as diferenças de um para o outro foram muito pequenas. E, por mais que uma técnica tenha superado a outra num conjunto específico, teríamos ainda que investigar se foi devido à natureza da técnica ou devido a condições secundárias, como hiperparâmetros ou inicialização aleatória. Uma hipótese que tínhamos era que o *fastText* superaria as outras técnicas, pois ele tem a capacidade de capturar similaridades morfológicas das palavras, de modo a aproximar palavras fora do vocabulário daquelas lexicalmente parecidas com elas.

Se tirarmos a média de todas as combinações de técnicas (*word2vec*, *wang2vec*, *fastText*) e algoritmos (CBOW, Skip-gram) para os três tipos de pacientes, a combinação que retorna maior F_1 e menor SER é o *word2vec* com Skip-gram usando 600 dimensões. E então, foi esse par

Figura 33 – *SER* para variação de *embeddings* no corpus Cinderela v2.

Fonte: Elaborada pelo autor.

que usamos nos próximos experimentos. A escolha de 600 dimensões ao invés de 300 baseou-se no fato de que, apesar de resultados em termos de *SER* mostrarem que para maioria dos tipos de transcrições os melhores resultados foram obtidos com 300 dimensões, analisando com a métrica F_1 nos diz que 600 é melhor, e como queremos comparar com trabalhos relacionados na literatura que publicaram em termos de F_1 , decidimos tentar maximizar essa métrica.

5.3 Importância das Features

Com o cenário de avaliação definido para *embeddings* de 600 dimensões induzidas pela técnica *word2vec* com o algoritmo Skip-gram, decidimos avaliar nosso modelo em relação às outras *features*: morfosintáticas e prosódicas. Avaliamos a nossa arquitetura RCNN proposta usando o corpus Cinderela v2 e Constituição, e excluimos o BALE porque ele é um corpus muito pequeno.

A partir deste experimento, usamos informações prosódicas, então deixamos de avaliar o modelo com as transcrições de DA, pois infelizmente não conseguimos realizar o alinhamento forçado entre as transcrições e o áudio. A razão para a exclusão foi pelos áudios dos participantes com DA estarem com qualidade muito baixa, de modo que não era possível identificar o que o participante estava falando dado o volume da voz dos participantes e a presença de ruídos de fundo (as gravações foram feitas em um hospital).

A [Tabela 23](#) sumariza os resultados da RCNN em termos de F_1 para as transcrições de CTLs e CCLs, e para os dois subconjuntos da Constituição: subconjunto pequeno (S) com tamanho próximo ao da Cinderela v2, e o conjunto inteiro (L).

Tabela 23 – Resultados de cada conjunto de *features* para Cinderela v2 e Constituição em termos de F_1 .

Features	Cinderela v2						Constituição					
	CTL			CCL			L			S		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Baseline	0.48	0.32	0.38	0.50	0.31	0.38	0.99	0.13	0.23	0.94	0.04	0.08
PoS	0.39	0.75	0.51	0.37	0.75	0.49	0.30	0.89	0.44	0.29	0.79	0.42
Prosódia	0.14	0.56	0.21	0.18	0.39	0.24	0.56	0.87	0.68	0.49	0.85	0.62
Embeddings	0.73	0.77	0.75	0.71	0.75	0.73	0.60	0.63	0.63	0.60	0.64	0.62
PoS + Emb.	0.73	0.80	0.76	0.71	0.75	0.73	0.64	0.72	0.68	0.63	0.67	0.65
PoS + Pros.	0.39	0.75	0.51	0.37	0.75	0.49	0.52	0.91	0.66	0.57	0.85	0.68
Emb. + Pros.	0.73	0.77	0.75	0.71	0.75	0.73	0.71	0.83	0.76	0.74	0.81	0.77
Todas	0.74	0.81	0.77	0.73	0.75	0.74	0.77	0.82	0.79	0.76	0.85	0.80

Fonte: Elaborada pelo autor.

PoS corresponde às etiquetas morfossintáticas (*Part-of-Speech Tags*).

Na [Tabela 23](#) podemos ver que a RCNN apresenta melhores resultados para a Constituição do que para Cinderela v2. Isso pode estar relacionado com a qualidade do texto, já que as transcrições da Cinderela possuem várias disfluências, que são características da fala espontânea comprometida. Como esperado, os resultados para CTLs foram maiores do que para CCLs, já que narrativas de pacientes de CTL não são potencialmente comprometidas por problemas cognitivos que afetam a fala. Outra observação importante é que nosso método supera a *baseline* por uma margem larga.

Os resultados da Constituição nos mostram que técnicas tradicionais de aprendizado de máquina usadas em PLN podem ser aplicadas em nosso cenário, já que a diferença para as transcrições da Cinderela é pequena. Outra razão que suporta essa afirmação é que resultados de F_1 de trabalhos relacionados para SS fala espontânea/preparada e não comprometida estão no intervalo de ~0.7 e ~0.8 ([SHRIBERG et al., 2009](#); [WANG; NG; SIM, 2012](#); [KHOMITSEVICH et al., 2015](#); [TILK; ALUMÄE, 2015](#); [TILK; ALUMÄE, 2016](#)). Quando comparamos a relação de tamanho da Constituição, descobrimos que o tamanho do corpus não afetou muito os resultados, já que a Constituição S foi levemente melhor que a Constituição L. Acreditamos que, até mesmo com poucos dados, nosso método consegue ter um melhor desempenho na Constituição S por causa da distribuição do tamanho das sentenças no corpus, sendo que a Constituição S tem em

média 23.5 sentenças por transcrição, enquanto a Constituição L tem apenas 7.6.

Analisando a importância dos diferentes conjuntos de *features*, percebemos que *embeddings* tem um grande impacto na classificação em ambos os corpú. Informações de etiquetas morfossintáticas (PoS) também foram decisivas em ambos os corpú, mas por uma margem pequena, já que tem uma pequena diferença de quando utilizado *embeddings* (0.01) na Cinderela v2, e (0.03) na Constituição. Isso nos mostra que as *embeddings* sozinhas já conseguem trazer informações morfossintáticas suficientes. É evidente que o peso para as *features* prosódicas é maior para Constituição, que é composta de fala preparada em comparação com o corpú Cinderela v2, que é fala comprometida. Esse resultado é consistente com os encontrados em Kolár, Liu e Shriberg (2009), Fraser *et al.* (2015), e também em Treviso, Shulby e Aluísio (2017).

Além disso, acreditamos que a qualidade das gravações podem ter influenciado no peso dado para as informações prosódicas, já que o corpú Constituição foi gravado por especialistas em processamento de fala e em um estúdio, já o corpú Cinderela v2 foi gravado no cenário clínico, em um hospital. Apesar disso, podemos ver que nosso método tem um melhor desempenho em todos os corpú quando todas as *features* são usadas. Aliás, os melhores resultados foram obtidos usando $\alpha = 0.8$ da interpolação linear da Equação 4.3 para a Cinderela v2, e $\alpha = 0.55$ para a Constituição, indicando que nosso modelo dá mais peso para o modelo textual.

Conforme vimos nas *baselines*, a métrica *SER* é muito mais punitiva que a F_1 . Devido a isso, decidimos buscar as classificações realizadas pelo nosso método e extrair a métrica *SER*, através da comparação dessas predições com o conjunto de etiquetas originais. Entretanto, o modo como modelamos a ferramenta e salvamos os dados nos permite reportar resultados apenas para a combinação com informações prosódicas, já que tratamos PoS e *embeddings* em conjunto em uma mesma RCNN. A Tabela 24 mostra o desempenho da RCNN com informações textuais e prosódicas em termos de *SER* apenas para as três últimas linhas da Tabela 23 do corpú Cinderela v2. Em adição, colocamos uma linha avaliando apenas as *embeddings* sozinhas (que são iguais aos resultados reportados na Seção 5.2 para o *word2vec* com Skip-gram).

Tabela 24 – Resultados para conjuntos de *features* para Cinderela v2 em termos de *SER*.

<i>Features</i>	Cinderela v2	
	CTL	CCL
Baseline	0.96	0.93
Embeddings	0.50	0.56
PoS + Pros.	1.44	1.53
Emb. + Pros.	0.98	0.99
Todas	0.51	0.58

Fonte: Elaborada pelo autor.

PoS corresponde às etiquetas morfossintáticas (*Part-of-Speech Tags*).

Olhando para os resultados da Tabela 24, podemos ver que usando *embeddings* sozinha

retornou os melhores resultados. Uma possível explicação para isso é que as informações prosódicas podem ter atrapalhado a classificação, aumentando o número de falsos positivos. Entretanto, a diferença absoluta entre usar *embeddings* sozinhas e todas as *features* também foi baixa em termos de F_1 , sugerindo que, de qualquer modo, as *embeddings* são bastante influentes para a tarefa.

Entretanto, não temos uma conclusão clara do potencial das informações prosódicas. Um trabalho futuro planejado para a nossa pesquisa consiste em verificar qual das peculiaridades seguintes podem ter influenciado mais:

1. A qualidade de gravação do áudio é muito ruim, e portanto as *features* extraídas não são discriminativas o suficiente. Para testar essa hipótese, podemos analisar os efeitos da prosódia em termos de *SER* para a Constituição - e como vimos em termos de F_1 , isso é muito provável que seja verdade;
2. A natureza inerente dos pacientes faz com que informações prosódicas não possam ser utilizadas eficientemente. Essa hipótese já sabemos que é verdade através das diferenças dos resultados entre pacientes de CTL e CCL. Entretanto, gostaríamos de descobrir se isso está relacionado à configuração clínica das gravações e das estatísticas demográficas dos pacientes (a maioria são idosos). Para testar essa hipótese, podemos usar o *cópus* do Museu da Pessoa, que também é de fala espontânea de narrativas, e comparar os resultados diretamente.
3. As *features* prosódicas extraídas não são suficientemente discriminativas. Extraímos poucas *features* com a esperança de que a rede conseguisse aprender relação importantes entre *pitch*, energia e duração dos fones, de modo que não fosse preciso extrair várias *features* manualmente através de passos de pré-processamento. Podemos testar essa hipótese ao replicar o conjunto de informações prosódicas mostrada por Shriberg *et al.* (2000), que foram amplamente utilizadas em trabalhos de SS (LIU *et al.*, 2006; GÜZ *et al.*, 2007; SHRIBERG *et al.*, 2009; BATISTA *et al.*, 2012), ou até replicar o uso das mesmas *features* acústicas utilizadas por Huang *et al.* (2014).

Quando comparamos nossos resultados medidos em *SER* com outros da literatura para a tarefa de SS, temos a impressão que os resultados da nossa RCNN não foram muito bons. Por exemplo, Stolcke e Shriberg (1996) conseguiu segmentar sentenças do Switchboard v1 com um $SER \approx 0.40$, e mais recentemente, Huang *et al.* (2014) conseguiu um *SER* de 0.30 para um *cópus* de programas de notícias jornalísticas (fala preparada) da série NIST-RT 2004. Entretanto, quando comparamos nossos resultados para saídas de sistemas de RAV para o mesmo trabalho de Huang *et al.* (2014), observamos que nosso resultado para pacientes de controle é equiparável, já que eles conseguiram um $SER = 0.53$ nessa avaliação. Além disso, no trabalho para o português europeu de Batista *et al.* (2012), os autores conseguiram um $SER = 0.60$ com o uso de *features*

prosódicas e lexicais para um corpus de fala preparada. Portanto, esses casos indicam que nossos resultados em termos de *SER* são bons também, já que estamos trabalhando com fala espontânea e comprometida, que naturalmente contém mais disfluências.

5.4 Comparação de Métodos

Para comparar nosso modelo de RCNN com trabalhos relacionados nós replicamos a abordagem usada por [Fraser et al. \(2015\)](#), a qual usa um modelo de CRF de cadeia linear para SS de fala comprometida para a língua inglesa. Detalhamos resumidamente esse trabalho na [Subsubseção 3.1.3.1](#). Além disso, para explicar nossa escolha por um modelo recorrente bidirecional, nós separamos nosso método em outros três:

- **MLP:** Removemos as camadas convolucionais e recorrentes, e adicionamos uma camada totalmente conectada oculta, de modo a formar uma MLP com uma camada oculta com 100 unidades de ativação. Na nossa arquitetura, isso pode ser visto como uma convolução com função de ativação linear que não possui pesos associados, de modo que o número de filtros seja igual ao tamanho da janela dos vetores de entrada concatenados $d \times h_c$. Esse modelo é similar, mas não exatamente o mesmo, que o usado por [Xu et al. \(2014\)](#).
- **RNN:** Removemos a camada de convolução, de modo que passamos a entrada diretamente para a camada recorrente bidirecional composta por 100 células de LSTM. Esse modelo é similar, mas não exatamente o mesmo, que o usado por [Tilk e Alumäe \(2015\)](#).
- **CNN:** De modo análogo ao modelo de RNN, removemos a camada recorrente bidirecional e passamos a saída da camada convolucional direto para a camada de saída. Esse modelo é similar, mas não exatamente o mesmo, que o usado por [Che et al. \(2016\)](#).

Os resultados usando a combinação de todas as *features* para cada método são apresentados na [Tabela 25](#).

Tabela 25 – Resultados em termos de F_1 para cada método.

Método	Cinderela v2				Constituição	
	CTL		CCL		L	S
	F_1	<i>SER</i>	F_1	<i>SER</i>	F_1	F_1
Baseline	0.38	0.96	0.38	0.93	0.23	0.08
CRF	0.62	0.72	0.61	0.68	0.51	0.48
MLP	0.71	0.67	0.67	0.72	0.77	0.78
RNN	0.41	2.04	0.41	2.14	0.58	0.57
CNN	0.73	0.69	0.70	0.65	0.73	0.70
RCNN	0.77	0.51	0.74	0.58	0.79	0.80

Fonte: Elaborada pelo autor.

Nosso método conseguiu os melhores resultados em ambos os corpú. Podemos ver que o CRF, usado por [Fraser et al. \(2015\)](#) obteve os piores resultados no corpú de fala preparada da Constituição, e foi melhor apenas que a RNN nos dados da Cinderela v2. Esses resultados são similares aos reportados no artigo deles, o que sugere que nossa replicação foi fiel. Apesar da nossa RCNN ter um conjunto maior de pesos que a RNN, acreditamos que a RNN teve um baixo desempenho devido o grande conjunto de pesos das células LSTM. Isso pode estar relacionado com o fato de que essas células são unidades complexas que precisam de mais dados para ajustar o modelo. E sem o auxílio de uma camada convolucional, a RNN falhou em conseguir bons resultados. Outra hipótese é de que o tamanho das *embeddings* seja muito grande, aliado a poucos dados. Olhando para os resultados da Constituição, podemos ver que até mesmo a RNN desempenhou melhor nesse corpú do que no Cinderela v2.

Em geral, a MLP e a CNN conseguiram atingir resultados melhores que os métodos de RNN e CRF. O resultado da MLP para o subconjunto de pacientes com CCL não foi tão bom quanto ao da CNN, o que indica, de certo modo, que um modelo neural simples com camadas totalmente conectadas não é capaz de lidar bem com narrativas potencialmente comprometidas. Em contrapartida, para os dados da Constituição, a MLP obteve resultados bem próximos (~ 0.02) do método com melhores resultados.

Por fim, a nossa RCNN atingiu os melhores resultados em ambos os corpú, implicando que a união de um modelo convolucional com um modelo recorrente foi uma boa escolha para lidar com fala comprometida. Além disso, acreditamos que a maior influência foi da camada convolucional, já a adição de uma camada recorrente com LSTM foi, potencialmente, capaz de lidar com alguns casos particulares, provavelmente de longa dependência, similares as descobertas encontradas em [Tilk e Alumäe \(2015\)](#). A CNN sozinha não é capaz de lidar com essas peculiaridades devido ao tamanho fixo do filtro no processo de convolução, um resultado também notado por [Che et al. \(2016\)](#).

5.5 Testes de Robustez

O termo “robustez” é usado aqui para indicar o quão bem nosso método desempenha para dados de teste que variam do conjunto usado para treinamento. Não realizamos validação cruzada nos testes de robustez, então decidimos tirar a média de 5 execuções para termos uma avaliação mais justa.

Foi realizado um teste de robustez ao treinar com o conjunto inteiro da Cinderela v2 e testar no corpú BALE, que também é do mesmo gênero que o nosso: composto por transcrições de narrativas de fala comprometida. A diferença entre eles é a estória utilizada da narrativa. Além disso, outro teste de robustez foi realizado para dados fora do nosso gênero: treinamos com a Constituição inteira (L) e testamos com cada subconjunto de pacientes do corpú Cinderela v2. Os resultados utilizando o modelo da RCNN são mostrados na [Tabela 26](#).

Tabela 26 – Resultados para os testes de robustez.

Córpus: Treino $\mapsto$ Teste	CTL				CCL			
	P	R	F_1	SER	P	R	F_1	SER
Baseline Cinderela v2	0.48	0.32	0.38	0.96	0.50	0.31	0.38	0.93
Constituição $\mapsto$ Cinderela v2	0.18	0.32	0.23	1.88	0.19	0.33	0.24	2.10
Baseline BALE	0.88	0.38	0.53	0.67	0.38	0.25	0.30	0.93
Cinderela v2 $\mapsto$ BALE	0.78	0.61	0.69	0.56	0.80	0.57	0.67	0.65

Fonte: Elaborada pelo autor.

Essas avaliações mostram que a RCNN não obteve bom desempenho quando treinamos na Constituição, em que a qualidade da prosódia é melhor, e testamos na Cinderela v2 (fala espontânea/comprometida), de modo que nosso método não conseguiu atingir a *baseline*. O baixo desempenho está muito provavelmente relacionado a baixa frequência de dicas lexicais entre os dois corpú. Por exemplo, vimos na [Tabela 21](#) que a dica lexical “né” não ocorre no corpú da Constituição, já para o corpú da Cinderela v2 ocorre frequentemente precedendo um limite de sentença. Essas diferenças de dicas lexicais se deve, muito provavelmente, à desigualdade entre o tipo de fala entre os corpú: preparada vs. espontânea/comprometida.

Quando mantemos o gênero do corpú mas mudamos a estória dos testes neuropsicológicos, nossa RCNN ainda consegue atingir bons resultados, retornando uma baixa diferença de $F_1 = 0.08$ para CTL e $F_1 = 0.07$ para CCL em relação aos nossos melhores resultados. Acreditamos que isso esteja associado com a interpolação linear da [Equação 4.3](#), pois nossos resultados foram obtidos usando um $\alpha = 0.8$, dando mais importância para o modelo lexical do que para o prosódico, assim como para nossos melhores resultados. Desde que o corpú BALE e Cinderela v2 são compostos por transcrições de fala espontânea comprometida, acreditamos que as dicas lexicais desse tipo de fala ajudou o método atingir um bom desempenho.

5.6 Análise de Erros

Um de nossos propósitos é verificar se o nosso modelo está aprendendo alguma estrutura sintática, semântica ou até mesmo morfológica para segmentar as transcrições. Na [Seção 5.2](#) fizemos uma análise variando diferentes técnicas de indução de *embeddings* de acordo com a proposta delas em capturar similaridades semânticas/sintáticas/morfológicas. Entretanto, essa análise não conseguiu mostrar qual é o tipo de informação mais importante, pois todas as técnicas conseguiram resultados bem próximos. Decidimos, então, seguir a ideia de [Palmer e Hearst \(1994\)](#) e apresentar os principais erros e acertos da nossa RCNN para o corpú da Cinderela v2. Para fazer isso, adotamos uma estratégia simples de verificar quais são os n -gramas mais prováveis de precederem/sucedem um limite de sentença para a transcrição segmentada pelo nosso método e para a transcrição original.

Mais especificamente, buscamos as k sequências de palavras $\mathbf{w}^*$ mais prováveis de preceder um limite de sentença:

$$\begin{aligned}\mathbf{w}^* &= \arg \max_{w_1, \dots, w_n}^k P(y_{n+1} = 1, w_1, \dots, w_n) \\ &= \arg \max_{w_1, \dots, w_n}^k P(y_{n+1} = 1 \mid w_1, \dots, w_n) \cdot P(w_1, \dots, w_n)\end{aligned}$$

Sendo que, de maneira análoga, podemos fazer o mesmo para palavras que sucedem um limite de sentença ao usar o n -gramas após o limite de sentença. E, podemos calcular essas probabilidades facilmente através da contagem de etiquetas/palavras nas transcrições.

Outra estratégia possível é seguir a ideia de [Cuendet et al. \(2007\)](#), que usaram uma métrica de extração de informação conhecida como *information gain*, que mede a quantidade de informação obtida para uma classificação a partir da presença ou ausência do termo sendo analisado. Entretanto, essa métrica também busca palavras que são evidências negativas, e então preferimos focar numa análise mais simples, que apenas olha as evidências que afetam positivamente a classificação.

De qualquer modo, queremos responder a pergunta: “Nosso modelo consegue capturar as evidências lexicais para a tarefa de SS?”. Para responder isso utilizamos a estratégia descrita acima, obtendo uma lista de $k = 10$ pistas lexicais mais prováveis que precedem/sucedem um limite de sentença nas transcrições originais $\mathcal{O}$ e nas transcrições previstas $\mathcal{P}$ pelo nosso modelo. Essas listas são apresentadas na [Tabela 27](#).

Tabela 27 – Dicas lexicais mais frequentes nas transcrições de CTL e CCL

CTL				CCL			
Precede $\mathcal{O}$	Precede $\mathcal{P}$	Sucedem $\mathcal{O}$	Sucedem $\mathcal{P}$	Precede $\mathcal{O}$	Precede $\mathcal{P}$	Sucedem $\mathcal{O}$	Sucedem $\mathcal{P}$
né (126)	né (166)	e (414)	e (442)	né (156)	né (175)	e (384)	e (437)
dela (64)	dela (66)	ai (164)	ai (164)	dela (52)	dela (54)	ai (156)	ai (161)
festa (40)	baile (42)	ela (82)	mas (86)	ela (38)	ela (48)	mas (76)	mas (100)
baile (40)	festa (39)	mas (80)	então (80)	sapatinho (34)	festa (41)	então (68)	então (71)
ela (38)	ela (38)	então (76)	ela (75)	cinderela (34)	cinderela (39)	ela (51)	ela (67)
cinderela (34)	príncipe (32)	só (54)	só (63)	festa (34)	sapatinho (38)	só (46)	a (53)
sapatinho (30)	sapatinho (31)	o (46)	a (39)	sapato (32)	sapato (37)	a (44)	só (48)
príncipe (28)	cinderela (29)	a (40)	o (34)	príncipe (28)	príncipe (34)	o (34)	o (45)
filhas (24)	bonita (29)	quando (26)	quando (24)	castelo (28)	castelo (29)	em (18)	depois (19)
bonita (24)	filhas (28)	ai (18)	ai (18)	noite (22)	noite (23)	as (16)	é (18)

Fonte: Elaborada pelo autor.

Primeiramente, podemos ver que as dicas lexicais que mais ocorrem para ambos conjuntos de CTL e CCL são “né” precedendo o limite de sentença (B), e “e” sucedendo B , conforme verificamos na construção das *baselines*. Tanto para as transcrições originais ($\mathcal{O}$) quanto para as previstas pelo nosso modelo ($\mathcal{P}$).

Comparando os vocabulários formados entre essas listas $\mathcal{O}$ e $\mathcal{P}$ para palavras que precedem e sucedem são praticamente as mesmas para CTL e também para CCL. Além disso, as palavras que sucedem/precedem nas listas $\mathcal{O}$ e $\mathcal{P}$ são também praticamente as mesmas quando

intercalamos os pacientes (CTL vs CCL). Isso demonstra que ambos os pacientes produzem pistas lexicais de palavras de maneira semelhante, e o nosso modelo conseguiu capturar essas evidências.

Além disso, podemos ver que nosso modelo não está apenas prevendo limites de sentença quando uma palavra que precede/sucede um limite de sentença aparece. Por exemplo, a palavra “cinderela” precede 34 vezes nas transcrições originais, mas aparece 29 vezes na predição do nosso modelo. Entretanto, olhando para a quantidade de vezes que as palavras ocorrem nessas listas, temos a impressão de que nosso modelo está simplesmente prevendo *Bs* sempre que essas palavras aparecem. Essa é uma suposição compreensível, já que temos uma quantidade pequena de dados e trabalhamos com redes neurais complexas, que possuem um grande conjunto de pesos a serem treinados, implicando em múltiplos hiperplanos cortando o espaço, de modo que é possível que eles superajustem um subespaço onde essas palavras ocorrem. Por exemplo, a palavra “e” sucede um *B* 442 vezes nas transcrições segmentadas automaticamente, enquanto nas transcrições originais ela precede apenas 414 vezes. Mostramos na [Tabela 28](#) que isso não ocorreu através dos diferentes erros cometidos pelo nosso modelo: falsos positivos (FP) e falsos negativos (FN).

Tabela 28 – Erros mais frequentes cometidos pela RCNN para transcrições de CTL e CCL

CTL				CCL			
Precede FP	Precede FN	Sucede FP	Sucede FN	Precede FP	Precede FN	Sucede FP	Sucede FN
né (42)	ela (10)	e (87)	e (59)	né (27)	dela (11)	e (88)	e (35)
dela (11)	cinderela (10)	ela (30)	ela (37)	ela (18)	ela (8)	mas (33)	ela (16)
não (11)	dela (9)	mas (14)	o (22)	dela (13)	noite (8)	ela (32)	as (13)
ela (10)	ta (7)	a (13)	a (14)	cinderela (11)	filhas (8)	o (20)	em (12)
noite (10)	correndo (7)	só (12)	você (12)	príncipe (10)	né (8)	a (20)	a (11)
fada (9)	madrasta (7)	não (11)	quando (8)	festa (10)	também (7)	ai (14)	não (10)
falou (9)	baile (7)	ele (11)	mas (8)	sapato (10)	casa (6)	eu (11)	ai (9)
baile (9)	que (6)	o (10)	ai (8)	sapatinho (9)	cinderela (6)	que (11)	mas (9)
príncipe (8)	tudo (6)	então (9)	não (7)	noite (9)	foi (6)	então (9)	o (9)
carruagem (8)	princesa (5)	você (9)	ele (7)	seguida (7)	sapatinho (5)	não (9)	eu (8)

Fonte: Elaborada pelo autor.

Na [Tabela 28](#), podemos ver que, no caso da palavra “e”, o modelo deixou de prever um limite de sentença antes dela exatamente 59 vezes, sendo essa a palavra que sucede um *B* que o modelo não previu com mais frequência. Olhando para o vocabulário da tabela de erros, podemos ver que eles são bem parecidos com os encontrados na [Tabela 27](#) de evidências lexicais. Isso significa que nosso modelo está, de certo modo, tentando colocar pontos finais em lugares prováveis em que eles aconteçam de fato.

Outra informação clara que essa tabela nos dá é que as palavras que nosso modelo mais errou são, na maioria dos casos, justamente as que são mais frequentes como dicas lexicais. E, como esperado, as seguintes palavras foram as mais ambíguas para um modelo preditivo tentar classificar: “né”, “dela”, “e”, “mas”, “as”, “ela”. Entre essas, acreditamos que as palavras “e”, “ela” são ainda mais difíceis de serem desambiguadas para SS, isso porque elas ocorrem tanto no

contexto logo à esquerda de B (precedem) quanto à direita de B (sucedem).

Podemos ver também um exemplo claro de palavra que foi mal classificada provavelmente devido a disfluências que o nosso modelo aprendeu: por exemplo, a preposição³ “que” precedendo um limite de sentença nas transcrições de CTL. Afirmamos que seja devido a disfluências, pois a palavra “que” acontece frequentemente em repetições (“aqui que que . bom a cinderela”) e interrupções (“sapato né . que . essas duas”).

Para mostrar que nosso método não aprende apenas evidências baseadas em dicas de palavras (unigramas), apresentamos na [Tabela 29](#) os dez trigramas mais prováveis que precedem um B para as transcrições originais ($\mathcal{O}$) e as geradas pela nossa RCNN ($\mathcal{P}$).

Tabela 29 – Trigramas mais frequentes que precedem um limite de sentença.

CTL		CCL	
Precede $\mathcal{O}$	Precede $\mathcal{P}$	Precede $\mathcal{O}$	Precede $\mathcal{P}$
felizes para sempre	felizes para sempre	com o príncipe	com o príncipe
com o príncipe	com o príncipe	felizes para sempre	felizes para sempre
com a madrasta	perdeu o sapatinho	ir ao baile	ir ao baile
perdeu o sapatinho	tinha duas filhas	tinha duas filhas	era muito bonita
da meia noite	ir ao baile	da meia noite	perdeu o sapatinho
tinha duas filhas	com a madrasta	dona do sapatinho	dona do sapatinho
ela saiu correndo	foi pra festa	era muito bonita	ela gostava muito
foi pra festa	ir pro baile	e tudo mais	foi atrás dela
filhas da madrasta	a meia noite	a roupa dela	da meia noite
ir pro baile	era muito bonita	perdeu o sapatinho	tinha duas filhas

Fonte: Elaborada pelo autor.

É possível perceber através da [Tabela 29](#) que praticamente os mesmos trigramas são frequentes nas transcrições originais e nas previstas pelo nosso modelo. Isso nos indica que nosso modelo não está se baseando apenas em unigramas para segmentar a fala em sentenças, mas provavelmente está se baseando em informações mais complexas, como semânticas e sintáticas.

Outra evidência para essa afirmação é a relação de tamanho médio de sentenças entre as transcrições. As sentenças das transcrições originais de CTL possuem em média 12.38 (± 7.40) palavras, já as sentenças previstas por nosso modelo possuem em média 11.79 (± 8.40) palavras. Ou seja, o tamanho das sentenças geradas pelo nosso modelo está bem próximo das definidas pelos anotadores. Além disso, o mesmo acontece para CCL, as sentenças das transcrições originais possuem uma média de 12.79 (± 7.23) palavras, e as previstas pelo nosso modelo possuem em média 11.45 (± 7.64) palavras.

Esses valores ainda nos mostram que, como visto na etapa de anotação, nosso modelo retorna sentenças mais curtas para transcrições de CCL do que para as de CTL. Podemos dizer ainda que essas sentenças variam bastante em seu tamanho, o que é comum para sentenças

³ A classe morfossintática do *que* é bastante ambígua (KEPLER; FINGER, 2010), portanto nesse caso assumimos que seja uma preposição.

vindas da fala, principalmente de fala espontânea (LIU *et al.*, 2005a; MROZINSKI *et al.*, 2006; BATISTA; MAMEDE, 2011).

5.7 Considerações Finais

Avaliamos o método proposto para a tarefa de SS com várias abordagens diferentes, e descobrimos que as *embeddings* de palavras induzidas pelo *word2vec* são mais eficientes para a nossa RCNN. Ao adicionar informações prosódicas em conjunto com essas *embeddings* e etiquetas morfossintáticas, verificamos que a combinação entre todas essas informações geralmente resulta em melhores resultados em termos de F_1 . E, analisando em termos de *SER*, verificamos que apenas as *embeddings* sozinhas nos dão um resultado melhor do que usando todas as *features*, indicando uma possível linha de investigação quanto a qualidade das informações prosódicas utilizadas. Justificamos a escolha de um modelo baseado em redes neurais recorrentes convolucionais ao verificar o desempenho de diferentes métodos de classificação baseados em redes neurais, sendo que nossa RCNN obteve melhores resultados em todos os *cópus* testados. Verificamos também que nosso modelo conseguiu bons resultados quando testamos com narrativas cuja história é diferente da história usada para as narrativas de treinamento. Por fim, com a análise de erros, mostramos que a RCNN proposta conseguiu aprender evidências lexicais, sintáticas e semânticas no *cópus* de narrativas transcritas de testes neuropsicológicos base desta pesquisa.

AVALIAÇÕES PARA DETECÇÃO DE DISFLUÊNCIAS

Para a tarefa de DD, realizamos os experimentos para preenchimentos ([Seção 6.1](#)) e disfluências de edição ([Seção 6.2](#)) com, essencialmente, a mesma RCNN já utilizada para a tarefa de SS com ambos tipos de *features*: textuais e prosódicas. A única diferença do método usado para detectar preenchimentos para o método usado para SS é em relação ao conjunto de etiquetas. Já para detectar disfluências de edição introduzimos *features* manuais destacadas na literatura, e adicionamos um modelo de CRF para decodificar a saída da RCNN, apresentado na [Subseção 6.2.1](#). Por fim, utilizamos a segmentação automática do capítulo anterior juntamente com a remoção de preenchimentos e disfluências de edição para avaliar nosso método extrinsecamente com o Coh-Metrix-Dementia na [Seção 6.3](#).

Para ambas as tarefas, primeiramente criamos *baselines* baseados em dicas lexicais observadas no corpus anotado. Após isso, verificamos o desempenho de algoritmos não supervisionados para indução de *word embeddings*, e então usamos o modelo de RCNN com essas *embeddings* para realizar avaliações em relação a importância das *features* e comparação de classificadores.

Em geral, todos os experimentos usando o modelo de RCNN foram realizados com os mesmos hiper-parâmetros apresentados anteriormente na [Tabela 20](#), mas para experimentos específicos, variamos alguns hiper-parâmetros com o objetivo de analisar o impacto deles. Além disso, todos os experimentos foram realizados utilizando o corpus Cinderela v2 anotado com disfluências e todos os resultados são reportados em termos de F_1 , pois a maioria dos trabalhos relacionados também utilizam essa métrica.

6.1 Preenchimentos

Conforme vimos na [Subsubseção 4.1.1.3](#), termos de edição explícitos são bastante raros em nossos corpus: apenas 12 palavras foram marcadas, o que corresponde em apenas 2% da anotação de preenchimentos (total de 650). Por esse motivo, em todos os experimentos realizados nesta seção, decidimos avaliar nossos métodos usando apenas Pausas Preenchidas (PPs) e Marcadores Discursivos (DDs) - removendo a marcação de termos de edição explícitos dos conjuntos de treinamento e teste.

6.1.1 Baselines

Os *baselines* computados para detecção de preenchimentos são mais fortes do que o *baseline* que simplesmente classifica todas as instâncias como a classe majoritária (o qual reportamos como *chance baseline*). Mais especificamente, os *baselines* reportados são:

- **Lista de palavras:** Vários trabalhos na literatura buscam identificar preenchimentos apenas utilizando uma lista de palavras. Como esse trabalho é o primeiro realizado para o português brasileiro, não há uma lista pronta. Então buscamos as palavras que mais ocorreram como cada tipo de preenchimento (PP - Pausa Preenchida, MD - Marcador Discursivo) em nosso corpus anotado e selecionamos uma porção delas. A lista das palavras selecionadas pode ser vista na [Tabela 30](#).

Tabela 30 – Palavras selecionadas para *baselines* para cada tipo de preenchimento.

PP		MD	
ahn	hrum	né	assim
ah	hinhun	então	sabe
hun	uhm	tal	ne
oh	hanhan	bom	-
hunhun	han	ta	-
hum	ohr	viu	-
heim	ha	enfim	-
uhn	eh	tá	-

Fonte: Elaborada pelo autor.

Vale lembrar que algumas palavras frequentes não foram selecionadas como, por exemplo, a palavra “é” que foi marcada várias vezes como pausa preenchida, e a palavra “viu” que foi marcada como marcador discursivo. Isso porque elas também ocorrem frequentemente como não sendo preenchimento.

- **Árvore de decisão:** Árvores de decisão são algoritmos simples que buscam dividir as instâncias homogeneamente de acordo com suas classes ao analisar as *features* que as descrevem. Nossa árvore de decisão baseou-se apenas na identidade das palavras e nas suas classes morfosintáticas dentro de uma janela de tamanho 7 (três palavras à esquerda

e três à direita). Além disso, esse nosso modelo realiza divisões recursivas até que as folhas sejam da mesma classe. Avaliamos cada grupo utilizando validação cruzada com 5 *folds*.

Os resultados desses *baselines* são apresentados na [Tabela 31](#) em termos de F_1 .

Tabela 31 – Resultados dos *baselines* para preenchimentos.

<i>Baseline</i>	CTL		CCL		DA	
	PP	MD	PP	MD	PP	MD
Chance	0.01	0.03	0.02	0.04	0.01	0.05
Lista de Palavras	0.69	0.73	0.79	0.75	0.68	0.57
Árvore de Decisão	0.62	0.84	0.76	0.79	0.41	0.73

Fonte: Elaborada pelo autor.

Primeiramente, podemos ver na [Tabela 31](#) que o *chance baseline* foi muito pior que os outros dois *baselines*. O *baseline* de lista de palavras desempenhou melhor que a árvore de decisão para a detecção de pausas preenchidas, com uma margem larga em transcrições de DA. Já marcadores discursivos são mais bem detectados por árvore de decisão, também com uma margem larga em transcrições de DA. Isso provavelmente está relacionado ao fato de que pausas preenchidas geralmente possuem uma identidade bastante reservada, mas marcadores discursivos não, os quais podem ser bastante ambíguos. Por exemplo, a pausa preenchida mais ambígua é a palavra “é”, e ela não foi incluída na lista de palavras.

Em média, os resultados da árvore de decisão são melhores que os de lista de palavras, o que é esperado pois a árvore também levou em consideração as classes morfossintáticas das palavras. De fato, as *features* mais importantes que a árvore de decisão encontrou para a classificação foi a etiqueta morfossintática da próxima palavra (c_{i+1} : 69%), e a da próxima depois dessa (c_{i+2} : 13%) e sua identidade (w_{i+2} : 5%), e apenas após essas três, a identidade da própria palavra foi a mais discriminativa (w_i : 3%) - que é justamente a única *feature* utilizada pelo *baseline* de listas de palavras.

Nas próximas tabelas de resultados, os *baselines* apresentados correspondem aos melhores resultados (em negrito) encontrados aqui, para cada grupo de pacientes e para cada tipo de preenchimento.

6.1.2 Avaliação de Word Embeddings

Na avaliação de *word embeddings* para SS ([Seção 5.2](#)) mostramos que em média o melhor algoritmo de indução de *embeddings* foi o Skip-gram para vetores de 600 dimensões. Portanto, nesses experimentos decidimos alterar apenas a técnica utilizada: word2vec, wang2vec, e fastText. Variar essas técnicas é interessante pois cada uma foi proposta para capturar diferentes similaridades. O word2vec foi proposto originalmente para capturar similaridades semânticas, o

wang2vec foi uma modificação para capturar informações sintáticas, e o fastText foi proposto para induzir vetor baseando-se na morfologia das palavras. Descrevemos cada uma das técnicas de indução de *embeddings* na Subseção 2.4.1, e apresentamos os resultados dessa variação na Tabela 32.

O corpus utilizado para a indução desses vetores é o mesmo utilizado para a avaliação de SS. O corpus é composto por artigos da Wikipédia de 2016, artigos de notícias do portal G1¹ e textos do corpus PLN-BR (BRUCKSCHEN *et al.*, 2008). Realizamos pré-processamento de modo a deixar todas as palavras em caixa baixa e separamos os *tokens* de sinais de pontuações. Após esses procedimentos, o corpus resultante para a indução das *embeddings* retornou ~356M *tokens* dentre os quais ~1.7M eram distintos um do outro.

Tabela 32 – Resultados de variação de *embeddings* para preenchimentos.

Técnica	CTL		CCL		DA	
	PP	MD	PP	MD	PP	MD
Baseline	0.69	0.84	0.79	0.79	0.68	0.73
fastText	0.63	0.92	0.70	0.90	0.64	0.82
wang2vec	0.58	0.91	0.67	0.93	0.66	0.83
word2vec	0.60	0.91	0.68	0.91	0.63	0.80

Fonte: Elaborada pelo autor.

Quando levamos em consideração o F_1 para pausas preenchidas e marcadores discursivos, o *fastText* desempenhou melhor que as outras técnicas na maioria dos casos. Mais especificamente, para CTL o melhor resultado foi do *fastText* para PP e MD. Para CCL o *fastText* foi o melhor para PP, mas o *wang2vec* foi melhor para MD. Já para DA o *wang2vec* desempenhou melhor para ambos os preenchimentos. O *word2vec*, que tinha sido o melhor na tarefa de SS, dessa vez foi a técnica que obteve os resultados mais baixos. Acreditamos que o *fastText* teve um bom desempenho bem para as PPs pois geralmente elas possuem a letra “h” próximas de consoantes como *m* e *n* em sua composição morfológica (e.g. *ahn*, *hum*, *hun*).

Podemos observar ainda que a RCNN obteve melhores resultados para MDs do que para PPs, sendo que os resultados para MDs foram bem acima do *baseline*, porém o mesmo não aconteceu para PPs, em que o método não conseguiu ultrapassar o *baseline* utilizando apenas as *word embeddings*. Quando comparamos com os resultados obtidos pela detecção humana (Subsubseção 4.1.1.3), percebemos que a RCNN com *embeddings* desempenhou melhor para detecção de marcadores discursivos em transcrições de CTL e CCL, e ficando apenas 2% atrás para transcrições de DA.

Nosso objetivo com esse experimento foi o de verificar qual tipo de *embedding* desempenha melhor para a tarefa de detecção de preenchimento: as que capturam similaridades semânticas (*word2vec*); sintáticas (*wang2vec*); ou morfológicas (*fastText*). Entretanto, não conse-

¹ <<http://g1.globo.com/>>

guimos discernir com precisão qual é o tipo que mais influência de modo geral, pois as diferenças de um para o outro foram muito pequenas. E por mais que uma técnica tenha superado a outra em um conjunto específico, teríamos ainda que investigar se foi devido a natureza da técnica ou devido a condições secundárias, como hiperparâmetros ou inicialização aleatória. Uma hipótese que tínhamos era que o *fastText* superaria as outras técnicas, pois ele tem a capacidade de capturar similaridades morfológicas das palavras, de modo a aproximar palavras fora do vocabulário daquelas lexicalmente parecidas com elas. E, conforme visto na [Tabela 32](#), o *fastText* realmente conseguiu bons desempenhos para detecção de pausas preenchidas, as quais possuem características morfológicas evidentes.

Se tirarmos a média de todas as combinações de técnicas (*word2vec*, *wang2vec*, *fastText*) para os três tipos de pacientes e para os dois tipos de preenchimento, a técnica que retorna maior F_1 é o *fastText*. E então, foi essa técnica (Skip-gram com 600 dimensões) que usamos em todos os próximos experimentos para detecção de preenchimentos.

6.1.3 Importância das Features

Com o cenário de avaliação definido com *embeddings* de 600 dimensões induzidas pela técnica *fastText* com o algoritmo Skip-gram, decidimos avaliar nosso modelo em relação às *features* morfossintáticas e prosódicas. A partir deste experimento, pelo fato de usarmos informações prosódicas, deixamos de avaliar as transcrições de DAs, pois infelizmente não conseguimos realizar o alinhamento forçado entre as transcrições e o áudio nelas. A razão para a exclusão foi pelos áudios dos participantes com DA estarem com qualidade muito baixa, de modo que não era possível identificar o que o participante estava falando dado o volume baixo da voz dos participantes e a presença de ruídos de fundo (as gravações foram feitas em um hospital).

A [Tabela 33](#) apresenta os resultados de avaliação da importância das *features*, usando a métrica F_1 para transcrições de CTL e CCL. Na [Tabela 33](#) podemos ver que a RCNN apresenta melhores resultados para transcrições de CCL do que para CTL, tanto para pausas preenchidas como para marcadores discursivos. Isso pode estar relacionado com a quantidade de disfluências presentes em cada grupo. Outra observação importante é que a RCNN não conseguiu ultrapassar o *baseline* forte de pausas preenchidas, porém o *baseline* de marcadores discursivos foi superado por uma margem larga.

Analisando a importância dos diferentes conjuntos de *features*, percebemos que *embeddings* tem um grande impacto na classificação para ambos os grupos de pacientes. Informações de etiquetas morfossintáticas (PoS) ajudou com uma margem pequena, já que tem uma pequena diferença de quando utilizado *embeddings*: pausas preenchidas (0.01), e marcadores discursivos (0.02) em transcrições de CCL; sendo que para transcrições de CTL não houve diferença. Isso nos diz que, como observado também para SS, as *embeddings* sozinhas já conseguem trazer informações morfossintáticas suficientes. É evidente que o peso para as *features* prosódicas é muito pequeno: quando usada sozinha a RCNN apresentou os piores resultados, e quando

Tabela 33 – Resultados de cada conjunto de *features* para detecção de preenchimentos.

<i>Features</i>	CTL						CCL					
	PP			MD			PP			MD		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
Baseline	0.99	0.53	0.69	0.86	0.83	0.84	0.98	0.66	0.79	0.78	0.80	0.79
PoS	0.02	0.26	0.05	0.13	0.49	0.20	0.07	0.19	0.09	0.18	0.39	0.25
Prosódia	0.02	0.42	0.03	0.09	0.22	0.13	0.04	0.51	0.07	0.06	0.15	0.08
Embeddings	0.50	0.84	0.63	0.92	0.91	0.92	0.62	0.81	0.70	0.91	0.90	0.90
PoS + Emb.	0.49	0.85	0.63	0.92	0.91	0.92	0.61	0.84	0.70	0.92	0.93	0.92
PoS + Pros.	0.02	0.26	0.05	0.13	0.49	0.20	0.07	0.19	0.09	0.18	0.39	0.25
Emb. + Pros.	0.50	0.84	0.62	0.92	0.91	0.92	0.62	0.81	0.70	0.91	0.90	0.90
Todas	0.49	0.85	0.63	0.92	0.91	0.92	0.61	0.84	0.71	0.92	0.94	0.93

Fonte: Elaborada pelo autor.

PoS corresponde às etiquetas morfossintáticas (*Part-of-Speech Tags*).

acrescentada com *features* lexicais, o resultado permaneceu praticamente o mesmo que somente usando as mesmas *features* lexicais sozinhas. A única melhoria ao acrescentar informações prosódicas com *embeddings* e etiquetas morfossintáticas foi de 0.01 nas transcrições de CCL (para PP e MD), o qual pode ter sido consequência de inicializações aleatórias, por ser muito pequeno.

Assim como aconteceu para as avaliações de SS, acreditamos que a qualidade das gravações pode ter influenciado no peso dado para as informações prosódicas, já o cópús Cinderela v2 foi gravado no cenário clínico, em um hospital. Apesar disso, podemos ver que nosso método obteve um desempenho levemente superior para ambos grupos de pacientes quando todas as *features* foram usadas. Aliás, esses resultados foram obtidos usando $\alpha = 0.95$ da interpação linear da Equação 4.3, indicando que nosso modelo baseia-se fortemente nas *features* textuais. Além disso, podemos ver que ao utilizar apenas *embeddings* os resultados obtidos ficam muito próximos ao utilizar todas as *features*, indicando que elas são suficientes para a detecção de preenchimentos.

O trabalho de Shriberg, Bates e Stolcke (1997) mostrou que, para fala espontânea de conversações telefônicas da língua inglesa, é possível usar apenas informações prosódicas para detectar preenchimentos com uma acurácia de 89.7%. Como isso é o inverso dos resultados encontrados aqui, não temos uma conclusão clara do potencial das informações prosódicas para essa tarefa, mesmo porque estamos em um outro cenário: fala comprometida e cópús pequeno. Assim como para SS, um trabalho futuro planejado para o projeto consiste em verificar algumas peculiaridades que podem ter influenciado os resultados, como: qualidade de gravação do áudio; natureza inerente dos pacientes (configuração clínica das gravações e estatísticas demográficas); insuficiência discriminativa das *features* prosódicas extraídas.

Se tirarmos a média entre os tipos de preenchimentos, os melhores resultados são os obtidos pela RCNN utilizando todas as *features* para transcrições de CTL ($F_1 = 0.78$) e CCL ($F_1 = 0.85$). Tirar a média entre os tipos de preenchimentos significa em decidir se uma

palavra faz parte ou não de uma região de disfluência, caracterizando um problema binário. Comparando nossos resultados com os resultados (F_1 entre 0.90 e 0.92) de trabalhos recentes que também usam redes neurais para detecção de preenchimentos (BERTERO *et al.*, 2015; HOUGH; SCHLANGEN, 2015; HOUGH; SCHLANGEN, 2017), percebemos que o desempenho da nossa RCNN está cerca de 10% abaixo. Entretanto, essa margem é menor se utilizarmos os resultados dos *baselines* de listas de palavras para detecção de pausas preenchidas. Isso indica que nosso modelo é bom o suficiente para a detecção de marcadores discursivos, porém para pausas preenchidas é necessário realizar uma investigação dos erros cometidos pela RCNN: erros derivados da ocorrência de palavras fora do vocabulário de *embeddings*, de modo a dificultar a captura dicas de identidade das palavras para a classificação; ou palavras muito ambíguas, como o caso da palavra “é”. Por exemplo, a precisão do *baseline* (99% CTL e 98% CCL) para pausas preenchidas é muito maior em relação a RCNN (49% CTL e 61% CCL), sugerindo que a RCNN se confunde bastante, entretanto a cobertura da RCNN é muito maior que do *baseline*, mostrando que o *baseline* deixa de classificar um maior número de pausas preenchidas que a RCNN. Investigamos esses fenômenos em mais detalhes na Subseção 6.1.5.

6.1.4 Comparação de Métodos

Usamos novamente o CRF (o mesmo usado para SS), já que modelos de CRF também foram bastante empregados para detecção de preenchimentos (LIU *et al.*, 2005; WANG *et al.*, 2010; BERTERO *et al.*, 2015). E para dar suporte à nossa escolha por um modelo recorrente bidirecional, nós separamos nosso método em outros três utilizados também para avaliações de SS (Seção 5.4):

- **MLP:** Removemos as camadas convolucionais e recorrentes, e adicionamos uma camada totalmente conectada oculta, de modo a formar uma MLP com uma camada oculta. Esse modelo é similar, mas não exatamente o mesmo, que um dos modelos usado por Bertero *et al.* (2015).
- **RNN:** Removemos a camada de convolução, de modo que passamos a entrada diretamente para a camada recorrente bidirecional. Esse modelo é similar, mas não exatamente o mesmo, que o usado por Hough e Schlangen (2015), Hough e Schlangen (2017).
- **CNN:** Removemos a camada recorrente bidirecional e passamos a saída da camada convolucional direto para a camada de saída.

Os resultados para cada método, apresentados na Tabela 34, foram obtidos usando todas as *features*.

Nosso método conseguiu os melhores resultados em ambos os grupos de pacientes para a detecção de marcadores discursivos, e empatou com o CRF para detecção de pausas preenchidas de CCL. Podemos ver que, em geral, a RNN obteve os piores resultados para detecção de

Tabela 34 – Resultados de cada método para detecção de preenchimentos.

Método	CTL						CCL					
	PP			MD			PP			MD		
	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>	<i>P</i>	<i>R</i>	<i>F₁</i>
Baseline	0.99	0.53	0.69	0.86	0.83	0.84	0.98	0.66	0.79	0.78	0.80	0.79
CRF	0.90	0.51	0.65	0.98	0.81	0.89	0.96	0.57	0.71	0.98	0.82	0.89
MLP	0.50	0.73	0.60	0.83	0.88	0.86	0.58	0.80	0.67	0.85	0.91	0.88
RNN	0.33	0.86	0.48	0.91	0.82	0.85	0.50	0.91	0.65	0.89	0.85	0.87
CNN	0.44	0.71	0.54	0.89	0.90	0.90	0.60	0.80	0.68	0.93	0.91	0.92
RCNN	0.49	0.85	0.63	0.92	0.91	0.92	0.61	0.84	0.71	0.92	0.94	0.93

Fonte: Elaborada pelo autor.

pausas preenchidas e também para marcadores discursivos. Em geral, a CNN conseguiu atingir melhores resultados que os métodos de CRF, MLP e RNN para marcadores discursivos, e o CRF conseguiu melhores resultados que os outros três para detecção de pausas preenchidas, inclusive melhor que a RCNN para transcrições de CTL. O resultado da MLP para o subconjunto de pacientes com CCL não foi tão bom quanto ao da CNN, o que indica, de certo modo, que um modelo neural simples com camadas totalmente conectadas não é capaz de lidar bem com narrativas potencialmente comprometidas. Em contrapartida, para os dados de CTL, a MLP obteve resultados bem próximos da RCNN, com diferença de 0.03 para PP e 0.06 para MD. Ao contrário do que vimos nas avaliações de SS, o modelo de CRF obteve bons resultados aqui, entretanto a sua distribuição de precisão e cobertura é considerável: a precisão sempre foi maior que a cobertura, indicando que o modelo se limitou a acertar apenas um conjunto fixo de palavras.

Por fim, assim como aconteceu nas avaliações de SS, a nossa RCNN atingiu em média os melhores resultados, implicando que a união de um modelo convolucional com um modelo recorrente foi uma boa escolha para detecção de preenchimento em fala comprometida. Analisando os resultados superiores da CNN em relação aos da RNN podemos ver que a maior influência foi da camada convolucional. A adição de uma camada recorrente com LSTM foi, potencialmente, capaz de lidar com alguns casos particulares, provavelmente de longa dependência, similares às descobertas ao usar células recorrentes normais e células de LSTM de [Hough e Schlangen \(2017\)](#). Teoricamente, a CNN sozinha não é capaz de lidar com que essas peculiaridades devido ao tamanho fixo do filtro no processo de convolução.

6.1.5 Análise de Erros

Com base nas avaliações verificamos que o *baseline* de lista de palavras desempenhou melhor na detecção de pausas preenchidas que vários modelos de redes neurais, inclusive a nossa abordagem RCNN. Já para detecção de marcadores discursivos nossa RCNN obteve os melhores resultados, superando o *baseline* de árvore de decisão por uma margem larga. Para elucidar os casos em que nossa RCNN está acertando e errando, destacamos os verdadeiros positivos (VP), falsos positivos (FP) e falsos negativos (FN) para cada tipo de preenchimento na [Tabela 35](#) para

transcrições de CTL e na [Tabela 36](#) para transcrições de CCL.

Tabela 35 – Acertos e erros mais frequentes para detecção de preenchimentos em transcrições de CTLs

PP			MD		
TP	FP	FN	TP	FP	FN
é (13)	é (10)	é (5)	né (103)	tal (5)	ta (5)
ahn (7)	sempre (10)	ha (1)	tal (7)	então (3)	tá (2)
hun (4)	felizes (4)	-	bom (6)	experimente (1)	é (1)
ah (3)	vida (1)	-	e (4)	embora (1)	tudo (1)
uhn (2)	casaram (1)	-	então (2)	e (1)	tal (1)
ohr (1)	ciúme (1)	-	tá (1)	-	bem (1)
hunhun (1)	toda (1)	-	ta (1)	-	sabe (1)
hum (1)	fiz (1)	-	enfim (1)	-	viu (1)
-	eles (1)	-	assim (1)	-	muito (1)

Fonte: Elaborada pelo autor.

Tabela 36 – Acertos e erros mais frequentes para detecção de preenchimentos em transcrições de CCLs

PP			MD		
TP	FP	FN	TP	FP	FN
ahn (41)	é (10)	é (8)	né (112)	então (5)	e (5)
é (17)	sempre (6)	hunhun (3)	tal (9)	bacana (1)	então (3)
ah (2)	felizes (4)	hinhun (1)	então (7)	bom (1)	enfim (2)
hun (2)	u (3)	hanhan (1)	e (4)	mas (1)	bom (1)
uhm (1)	princesa (1)	-	bom (3)	né (1)	né (1)
hrum (1)	casaram (1)	-	-	é (1)	olha (1)
han (1)	história (1)	-	-	assim (1)	assim (1)

Fonte: Elaborada pelo autor.

É possível ver nas duas tabelas de acertos e erros que, como previsto, a maioria dos erros de pausas preenchidas são devido à palavra “é”. Mais especificamente, a RCNN os classificou erroneamente mais vezes (FPs) do que deixou de classificá-los (FNs). Já pausas preenchidas que estavam na listas de palavras ([Tabela 30](#)) deixaram de ser classificadas algumas poucas vezes, provavelmente porque elas não estão no vocabulário das *embeddings* e, portanto, elas compartilhavam uma mesma representação aleatória, dificultando a análise da RCNN em relação às suas identidades. Podemos ver também que a RCNN conseguiu acertar o marcador discursivo “né” todas as vezes (deixando de classificá-lo apenas uma vez para transcrições de CCL). A palavra *então* é o marcador discursivo mais difícil de classificar, representando mais erros do que acertos para ambas as transcrições.

Nessas tabelas, podemos perceber mais claramente que marcadores discursivos e pausas preenchidas são, geralmente, facilmente identificáveis apenas analisando o token em si. Por isso o *baseline* de lista de palavras conseguiu bons resultados para pausas preenchidas. Por exemplo, a RCNN classificou erroneamente as palavras “sempre” e “felizes” como pausas preenchidas, as quais evidentemente não possuem essa função e, portanto, utilizando listas de palavras esse erro dificilmente seria cometido.

Em geral, acreditamos que com mais dados é possível aprimorar ainda mais o modelo para não cometer erros evidentes como os FPs de pausas preenchidas, os quais foram cometidos poucas vezes mas ainda assim estão entre os mais frequentes. Para marcadores discursivos os erros cometidos são um pouco mais compreensíveis, mas ainda assim a grande maioria dos acertos foi em relação à palavra “né”, que ocorre frequentemente no corpus. Na verdade, para as pausas preenchidas, podemos observar que os FNs (com exceção do “é”), são palavras que estavam na lista da [Tabela 30](#), isso indica que usar um modelo neural para tentar classificá-las é um desperdício computacional. Com essa análise, fica evidente que a melhor escolha para a detecção de preenchimentos é combinar lista de palavras para a detecção de pausas preenchidas e nosso modelo de RCNN para detecção de marcadores discursivos. No caso de se desejar um desempenho maior, é possível treinar a RCNN apenas para decidir o caso da pausa preenchida “é”, pois ela é muito ambígua e ocorre frequentemente nos textos.

Nossas transcrições foram pré-processadas de modo a tirar as marcações do manual do NURC (Anexo A) para simular a saída de um RAV. Entretanto, essas marcações traziam informações valiosas para pausas preenchidas, como a marcação de prolongamento e pausas. Embora essas informações não estejam disponíveis na transcrição para um método que as recebe de um RAV, elas podem ser incorporadas no momento da anotação para ajudar o trabalho do anotador. Desse modo, verificamos o caso da ambiguidade da palavra “é” em relação à nossa anotação ao solicitar que um anotador revisasse a anotação utilizando essas marcações juntamente com áudio do reconto. Nossa próxima análise consiste em utilizar a estratégia de combinar o método de lista de palavras para detectar pausas preenchidas, e retrainar a RCNN para detectar marcadores discursivos e a pausa preenchida “é”. Os resultados ao utilizar essa estratégia antes (transcrições originais) e após a revisão da anotação (transcrições revisadas) podem ser vistas na [Tabela 37](#).

Tabela 37 – Detecção de pausas preenchidas para transcrições originais e revisadas ao combinar a RCNN com a lista de palavras.

Transcrições	CTL			CCL			DA		
	P	R	F ₁	P	R	F ₁	P	R	F ₁
Baseline	1.00	0.53	0.69	1.00	0.66	0.79	1.00	0.52	0.68
Originais	0.77	0.87	0.82	0.87	0.89	0.88	0.77	0.84	0.80
Revisadas	0.86	0.94	0.90	0.90	0.95	0.92	0.60	0.82	0.70

Fonte: Elaborada pelo autor.

Para deixar o experimento mais transparente, informamos também o número de vezes que a palavra “é” foi marcada como pausa preenchida antes vs. depois da revisão: 18 vs. 28 para CTL; 25 vs. 33 para CCL; e 21 vs. 41 para DA. Primeiramente, na [Tabela 37](#) percebemos que a estratégia de usar a RCNN para detectar a pausa preenchida “é” e a lista de palavras para detectar as outras retornou resultados acima do *baseline*, que até então não havia sido ultrapassado. Ao utilizar as transcrições revisadas melhoramos ainda mais o desempenho para

CTL e CCL, provavelmente porque agora o peso dado para o modelo prosódico subiu de 5% para 10%. Porém, para DA esse número ainda ficou abaixo das transcrições originais, já que não extraímos informações prosódicas para as gravações desses pacientes. Isso tudo indica que tanto a revisão da anotação quanto a estratégia de combinação do *baseline* com a RCNN contribuíram bastante para melhorar os resultados para pausas preenchidas. Ao tirar a média de pausas preenchidas com marcadores discursivos, obtemos resultados com F_1 entre 90% e 92%, assim como os resultados reportados em vários trabalhos relacionados (SHRIBERG; BATES; STOLCKE, 1997; BERTERO *et al.*, 2015; HOUGH; SCHLANGEN, 2017).

6.2 Disfluências de Edição

Para a detecção de disfluências de edição realizamos dois pré-processamentos: introduzimos limites de sentença utilizando a anotação manual (Subsubseção 4.1.1.2) e removemos preenchimentos utilizando a anotação manual também (Subsubseção 4.1.1.2). Embora utilizamos a anotação manual nesses pré-processamentos, é possível realizar todos os passos de forma automática, de modo a marcar os limites de sentença de acordo com o Capítulo 5 e eliminar os preenchimentos de acordo com a Seção 6.1. Além disso, é importante salientar que a anotação manual utilizada não é a mesma que foi revisada para a anotação da pausa preenchida “é”, e portanto pequenos erros subsequentes são esperados após o passo de remoção de preenchimentos.

Lembramos que disfluências de edição possuem uma estrutura dividida em: enunciado original com um ponto de interrupção seguido possivelmente de um termo de edição explícito e então a correção. Nesta dissertação detectar disfluências de edição significa encontrar a região de disfluência correspondente ao enunciado original. Além disso, conforme vimos na Subsubseção 4.1.1.4, anotamos nosso corpus com três tipos de disfluências de edição: repetições, revisões e recomeços. Modelamos nossos métodos de modo a verificar se uma palavra faz parte da região do enunciado original da disfluência e também distinguir o enunciado e, portanto, reportamos os resultados em F_1 para cada tipo: Repetição (*Rep*); Revisão (*Rev*); e Recomeço (*Rec*).

Vale lembrar que utilizamos os limites de sentenças apenas como *features* locais e não incluímos elas no conjunto de etiquetas da RCNN. Além disso, antes de apresentar os experimentos realizados (Subseção 6.2.2 a Subseção 6.2.6), detalhamos algumas modificações simples na RCNN para melhorar o desempenho na Subseção 6.2.1, de acordo com o que foi proposto por trabalhos relacionados.

6.2.1 Modificações na RCNN

Fizemos duas adaptações no nosso modelo de RCNN para a tarefa de detecção de disfluências de edição. A primeira foi a inclusão de *features* manuais propostas em vários trabalhos relacionados (QIAN; LIU, 2013; WANG; NG; SIM, 2014; BERTERO *et al.*, 2015;

ZAYATS; OSTENDORF; HAJISHIRZI, 2016; WANG *et al.*, 2017), as quais seguem a ideia de usar a identidade de palavras próximas. Mais especificamente, assumindo uma palavra na i -ésima posição da transcrição $w_i | 0 < i \leq \varphi$, adicionamos as seguintes *features* binárias (1 se a condição é verdadeira, 0 caso contrário):

- **Palavra duplicada:** se a palavra w_i é igual a palavra w_{i+k} , onde $-4 \leq k \leq 4$
- **PoS duplicado:** se a classe morfossintática c_i é igual a classe c_{i+k} , onde $-4 \leq k \leq 4$
- **Bigrama de palavra duplicado:** se o bigrama de palavras (w_i, w_{i+1}) é igual ao bigrama (w_{i+k}, w_{i+k+1}) , onde $-4 \leq k \leq 4$
- **Bigrama de PoS duplicado:** se o bigrama de classes morfossintáticas (c_i, c_{i+1}) é igual ao bigrama (c_{i+k}, c_{i+k+1}) , onde $-4 \leq k \leq 4$
- **Prefixo:** se a palavra w_i é um prefixo da palavra w_{i+k} , onde $1 \leq k \leq 2$, de modo que prefixos são considerados subsequências contínuas com tamanho 2, 3 e 4 a partir do início da palavra (e.g. *cinderela*: *ci*, *cin*, *cind*).
- **Fim de sentença:** se a palavra w_{i+k} , onde $-2 \leq k \leq 2$, é uma marcação de limite de sentença.

As *features* cujas posições são inválidas ($i+k \leq 0 \vee i+k > \varphi$) foram preenchidas com o valor 0, ignorando as *features* da própria posição ($k = 0$). As quatro primeiras *features* são exatamente as mesmas que as indicadas no trabalho de Wang *et al.* (2017), com exceção que o valor de k deles é 15 e o nosso é 4, pois nossas disfluências são consideravelmente menores (nosso maior enunciado original possui exatamente 4 palavras).

Já as últimas duas *features* foram utilizadas apenas nesta dissertação. As *features* de prefixo são interessantes pois muitas repetições começam com palavras incompletas; os pacientes começam a falar a palavra e interrompem ela quase em seguida, por exemplo: *a ma ma madrinha*, ou *fo foi ela*. Já a *feature* de fim de sentença pode ser útil para verificar que nenhum enunciado original deve ser marcado ao final da sentença, pois após ele geralmente há outro trecho de correção. Wang *et al.* (2017) propõem verificar se palavras próximas são semelhantes usando métricas baseadas na distância de edição, porém acreditamos que isso pode ser interpretado de maneira equivocada pelo classificador. Vários trabalhos também utilizaram modelos de língua treinados com corpus bem escritos na esperança de pontuar uma região de disfluência com um escore baixo e regiões fluentes com um escore alto (LIU *et al.*, 2005; QIAN; LIU, 2013; WANG; NG; SIM, 2014; FERGUSON; DURRETT; KLEIN, 2015; ZAYATS; OSTENDORF; HAJISHIRZI, 2016), entretanto não utilizamos essa estratégia por demandar recursos extras.

Para incorporar essas *features* manuais em nosso modelo de RCNN foi necessário adicionar mais uma matriz na camada de entrada. Ao contrário das matrizes das classes morfossintáticas,

essa nova matriz não é inicializada aleatoriamente, de modo que simplesmente passamos as *features* manuais binárias como entrada. Após obter o vetor concatenado de *word embeddings* (com tamanho d_e) e o vetor da classe morfossintática (com tamanho d_c), concatenamos ele novamente com os vetores das *features* manuais, o qual possui tamanho $d_m = 8 + 8 + 8 + 8 + 6 + 4 = 42$. Ou seja, o vetor de entrada para a camada convolucional do modelo textual possui agora um tamanho $d = d_e + d_c + d_m$.

A segunda adaptação foi a inclusão de um modelo de CRF de cadeia linear na última camada da rede neural. Apesar de parecer uma modificação complexa, podemos ver isso como a adição de mais uma camada na rede neural. De fato, a última camada passa agora a sua saída para um modelo de CRF (SUTTON; MCCALLUM *et al.*, 2012), o qual usa informações de etiquetas vizinhas para melhorar a predição da rede neural. Vários trabalhos fazem o uso de mecanismos de decodificação na saída de seus modelos preditivos, pois disfluências de edição possuem uma estrutura nas quais previsões vizinhas são dependentes, de modo que previsões não lógicas podem ser feitas. Por exemplo, dificilmente veríamos 10 etiquetas de repetições seguidas, ou várias repetições seguidas de vários recomeços. Entre esses mecanismos de decodificação estão o uso do algoritmo de Viterbi (BERTERO *et al.*, 2015), *beam-search* (WANG; NG; SIM, 2014), HMMs (HOUGH; SCHLANGEN, 2015; HOUGH; SCHLANGEN, 2017), e programação linear (ZAYATS; OSTENDORF; HAJISHIRZI, 2016).

Para incorporar o CRF após a última camada da rede neural, modificamos a função de ativação na Equação 4.12 de *softmax* para a função logística (*sigmoid*):

$$\mathbf{s}_t = \text{sigmoid}\left(\boldsymbol{\theta}^{(f)} \mathbf{u}_t + \mathbf{b}^{(f)}\right) \quad (6.1)$$

E passamos a saída da camada totalmente conectada em cada instante de tempo $\mathbf{s}_t$ para o modelo de CRF:

$$P(\mathbf{y} | \mathbf{s}) = \frac{\exp\left(\sum_{t=1}^{\varphi} f_t(\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{s})\right)}{Z(\mathbf{s})} \quad (6.2)$$

em que $f_t(a, b, \mathbf{s}) = \exp(\boldsymbol{\theta}_{a,b}^{(h)} \cdot \mathbf{s}_t)$ são funções que computam o potencial de duas etiquetas seguidas de acordo com a sequência de previsões da camada anterior $\mathbf{s}_t$ para um instante de tempo t . $Z(\mathbf{s})$ é uma constante de normalização que aplica a mesma função f para diferentes etiquetas do conjunto $y' \in \mathcal{Y}$. Desse modo, a Equação 6.2 retorna a probabilidade para cada palavra da sentença pertencer a cada uma das classes em $\mathcal{Y}$. Para integrar esse modelo em nossa RCNN, basta adicionar os pesos da camada de CRF linear $\boldsymbol{\theta}^{(h)}$ ao conjunto de pesos Θ à serem ajustados no treinamento, conforme visto na Subsubseção 4.2.2.5. Por fim, usamos o algoritmo de Viterbi para obter a melhor sequência de etiquetas $\mathbf{y}^*$ (MA; HOVY, 2016).

6.2.2 Baselines

Nossa primeira *baseline* para detecção de disfluência de edição baseia-se apenas em dicas lexicais. Criamos as seguintes regras simples para detectar se uma palavra faz parte de uma

repetição ou de uma revisão:

- A palavra w_i é repetição se for igual a palavra w_{i+1} (e.g. *foi a a irmã*)
- O bigrama (w_i, w_{i+2}) é repetição se for igual ao bigrama (w_{i+1}, w_{i+3}) (e.g. *foi ela que ela que chorou*)
- A palavra w_i é revisão se tiver o mesmo tamanho da próxima palavra e a distância de edição for menor que 3 (e.g. *uma um cavaleiro; ele elas rasgaram tudo*)

Não criamos nenhuma regra para detecção de recomeços pois eles não possuem dicas lexicais óbvias. É bastante claro que esse *baseline* pode ser melhorado com a inclusão de informações morfosintáticas. Para não deixar de criar um *baseline* sem essas informações, aplicamos o método de árvore de decisão usado anteriormente para detecção de preenchimentos, o qual usa informações lexicais e classes morfosintáticas de três palavras anteriores e três palavras posteriores para classificar w_i . Os resultados desses *baselines* são apresentados para cada tipo de disfluência de edição na [Tabela 38](#).

Tabela 38 – Resultados dos *baselines* para disfluências de edição.

<i>Baseline</i>	CTL			CCL			DA		
	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>
Chance	0.06	0.06	0.01	0.08	0.04	0.01	0.08	0.06	0.02
Regras de Rep. e Rev.	0.80	0.10	0.00	0.79	0.08	0.00	0.76	0.16	0.00
Árvore de Decisão	0.36	0.09	0.02	0.48	0.10	0.02	0.31	0.10	0.04

Fonte: Elaborada pelo autor.

Podemos ver na [Tabela 38](#) que, em geral, o *baseline* de regras obteve resultados muito melhores que os outros dois *baselines*, principalmente para disfluências de repetição. Como não usamos informações morfosintáticas para esse *baseline* então podemos dizer que as repetições estão fortemente relacionados com a identidade das palavras. Mas fica evidente que essas regras não são capazes de detectar revisões e recomeços. Além disso, usando apenas informações de identidade das palavras e classes morfosintáticas não parecem ser o suficiente para detectar esses dois últimos tipos de disfluências. Embora os *baselines* sejam métodos simples, é possível ver que é mais difícil detectar revisões e recomeços do que repetições, o que já é bem consolidado na literatura ([ZAYATS; OSTENDORF; HAJISHIRZI, 2016](#); [HOUGH; SCHLANGEN, 2017](#)). Os bons resultados do *baseline* de regras mostra que a adição das *features* na nossa RCNN pode ter sido uma boa escolha.

Ao contrário do que vimos para as tarefas anteriores, não há muita diferença entre os melhores resultados para cada grupo de paciente, sugerindo que esses tipos de disfluências ocorrem de maneira uniforme entre eles. Em média, os resultados do *baseline* de regras são

melhores que os da árvore de decisão, o que não é esperado pois o último método utilizou informações morfosintáticas também. Isso pode ser visto através da importância que a árvore de decisão deu às *features*: as quatro primeiras mais importantes são relativas a identidade das palavras $w_{i+1}, w_i, w_{i+3}, w_{i+2}$, e só depois é que a classe morfosintática da próxima palavra c_{i+1} aparece. Isso também indica que *features* posteriores são mais importantes do que *features* anteriores à palavra sendo analisada.

Nas próximas tabelas de resultados, os *baselines* apresentados correspondem aos melhores resultados (em negrito) encontrados aqui, para cada grupo de pacientes e para cada tipo de disfluência de edição.

6.2.3 Avaliação de Word Embeddings

Na avaliação de *word embeddings* para SS (Seção 5.2) mostramos que, em média, o melhor algoritmo de indução de *embeddings* foi o Skip-gram para vetores de 600 dimensões. Portanto, nesses experimentos decidimos alterar apenas a técnica utilizada: word2vec, wang2vec, e fastText, assim como fizemos para a detecção de preenchimentos. Variar essas técnicas é interessante pois cada uma foi proposta para capturar diferentes similaridades. O word2vec foi proposto originalmente para capturar similaridades semânticas, o wang2vec foi uma modificação para capturar informações sintáticas, e o fastText foi proposto para induzir vetor baseando-se na morfologia das palavras. Descrevemos cada uma das técnicas de indução de *embeddings* na Subseção 2.4.1, e apresentamos os resultados dessa variação na Tabela 39.

O corpus utilizado para a indução desses vetores é o mesmo utilizado para a avaliação de SS e detecção de preenchimentos. O corpus é composto por artigos da Wikipédia de 2016, artigos de notícias do portal G1² e textos do corpus PLN-BR (BRUCKSCHEN *et al.*, 2008). Realizamos pré-processamento de modo a deixar todas as palavras em caixa baixa e separamos os *tokens* de sinais de pontuações. Após esses procedimentos, o corpus resultante para a indução das *embeddings* retornou ~356M *tokens* dentre os quais ~1.7M eram distintos um do outro.

Tabela 39 – Resultados de variação de *embeddings* para disfluências de edição.

Técnica	CTL			CCL			DA		
	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>
Baseline	0.80	0.10	0.02	0.79	0.10	0.02	0.76	0.16	0.04
fastText	0.48	0.09	0.00	0.50	0.12	0.00	0.37	0.16	0.00
wang2vec	0.54	0.09	0.00	0.57	0.10	0.00	0.41	0.12	0.00
word2vec	0.52	0.15	0.00	0.50	0.20	0.00	0.36	0.11	0.00

Fonte: Elaborada pelo autor.

Quando levamos em consideração o F_1 para todos os tipos de disfluências de edição, o wang2vec desempenhou melhor que as outras técnicas. Isso é interessante, pois difere das

² <<http://g1.globo.com/>>

técnicas encontradas para SS e detecção de preenchimentos. Mais especificamente, a RCNN com CRF desempenhou melhor com o *wang2vec* para detecção de repetições em todas as transcrições, o *word2vec* obteve melhores resultados entre as técnicas e superou os *baselines* para detecção de revisões de CTL e CCL, e o *fastText* obteve melhores resultados apenas para a detecção de revisões em transcrições de DA, empatando com o *baseline*. Já para detecção de recomeços, todas as técnicas não conseguiram detectar pelo menos um recomeço, ficando até mesmo abaixo do *baseline*. Além disso, apesar do *wang2vec* ter superado as outras técnicas para detecção de repetições, apenas usando *embeddings* não é o bastante para superar o *baseline* de repetição de palavras. O fato do *wang2vec* ter obtido melhores resultados que as outras técnicas para repetições é interessante, pois repetições são trechos que não são bem formados sintaticamente, o que pode ter beneficiado nosso modelo.

Assim como vimos anteriormente com os *baselines*, podemos observar que em todos os casos a nossa RCNN-CRF obteve resultados melhores para repetições do que para as outras duas disfluências de edição. A diferença de desempenho para cada tipo de transcrição fica mais evidente usando *embeddings*, em que nossa RCNN-CRF obteve melhores resultados para transcrições de CTL e CCL do que para DA, sendo os resultados para CCL superiores aos outros dois, provavelmente pela quantidade de disfluências cometidas por esses pacientes. O desempenho baixo para recomeços é algo que vamos investigar com a importância de *features* a seguir, sendo que conforme vimos também com os *baselines*, acreditamos que isso deve-se principalmente ao fato de que *features* que baseiam-se na identidade das palavras não são muito discriminantes para a detecção desse fenômeno.

Nosso objetivo com esse experimento foi o de verificar qual tipo de *embedding* desempenha melhor para a tarefa de detecção de disfluências de edição: as que capturam similaridades semânticas (*word2vec*); sintáticas (*wang2vec*); ou morfológicas (*fastText*). Devido a diferença considerável entre os resultados de cada técnica, podemos concluir que similaridades sintáticas (*wang2vec*) têm mais impacto na detecção de repetições, e similaridades semânticas (*word2vec*) têm mais impacto na detecção de revisões. Entretanto, não conseguimos discernir com precisão qual é o tipo que mais influencia de modo geral, pois as diferenças entre as médias dos tipos de disfluência de uma técnica para o outra foram muito pequenas. Além disso, nenhuma das técnicas conseguiu acertar pelo menos um trecho de recomeço, o que significa que mais análises quanto ao melhor conjunto de *embeddings* deve ser feito.

Se tirarmos a média de todas as combinações de técnicas (*word2vec*, *wang2vec*, *fastText*) para os três tipos de pacientes e para os três tipos de disfluências, a técnica que retorna maior F_1 é o *wang2vec*. E foi justamente essa técnica (Skip-gram com 600 dimensões) que usamos em todos os próximos experimentos para detecção de disfluências de edição.

6.2.4 Importância das Features

Com o cenário de avaliação definido com *embeddings* de 600 dimensões induzidas pela técnica *wang2vec* com o algoritmo Skip-gram, decidimos avaliar nosso modelo em relação às outras *features*: manuais, morfossintáticas e prosódicas. A partir deste experimento, usamos informações prosódicas apenas com a combinação de *embeddings* com *features* manuais e morfossintáticas que desempenharam melhor em nossa RCNN-CRF. Então, avaliamos também as transcrições de DA até o momento em que utilizamos as *features* prosódicas, pois infelizmente não conseguimos realizar o alinhamento forçado entre as transcrições e o áudio para essas transcrições. A razão para a exclusão foi pelos áudios dos participantes com DA estarem com qualidade muito baixa, de modo que não era possível identificar o que o participante estava falando dado o volume da voz dos participantes e a presença de ruídos de fundo (as gravações foram feitas em um hospital).

A Tabela 40 apresenta os resultados de variação de *features* usando a métrica F_1 para transcrições de CTL, CCL e DA.

Tabela 40 – Resultados de cada conjunto de *features* para disfluências de edição.

<i>Features</i>	CTL			CCL			DA		
	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>	<i>Rep</i>	<i>Rev</i>	<i>Rec</i>
Baseline	0.80	0.10	0.02	0.79	0.10	0.02	0.76	0.16	0.04
PoS	0.32	0.00	0.00	0.39	0.00	0.00	0.34	0.00	0.00
Embeddings	0.48	0.09	0.00	0.50	0.12	0.00	0.37	0.16	0.00
Manuais	0.89	0.22	0.03	0.85	0.21	0.03	0.86	0.30	0.01
PoS + Embeddings	0.54	0.07	0.01	0.58	0.10	0.01	0.40	0.15	0.02
PoS + Manuais	0.89	0.35	0.03	0.85	0.35	0.03	0.86	0.37	0.05
Emb. + Manuais	0.83	0.26	0.03	0.83	0.25	0.03	0.78	0.19	0.01
PoS + Emb. + Man.	0.83	0.30	0.03	0.80	0.30	0.03	0.79	0.27	0.04
PoS + Man. + Pros.	0.83	0.35	0.03	0.80	0.23	0.02	-	-	-
Todas	0.79	0.31	0.01	0.79	0.27	0.02	-	-	-

Fonte: Elaborada pelo autor.

PoS corresponde às etiquetas morfossintáticas (*Part-of-Speech Tags*).

Na Tabela 40, podemos ver que a RCNN-CRF apresenta resultados de detecção de repetições levemente melhores para transcrições de CTL, e para detecção de revisões os melhores resultados foram obtidos para transcrições de DA. Porém, a diferença dos resultados entre os grupos é muito pequena, sugerindo que o método possui o mesmo desempenho independente das condições neuropsicológicas dos pacientes. Os *baselines* de repetição foram ultrapassados com uma margem considerável, 9% para CTL, 6% para CCL e 10% para DA.

Os *baselines* de revisões também foram superados, com uma margem larga (mais de 20%), para todas os tipos de transcrição. Já os resultados para detecção de recomeços continuam muito baixos, ultrapassando o *baseline* por apenas em 1% para cada tipo de transcrição, indicando que as *features* utilizadas não são discriminativas o suficiente para encontrar esse tipo de disfluência.

Portanto, como nosso modelo obteve resultados muito baixos para detecção de recomeços, discutiremos apenas os resultados para repetições e revisões, e de qualquer modo, estudaremos esses resultados com mais detalhes na análise de erros.

Analizando a importância dos diferentes conjuntos de *features*, percebemos que as *features* manuais tem um grande impacto na classificação para todos os grupos de pacientes, atingindo os melhores resultados para detecção de repetições. Informações de etiquetas morfossintáticas (PoS) ajudaram bastante para a detecção de revisões, e mantiveram o desempenho para detecção de repetições quando utilizadas em conjunto com *features* manuais. Ao contrário do que encontramos nas tarefas de SS e detecção de preenchimentos, as informações de *embeddings* não possuem um grande impacto em nosso modelo neural, pois quando usadas sozinhas não conseguem bons resultados; quando usadas com etiquetas morfossintáticas agregam apenas 4-8% de F_1 , e quando utilizadas em conjunto com etiquetas morfossintáticas e *features* manuais fazem o modelo desempenhar pior que utilizando as últimas duas, tanto para detecção de repetições como para detecção de revisões. Isso nos indica que, como observamos previamente, o que realmente tem impacto para detecção de repetições não são informações sobre a palavra em si *embeddings*, mas sim em relação as suas vizinhas. Além disso, comparando o *baseline* de repetição com as *features* manuais, podemos ver que a introdução das *features* de PoS duplicados, de prefixos e de fim de sentença ajudaram a RCNN-CRF.

Moniz, Isabel e Trancoso (2011) mostram que informações prosódicas como presença/ausência de pausas podem ser usadas com sucesso para identificar se uma sequência de palavras corresponde a uma região de disfluências ou não. Entretanto, nossos resultados mostram que informações prosódicas não contribuem para detecção de repetições e revisões. Sendo que o desempenho da RCNN-CRF até mesmo piorou quando essas informações foram acrescentadas. De fato, em ambos os tipos de transcrição, os melhores resultados foram obtidos usando $\alpha = 1.0$ da interpação linear da Equação 4.3, indicando que nosso modelo baseia-se inteiramente nas *features* textuais. Trabalhos relacionados recentes que usam informações prosódicas em conjunto com *features* textuais (FERGUSON; DURRETT; KLEIN, 2015; HOUGH; SCHLANGEN, 2017) mostram que a inclusão de informações prosódicas melhoram os resultados para detecção do enunciado original (sem distinguir o tipo) em menos de 1%. Isso sugere que apesar de informações prosódicas contribuírem para a detecção de disfluências de edição, ainda assim o seu impacto é baixo quando comparado com informações textuais. Além disso, assim como aconteceu para as avaliações das tarefas anteriores, acreditamos que a qualidade das gravações podem ter influenciado no peso dado para as informações prosódicas, já o corpus Cinderela v2 foi gravado no cenário clínico, em um hospital. Um trabalho futuro planejado para o projeto consiste em verificar algumas peculiaridades que podem ter influenciado os resultados, como: qualidade de gravação do áudio; natureza inerente dos pacientes (configuração clínica das gravações e estatísticas demográficas); insuficiência discriminativa das *features* prosódicas extraídas.

Não distinguindo entre os tipos e considerando apenas os acertos e erros para as disfluên-

cias de edição, os melhores resultados são os obtidos pela RCNN-CRF utilizando apenas as *features* manuais em conjunto com etiquetas morfossintáticas para todas as transcrições: CTL ($F_1 = 0.69$); CCL ($F_1 = 0.71$); e DA ($F_1 = 0.68$). Não distinguir entre os tipos de disfluências de edição significa em decidir se uma palavra faz parte ou não de um enunciado original disfluente, caracterizando um problema binário. Comparando esses nossos resultados com os resultados de trabalhos que usam apenas informações de contexto à esquerda (HOUGH; SCHLANGEN, 2015; HOUGH; SCHLANGEN, 2017), percebemos que nossos resultados estão no mesmo intervalo: F_1 entre 0.68 e 0.72. Porém, quando comparamos com resultados de trabalhos recentes no Switchboard-2 que, assim como o nosso, utilizam contexto à esquerda e à direita (QIAN; LIU, 2013; WANG; NG; SIM, 2014; ZAYATS; OSTENDORF; HAJISHIRZI, 2016; WANG *et al.*, 2017), percebemos que nossos resultados estão cerca de 18% abaixo. Acreditamos que isso aconteceu porque temos uma quantidade muito menor de dados (3M vs 24K). Isso é reforçado ao verificar que o trabalho que é estado-da-arte para o Switchboard-2 (WANG *et al.*, 2017) conseguiu apenas $F_1 = 0.51$ para um *corpus* da língua chinesa (25K sentenças), o qual é cerca de 12 vezes maior que o Cinderela v2 (2K sentenças).

Detectar disfluências de edição (e seus tipos) é uma tarefa que envolve inferências mais semânticas que detectar limites de sentenças e preenchimentos; nossas *features* não conseguiram discriminar bem o problema com apenas essa quantidade de dados. Isso é visto pelo fato de que nas tarefas anteriores as *embeddings* sozinhas obtiveram desempenho similar quando utilizadas com etiquetas morfossintáticas, e para detecção de disfluências de edição isso não aconteceu. Apesar de nossa arquitetura ter a capacidade teórica de verificar se uma *embedding* é igual a *embedding* vizinha, elas não desempenharam bem, provavelmente porque precisam de mais dados para serem ajustadas à detecção do enunciado original disfluente. Essas observações são apoiadas ao analisar o impacto das *word embeddings* também vistos por Wang, Che e Liu (2016), Zayats, Ostendorf e Hajishirzi (2016), nos quais as informações manuais também tiveram um grande impacto, porém para eles as *word embeddings* de fato melhoraram os resultados.

De acordo com (BARROS, 1999), os mecanismos de correção na língua falada são empregados para sanar tipos de erros diferentes: de pronúncia, morfossintáticos, ou semânticos-pragmáticos, que podem ser caracterizados como impropriedade de informação ou imprecisões nas expressões de sentimentos e opiniões dos falantes. Quando esses mecanismos são usados com marcadores discursivos de correção têm a função de fornecer pistas para que o ouvinte perceba a correção e, por meio delas, a intenção do falante, assumindo um papel na interpretação de uma conversação. Há correções que não empregam procedimentos de produção (pausas, repetições, prolongamentos, interrupções), nem marcadores discursivos. A ausência de marcadores de quaisquer tipo caracteriza, em geral, certas classes de correções: as que visam apenas a precisar as opiniões e intenções dos falantes e as que tentam reforçar a intenção do falante e, em ambos os casos, podem ser vistas como correções atenuadas.

Um trabalho futuro para poder esclarecer a relação dos três tipos de disfluências de edição

(repetição, revisão e recomeço) de acordo com a categorização linguística colocada acima por (BARROS, 1999) poderá ser realizado e ajudar a explicar também a razão das *word embeddings* não conseguirem ajudar a melhorar o desempenho em nosso corpus.

6.2.5 Comparação de Métodos

Usamos aqui o mesmo método de CRF usado para SS e detecção de preenchimentos, já que modelos de CRF também foram bastante empregados para detecção de disfluências de edição (LIU *et al.*, 2005; WANG *et al.*, 2010; BERTERO *et al.*, 2015). E para apoiar nossa escolha por um modelo recorrente bidirecional, separamos nosso método em outros três utilizados também para avaliações de SS (Seção 5.4):

- **MLP:** Removemos as camadas convolucionais e recorrentes, e adicionamos uma camada totalmente conectada oculta, de modo a formar uma MLP com uma camada oculta. Esse modelo é similar, mas não exatamente o mesmo, que um dos modelos usado por Bertero *et al.* (2015).
- **RNN:** Removemos a camada de convolução, de modo que passamos a entrada diretamente para a camada recorrente bidirecional. Esse modelo é similar, mas não exatamente o mesmo, que o usado por Hough e Schlangen (2015), Hough e Schlangen (2017).
- **CNN:** Removemos a camada recorrente bidirecional e passamos a saída da camada convolucional diretamente para a camada de saída.

Na subseção anterior percebemos as *features* prosódicas têm pouco impacto na classificação, de modo que a RCNN-CRF possui um desempenho superior quando apenas as *features* de classes morfossintáticas e as *features* manuais são utilizadas. Os resultados para cada método apresentado na Tabela 41 foram obtidos usando apenas essas duas *features*, e aproveitamos para realizar avaliações para transcrições de DA já que informações prosódicas não se mostraram benéficas.

Tabela 41 – Resultados de cada método para disfluências de edição.

Método	CTL			CCL			DA		
	Rep	Rev	Rec	Rep	Rev	Rec	Rep	Rev	Rec
Baseline	0.80	0.10	0.02	0.79	0.10	0.02	0.76	0.16	0.04
CRF	0.82	0.12	0.01	0.81	0.10	0.00	0.77	0.08	0.01
MLP	0.85	0.18	0.01	0.82	0.13	0.01	0.78	0.17	0.03
CNN	0.81	0.26	0.05	0.79	0.19	0.01	0.77	0.29	0.05
RNN	0.61	0.21	0.02	0.69	0.17	0.05	0.71	0.24	0.01
RCNN	0.85	0.27	0.06	0.81	0.23	0.04	0.77	0.30	0.02
RCNN-CRF	0.89	0.35	0.03	0.85	0.35	0.03	0.86	0.37	0.05

Fonte: Elaborada pelo autor.

Podemos ver que a nossa RCNN-CRF conseguiu os melhores resultados na detecção de repetições e de revisões para todas as transcrições, e a RNN os piores, até mesmo abaixo do *baseline* para as repetições. Em geral a RCNN e a MLP conseguiram atingir resultados melhores que a CNN, RNN e CRF. Além disso, como visto anteriormente, fica evidente a diferença dos resultados de todos os métodos entre repetições, revisões e recomeços, sendo que para recomeços os resultados continuam muito baixos, próximos ao *baseline*. Há também um grande salto entre a RCNN-CRF e os outros métodos para detecção de revisões em todas os grupos de pacientes, e também para detecção de repetições em transcrições de DA. Em geral, os métodos CRF, MLP, CNN e RCNN obtiveram resultados bem próximos para detecção de repetições, com 1% a 3% de diferença entre eles. Já para detecção de revisões a RCNN e a CNN tiveram desempenhos consideravelmente superiores aos outros dois (~10% ou mais).

Quando comparamos as combinações de métodos, percebemos claramente que a introdução do modelo de CRF em nosso modelo de RCNN original foi uma boa decisão, já que a RCNN-CRF obteve os melhores resultados para todas as transcrições com uma margem larga para detecção de revisões. É possível ver que o maior impacto para detecção de revisões provavelmente veio das camadas convolucionais e recorrentes, já que a CNN e a RNN apresentam melhores resultados que a MLP e CRF. Já para a detecção de repetições é difícil discernir, pois os métodos obtiveram resultados muito próximos. Assim como visto em outros trabalhos, a introdução de um decodificador na saída da rede neural parece melhorar o desempenho (QIAN; LIU, 2013; FERGUSON; DURRETT; KLEIN, 2015; WANG *et al.*, 2017), até mesmo para o nosso caso em que temos poucos dados de treinamento.

Por fim, podemos observar que a nossa RCNN com CRF atingiu os melhores resultados. Isso indica, primeiramente, que a união de um modelo convolucional com um modelo recorrente foi uma boa escolha para detecção de disfluências de edição em fala comprometida. E os bons resultados com a introdução de um CRF de cadeia linear como decodificador indicam que levar em consideração as classes das palavras vizinhas foi, como esperado, bastante benéfico para detectar a região de disfluência correspondente ao enunciado original.

6.2.6 Análise de Erros

Nas subseções anteriores vimos que a RCNN-CRF com *features* manuais e etiquetas morfossintáticas obteve os melhores resultados para detecção de repetições e revisões nos três tipos de transcrições. Entretanto, os resultados diferiram para cada tipo de disfluência: repetições foram identificadas com $F_1 = 0.85 - 0.89$, já revisões ficaram na faixa de $F_1 = 0.35 - 0.37$, e recomeços ficaram na faixa de $F_1 = 0.05$, muito mais abaixo que os outros. Para entender melhor o que nosso modelo aprendeu, mostramos os principais erros e acertos cometidos para cada tipo de disfluência de edição utilizando os melhores resultados da RCNN-CRF.

Repetições

Nosso método conseguiu resultados muito melhores para repetições do que para as outras duas disfluências. Isso deve-se principalmente pelo fato de que repetições possuem uma estrutura lexical bastante óbvia. Em geral, basta que duas palavras seguidas sejam iguais e a chance de ser uma repetição é alta; isso é ressaltado pelo nosso *baseline* de identidade de unigramas e bigramas de palavras. Entretanto, há muitos casos em que palavras duplicadas em sequência não representam repetições; buscamos em nosso corpus seis delas e apresentamos na [Tabela 42](#).

Tabela 42 – Exemplos de sequências de palavras duplicadas que não são repetições.

parecia **tudo tudo** muito bonito ...
e **não não** serve ...
tinham **muita muita** inveja da cinderela .
não sei o **que que** é ...
ficou uns dia **pensando pensando pensando** como ...
foi na festa **dançou dançou dançou dançou** .

Fonte: Elaborada pelo autor.

É possível ver na [Tabela 42](#) que possuem casos em que a mesma palavra é repetida mais de duas vezes e não caracteriza uma disfluência de edição, mas sim uma ênfase do interlocutor. A maioria dos casos é um advérbio ou um verbo que é repetido, o que é comum em fala espontânea. O fato de várias palavras se repetirem chama a atenção em relação ao tamanho do enunciado original disfluente. Para investigar se o nosso modelo possui algum viés em relação ao tamanho do enunciado original, apresentamos na [Tabela 43](#) os acertos e erros mais frequentes de acordo com o tamanho das repetições previstas pela nossa RCNN-CRF. Reportamos apenas para transcrições de CTL e CCL, pois o primeiro é um grupo de idosos saudáveis e outro é o grupo de idosos com a memória comprometida que analisamos também para SS ([Seção 5.6](#)).

Tabela 43 – Acertos e erros mais frequentes de acordo com o tamanho das repetições.

CTL			CCL		
TP	FP	FN	TP	FP	FN
a (12)	experimente (2)	do (2)	a (15)	que (4)	e (2)
o (10)	as (2)	uma (2)	e (14)	as (3)	ti (1)
as (9)	tem (1)	conheceu (1)	as (12)	não (2)	dev (1)
da (8)	que (1)	não (1)	um (9)	muito (2)	aqui (1)
e (8)	muitos (1)	cinderela (1)	da (9)	um (2)	morre (1)
e e (4)	foi foi (1)	uma ou (1)	a a (8)	a a (1)	a a (3)
da da (4)	um convite (1)	e as (1)	e e (4)	você vai (1)	que a (1)
a a (2)	dançaram e (1)	o sapato (1)	o o (3)	ela perdeu (1)	um funci (1)
as as (2)	um uma (1)	o sa (1)	os os (2)	e as (1)	e e (1)
que ela (2)	princesa não (1)	e le (1)	do do (2)	u u (1)	as a (1)
os os os (1)	com com uma (1)	foi foi espa (1)	uma uma carta (1)	que ela que (1)	que o ou (1)
em em em (1)	se iria se (1)	e e perg (1)	a a a (1)	pra pra pra (1)	as a a (1)
da da da (1)	dançou dançou dançou (1)	-	no no no (1)	a pessoa a (1)	na na na (1)
-	-	-	as as filhas (1)	-	não gostaram muito (1)
-	-	-	uns uns uns (1)	-	a as as (1)

Fonte: Elaborada pelo autor.

Podemos observar com a [Tabela 43](#) que os acertos (TP) para CTL e CCL são bastante semelhantes entre os dois grupos, para repetições com tamanho do enunciado 1, 2 e 3. A maioria desses acertos são artigos ou preposições repetidos como, por exemplo, “a a” que foi a repetição mais acertada para os dois grupos com tamanho 1. Já “e e e” e “a a a” foram os acertos mais frequentes para tamanho 2, e para os de tamanho 3, a RCNN-CRF cometeu mais erros do que acertos, sendo que os acertos geralmente refletiam quatro palavras duplicadas: “os os os os” ou “no no no no”. Casos como “uma uma carta” na verdade possui duas repetições em sequência, e não uma repetição cujo enunciado original possui tamanho 3: uma uma carta carta.

Quanto aos erros cometidos pela nossa RCNN-CRF, separamos eles em dois tipos: repetições que não deveriam ser marcadas como tais (FPs), e repetições que deveriam ter sido classificadas mas foram ignoradas (FNs). A respeito dos FPs, percebemos que os erros cometidos com tamanho 1 são compreensíveis, pois nesse caso as palavras repetidas também são comumente vistas em repetições: “as as”, “que que”, “muitos *ahn* muitos”, “não não”, apesar dos dois últimos também serem usados para ênfases. Para analisar os FPs de tamanhos maiores que 1, selecionamos alguns casos interessantes e mostramos o contexto no qual o erro foi cometido na tabela [Tabela 44](#), deixando a repetição prevista pelo nosso classificador em negrito.

Tabela 44 – Contextos de FPs para repetições de tamanho 2 e 3.

então foram foi foi espa foi espalhado
a princesa não não a princesa
apresentou uma um uma um vestido
que ia se iria se iria se ia se casaria com
o convite pra pra pra ela pra cinderela pra cinderela

Fonte: Elaborada pelo autor.

Podemos ver com base nos contextos que no primeiro exemplo a classificação correta seria “foi espa” como repetição de “foi espalhado”. No segundo caso o “não” deveria ter sido ignorada. No terceiro o podemos observar algo interessante: o “um” não é duplicação de “uma”, porém é prefixo dela. Isso nos diz que a *feature* manual de prefixo pode funcionar bem para casos como “madra madrasta”, porém também induz a erros em casos como esse, em que o correto seria uma revisão, pois o interlocutor está trocando o gênero do artigo. Mais erros semelhantes a esses foram encontrados para casos em que o interlocutor adicionava plural aos artigos (“a as a as”). O erro cometido para o quarto e o quinto exemplos são semelhantes: no quarto deveria ser uma revisão de “se iria” para “se ia”, e no quinto de “pra ela” para “pra cinderela”. Esses dois casos indicam que provavelmente a *feature* que verifica se a mesma palavra aparece *k* palavras adiante pode acabar atrapalhando o classificador. Como nesses casos as etiquetas morfossintáticas das palavras são as mesmas, fica difícil para a RCNN-CRF discernir entre revisão e repetição. Para melhorar o desempenho devemos acrescentar informações relativas a identidade da palavra em si, como as *word embeddings*, que infelizmente não retornaram bons resultados.

Quanto olhamos nas repetições de tamanho 1 que o nosso classificador ignorou (FNs),

percebemos que uma grande parcela delas são fáceis: “do do”, “uma uma”, “as as”. E outras já são mais ambíguas ou não são duplicadas: “não não”, “dev deveria”, “aqui aqui”, “ti tinha”. Para repetições maiores, decidimos mostrar o contexto nos quais os exemplos mais interessantes aparecem na [Tabela 45](#) com a região do enunciado original em negrito.

Tabela 45 – Contextos de FNs para repetições de tamanho 2 e 3.

as fe as feias rasgaram a
a que a a a madrastra alegou pra
quem o sapato é o sapato servisse
mais bonito ainda que o ou que o outro e verificando
as e as a a as suas irmãs muito invejosas olhando

Fonte: Elaborada pelo autor.

Com os contextos fica claro que algumas repetições são mais difíceis de serem identificadas que outras. Por exemplo, no primeiro exemplo o classificador precisa entender que “fe” é uma interrupção de “feitas”, já o segundo exemplo são simplesmente três palavras duplicadas, algo que o *baseline* lexical teria acertado. Já o terceiro exemplo mostra que erros da etapa anterior que consistem em remover os preenchimentos pode atrapalhar na classificação; nesse caso a palavra “é” não foi removida e ficou entre o bigrama “o sapato”, aumentando a distância da relação local entre o enunciado original e a correção. Os últimos dois exemplos, de tamanho 3, suportaram ainda mais o caso de prefixos como “ou” de “outro” são casos difíceis de serem interpretados, e o último exemplo nos mostra que é difícil até mesmo para um humano decidir se é mesmo de fato uma repetição o “a” seguido de “as”, pois apesar de um prefixo ser prefixo do outro, os dois são unidades diferentes porém com significados semelhantes.

Podemos ver na [Tabela 43](#) que a ênfase “dançou dançou dançou dançou” que mostramos na [Tabela 42](#) foi classificada erroneamente pelo nosso modelo, entretanto a ênfase “pensando pensando” não foi³, o que indica que nosso modelo simplesmente não marca palavras duplicadas em sequência como repetições. Isso fica ainda mais claro pelo fato de repetições como “a a a” ou “e e” terem sido ignoradas algumas vezes.

Ao analisar a matriz de cofusão percebemos também que repetições são confundidas com revisões em alguns casos, mais especificamente, verificamos que isso aconteceu 61 vezes, e que 30 vezes as revisões foram classificadas como repetições. Analisando os erros das repetições, percebemos que repetições são mais confundidas com revisões do que com regiões fluentes (61 vs 57 vezes). Além disso, podemos perceber com o aumento do tamanho das repetições na [Tabela 43](#) que a RCNN-CRF acerta mais quanto menor é o tamanho, por isso, apresentamos na [Tabela 46](#) resultados em F_1 para cada tamanho de repetição.

É possível ver na [Tabela 46](#) que repetições maiores que 2 são bastante raras em nosso corpus e também nas predições da RCNN-CRF, o que indica que nosso modelo simplesmente

³ Não aparece na tabela como TP porque ocorre apenas uma vez no corpus de CCL.

Tabela 46 – Resultados para cada tamanho de repetição.

Tamanho	CTL				CCL			
	# Orig / Pred	P	R	F ₁	# Orig / Pred	P	R	F ₁
Tam. 1	135 / 125	0.85	0.92	0.88	155 / 159	0.85	0.83	0.84
Tam. 2	50 / 47	0.84	0.89	0.87	68 / 57	0.68	0.81	0.74
Tam. 3	5 / 6	0.60	0.50	0.55	12 / 8	0.42	0.62	0.50
Tam. 4	4 / 5	0.75	0.60	0.67	3 / 1	0.00	0.00	0.00

Fonte: Elaborada pelo autor.

não classifica tudo como repetição. Apesar dos tamanhos 3 e 4 terem poucas instâncias, podemos afirmar, a grosso modo, que nossa RCNN-CRF tem melhor desempenho para repetições pequenas, o que é esperado e também foi mostrado por [Hough e Schlangen \(2017\)](#).

Revisões

As revisões foram o segundo tipo de disfluência de edição que nossa RCNN-CRF acertou mais vezes, porém cometeu mais erros do que acertos. Vimos anteriormente que nossos melhores resultados para revisões ficaram na faixa de $F_1 = 0.35 - 0.37$, o que é um bem mais abaixo que o obtido para repetições, talvez pelo fato de que revisões precisam de mais conhecimento semântico que repetições. Nosso modelo acertou apenas um único caso em que o tamanho do enunciado era igual a 1 (“entregaram **um** uma carta”) para transcrições de CTL, e nenhum para transcrições de CCL. Para verificar os erros cometidos e a limitação semântica da nossa RCNN-CRF, mostramos na [Tabela 47](#) os cinco erros e acertos mais frequentes e seus respectivos contextos de acordo com o tamanho das revisões, combinando os FPs e FNs de transcrições de CTL e CCL. Decidimos mostrar o contexto para revisões pois ao contrário das repetições, a região à direita geralmente não é igual a região sendo analisada (e.g. “**pra ela** pra ela” vs. “**pra ela** pra cinderela”).

Tabela 47 – Contextos de erros e acertos mais frequentes de acordo com o tamanho das revisões.

TP	FP	FN
entregaram um uma carta - - - -	dançaram e dançaram e dançaram pra ela pra cinderela ela que ela que ela ia de casa em casa dela da menina da princesa	anunciou uma um baile ai ela elas sempre uma ocasião havia houve um experimental todas em todas as irmãs jogaram estragaram
que as as madrinhas as fadas madrinhas tinha uma fada uma fadinha aqui casou e ficou e ficaram felizes ela quer casar queria casar ela perdeu o chinelos o sapato de cristal	na casa da madrasta da cinderela cinderela era uma vez uma cinderela que ela fugiu ela tava trancada pessoal começou a dançar a valsa ficou com o que com o sapatinho	ela veio a branca de neve veio uma uma menina uma garota que vai ter uma festa um baile madrasta rasgou toda a toda roupa rasgou toda a sua o seu vestido
a madrastra e três filhas e duas filhas as duas enteadas as duas filhas da madrastra um dia num outro reinado num outro castelo que tinha que a cinderela que a fada deu ai a madrastra ai o príncipe ordenou	e andou pra cá andou pra lá arrancou o colar dela arrebitaram o colar dela humilde tava com os pais vivia com os pais ele queria conhecer a queria conhecer a moça que ia dar o vestido ia dar uma carruagem	ofereceu um baile o príncipe ia oferecer pediu as mandantes dele os servos dele limpeza na casa toda no castelo alias um dia o pai morreu a mãe morreu uma das amigas uma das colegas dela

Fonte: Elaborada pelo autor.

Analisando os acertos é possível ver que a maioria deles possui uma estrutura semelhante: a primeira palavra do enunciado original é igual à primeira palavra da correção, e as últimas geralmente refletem uma mudança de verbo ou substantivo. Essas dicas lexicais provavelmente ajudaram a RCNN-CRF em categorizar corretamente essas disfluências, e como revisões de tamanho um não possuem essas dicas, elas acabaram sendo ignoradas ou classificadas como repetições (como vimos anteriormente). Acreditamos que para solucionar esse problema é possível utilizar informações referentes a identidade da palavra em si, como *word embeddings*, porém necessitamos de mais dados para elas serem ajustadas ao nosso cenários. Por exemplo, usando o *wang2vec* com Skip-gram de 600 dimensões, a palavra mais próxima de “as” é “a” e vice-versa, e o mesmo ocorre para “um” e “uma”, “o” e “os” e “ela” e “ele”. Mas no nosso cenário em que o interlocutor explicitamente corrige o gênero do artigo é necessário uma representação que consiga distinguir entre esses casos. É possível fazer isso com as *embeddings* ao ajustá-las conforme o treinamento (como de fato fizemos), porém é necessário uma quantidade considerável de dados para que isso aconteça. Outra estratégia é utilizar *features* manuais que digam se essas palavras são do gênero feminino/masculino e se são plurais ou não.

Quando olhamos para as revisões que a RCNN-CRF marcou em lugares errados (FPs), é possível ver que os erros de tamanho 1 foram cometidos porque eram repetições (“que ela que ela”), ou porque foi marcado apenas uma parte da região de disfluência: foi marcado “**pra ela** pra cinderela” ao invés de “**pra ela** pra cinderela”. Outros erros são em virtude da ambiguidade do fenômeno, como em “de casa em casa”, no qual analisando apenas esse trecho parece que as duas primeiras palavras foram corrigidas pelas duas últimas, entretanto é uma expressão multipalavras utilizada no dia a dia. Quando aumentamos o tamanho das revisões é possível ver que os erros cometidos são consequências da ambiguidade do fenômeno, por exemplo “na casa **da madrasta** da cinderela” poderia ser reescrito como “na casa da cinderela” sem perdermos a estrutura semântica e sintática. Outro exemplo de ambiguidade de expressões multipalavras é o “**andou pra cá** andou pra lá”, o qual apresenta a estrutura típica dos acertos da RCNN: trecho inicial do enunciado original igual ao da correção. Nos casos em que temos revisões de tamanho três, podemos ver que o a palavra anterior ou posterior a região identificada também deveria ter sido incluída, como em “tava **com os pais** vivia com os pais” e “ia **dar o vestido** ia dar a carruagem”, isso indica que talvez devêssemos aumentar o tamanho do filtro de convolução da rede para 9 ou 11 (usamos 7 em nossos experimentos) e aumentar o número de palavras vizinhas que devem ser consideradas para ± 5 ou ± 6 (usamos ± 4 em nossos experimentos).

Com as revisões que nosso modelo deixou de classificar (FNs), fica bastante óbvio que a RCNN-CRF sofre com revisões de tamanho 1, nas quais não há repetições de palavras no início/fim da região do enunciado original e da correção (“**ela** elas”, “**havia** houve”, etc.). Ao aumentar o tamanho das revisões podemos ver que nosso modelo erra bastante quando há pouca similaridade lexical entre os trechos do enunciado original e da correção, por exemplo: “**ela** **veio** a branca de neve veio” há apenas a palavra “veio” em comum e somente após uma distância de 4 palavras; em “**uma festa** um baile” nenhuma palavra é igual a outra, embora “um” e “uma” sejam

ambos o mesmo artigo com gêneros diferentes, e o mesmo ocorre para “**a sua** o seu vestido”. Os casos com tamanho 3 mostram que conforme o tamanho das revisões aumentam, mais difícil fica identificá-las, pois há casos ignorados em que há as mesmas palavras no início/final das regiões do enunciado original e da correção (“**o pai morreu** a mãe morreu”, “**uma das amigas** uma das colegas dela”). Acreditamos que isso deve-se principalmente ao baixo número de vezes que revisões de tamanho 3 ocorrem (apenas 49 vezes: 17 CTL + 13 CCL + 19 DA), um valor ~3 vezes menor que o encontrado no *córpus* Switchboard-2 (HOUGH; SCHLANGEN, 2017).

Devemos lembrar também que nosso cenário produz uma fala bastante comprometida, com disfluências bastante complexas que acabam contribuindo para o número de erros. Por exemplo, a seguinte disfluência apresenta três revisões em sequência: “**roubaram arrancou o colar dela arrebutaram o colar dela** tiraram o colar puxaram a roupa dela”, o que é algo bastante difícil de acontecer em fala espontânea normal.

Ao analisar a matriz de confusão, percebemos também que revisões são confundidas com repetições em alguns casos, mais especificamente, verificamos que isso aconteceu 30 vezes. Mas a maioria dos erros resultam de revisões ignoradas por nossa RCNN-CRF (452 vezes), enquanto acertos representam menos da metade desse valor (206 vezes). Para elucidar melhor as descobertas em relação ao tamanho das regiões do enunciado original, apresentamos na [Tabela 48](#) resultados em F_1 para cada tamanho de revisão.

Tabela 48 – Resultados para cada tamanho de revisão.

Tamanho	CTL				CCL			
	# Orig / Pred	P	R	F_1	# Orig / Pred	P	R	F_1
Tam. 1	54 / 16	0.02	0.06	0.03	52 / 27	0.00	0.00	0.00
Tam. 2	63 / 51	0.33	0.41	0.37	28 / 27	0.25	0.26	0.25
Tam. 3	17 / 12	0.18	0.25	0.21	13 / 8	0.23	0.38	0.29
Tam. 4	6 / 2	0.00	0.00	0.00	4 / 6	0.00	0.00	0.00

Fonte: Elaborada pelo autor.

É possível ver na [Tabela 48](#) que, ao contrário do que aconteceu para as repetições, a nossa RCNN-CRF acerta mais para revisões de tamanho 2 e 3 do que em para revisões de tamanho 1. Esses resultados mostram que nosso modelo praticamente não consegue detectar revisões de tamanho 1 e 4+. Apesar dos tamanhos 3 e 4 terem poucas instâncias, podemos afirmar, a grosso modo, que nossa RCNN-CRF tem melhor desempenho para revisões maiores, em que há estrutura lexical presente.

Recomeços

Recomeço é a categoria na qual a nossa RCNN-CRF obteve os piores resultados, sendo que nenhum deles passou de 5% de F_1 . O que queremos mostrar aqui é o porquê dos recomeços terem sido tão mal classificados, e para isso elencamos duas hipóteses principais:

- Eles são muito raros em nosso corpus: há um total de 149 palavras que são recomeços, o que representa apenas 8% de disfluências de edição em nosso corpus.
- Eles são naturalmente difíceis: vários trabalhos já mostraram isso (SNOVER; DORR; SCHWARTZ, 2004; BERTERO *et al.*, 2015; WANG *et al.*, 2017; HOUGH; SCHLANGEN, 2017). É possível afirmar que tanto o enunciado original quanto o região de correção parecem bastante fluentes: “e a fada então quando as irmãs”, ou “depois ela acho que virei uma página a mais”. Além disso, geralmente não há dicas lexicais que marcam o início e o fim de uma região de recomeço. Acreditamos que esse é o principal motivos pelo qual os trabalhos relacionados não reportam resultados apenas para essa categoria e preocupam-se em apenas identificar a região dos enunciados originais, ou encontrar o intervalo entre duas palavras que representa o ponto de interrupção (LIU *et al.*, 2005; MEDEIROS *et al.*, 2013).

Como testamos vários métodos estado-da-arte e várias *features* diferentes também utilizadas por trabalhos relacionados, acreditamos que as duas hipóteses explicam os baixos resultados encontrados, e assim não há problemas com a metodologia aplicada. Entretanto, há métodos que podem ser aplicados para tentar melhorar o desempenho para recomeços. Um deles é o método aplicado por Bertero *et al.* (2015), que consiste em primeiramente identificar e remover repetições e revisões para depois detectar recomeços, separadamente, em que os autores mostraram que o desempenho foi de 44% para 52% de F_1 , porém quando comparados, a grosso modo, com os nossos, eles ainda conseguiram bons resultados com o primeiro passo. Os trabalhos de Zayats, Ostendorf e Hajishirzi (2016), Wang *et al.* (2017), Hough e Schlangen (2017) mostraram apenas a diferença dos resultados entre repetições e a média de outros tipos de disfluência de edição, sem separar os resultados para recomeços.

Problema binário sem recomeços

Para analisar em números reais o desempenho do nosso modelo ao ignorar as diferenças entre os tipos de disfluências e ignorando os recomeços (i.e. tratando-os como fluência), apresentamos na Tabela 49 os resultados em termos de F_1 para cada grupo de pacientes.

Tabela 49 – Resultados para detecção binária de disfluências de edição.

Grupo	Com recomeços			Sem recomeços		
	P	R	F_1	P	R	F_1
CTL	0.77	0.61	0.69	0.76	0.64	0.70
CCL	0.80	0.64	0.71	0.80	0.71	0.75
DA	0.83	0.59	0.69	0.82	0.65	0.73

Fonte: Elaborada pelo autor.

Na [Tabela 49](#) podemos ver que remover os recomeços aumenta o desempenho da RCNN-CRF em 1% para CTL, e 4% para CCL e DA. Ao fazer isso os valores de precisão diminuíram 1% para CTL e DA, e os de cobertura aumentaram 3% para CTL, 6% para CCL e DA. Isso sugere que se não nos importarmos com o tipo específico da disfluência de edição, de modo que seja interessante apenas saber se pertence ou não a uma região de disfluência, então a RCNN-CRF é capaz de obter um desempenho superior, pois os erros de substituição entre repetições e revisões são considerados acertos, e os erros de recomeços são desconsiderados.

Com esses resultados percebemos que anotar os recomeços mais atrapalharam a RCNN-CRF do que a beneficiaram, e portanto sugerimos que a categoria de recomeços deva ser retirada para a detecção de disfluências de edição no método final até que haja mais dados para treinamento do modelo proposto.

6.3 Avaliação Extrínseca no Coh-Metrix-Dementia

Após obter as transcrições segmentadas e com as disfluências removidas automaticamente, passamos elas para a próxima ferramenta no *pipeline* do projeto, o Coh-Metrix-Dementia. As transcrições automáticas passaram por todo o processo da [Figura 22](#), que consiste em:

1. Usar a RCNN para segmentar a transcrição original em sentenças;
2. Combinar o método de lista de palavras para remover pausas preenchidas e a RCNN para remover marcadores discursivos e a pausa preenchida “é”;
3. Passar a transcrição segmentada sem pausas preenchidas e marcadores discursivos para a RCNN-CRF, a qual então remove repetições e revisões.

O primeiro e o segundo passo usam *word embeddings*, classes morfofssintáticas e informações prosódicas para a detecção. Já o último passo utiliza apenas classes morfofssintáticas e *features* manuais. É importante informar que, nessa análise, as transcrições originais são as que foram revisadas para a anotação da pausa preenchida “é”, já que obtivemos resultados melhores para a detecção de preenchimentos com a nova anotação ([Subseção 6.1.5](#)).

Testamos também a anotação revisada na detecção de disfluências de edição com a melhor configuração da RCNN-CRF e obtivemos um $F_1 = 0.88$ para repetições e $F_1 = 0.35$ para revisões em transcrições de Controle, $F_1 = 0.85$ para repetições e $F_1 = 0.35$ para revisões em transcrições de CCL, e $F_1 = 0.85$ para repetições e $F_1 = 0.36$ para revisões em transcrições de DA. Esses novos resultados são praticamente os mesmos que encontramos anteriormente, com uma leve perda de desempenho de apenas 1% para repetições de CTL e DA. Isso indica que a revisão da anotação de preenchimentos e consequentemente a remoção manual desses preenchimentos não afetaram o método de detecção de disfluências de edição.

Algumas das métricas usadas em estudos de avaliação automática da fala na área clínica são influenciadas pelo método usado para segmentar a transcrição e remover as disfluências dela, pois essas métricas dependem da robustez do *parser* sendo utilizado e das características da anotação dos *corpus* usados para treinar esses *parsers*. Em nosso estudo com narrativas em português brasileiro, as métricas sintáticas extraídas do Coh-Metrix-Dementia usaram o *parser* de dependências MALT-Parser (NIVRE; HALL; NILSSON, 2006), e o *parser* de constituintes LX-Parser (SILVA; BRANCO; GONÇALVES, 2010). O primeiro foi o *parser* com melhor desempenho e a escolha pelo segundo foi pelo fato dele ser o único *parser* gratuitamente disponível para a língua portuguesa.

Nesse estudo, pretendemos avaliar se a segmentação e remoção de disfluências automáticas do DeepBonDD influenciaram no cálculo das métricas sintáticas. Para isso, selecionamos 9 métricas sintáticas do Coh-Metrix-Dementia para verificar se houve diferença significativa entre as transcrições manuais e as geradas automaticamente pelo DeepBonDD. De acordo com (CUNHA, 2015), as 9 métricas selecionadas podem ser descritas como:

- **Complexidade de Yngve:** Calcula-se a complexidade sintática de cada sentença, e então calcula-se a média desses valores. No cálculo da complexidade, utiliza-se a média dos escores das palavras (YNGVE, 1960).
- **Complexidade de Frazier:** Calcula-se a complexidade sintática de cada sentença, e então calcula-se a média desses valores. No cálculo da complexidade, utiliza-se o máximo da soma de trigramas dos escores das palavras (FRAZIER, 1985).
- **Distância de dependência:** Média da distância de dependência das sentenças de um texto. Para cada sentença do texto, a distância de dependência é calculada como a soma da distância entre as palavras associadas a cada relação de dependência. As árvores de dependência são extraídas pelo MaltParser⁴.
- **Média de cláusulas por sentença:** Calcula-se o número médio de cláusulas por sentença. Considera que uma cláusula em uma sentença é caracterizada pela presença de um sintagma verbal.
- **Incidência de sintagmas nominais:** Proporção de sintagmas nominais em relação à quantidade de palavras do texto, dividida por 1000. As árvores de constituintes são extraídas pelo LX-parser.⁵

⁴ <http://www.maltparser.org/malteval.html>

⁵ Essa métrica está passando por uma revisão, pois, NPs dentro de NPs são contados da mesma forma que NPs independentes. Sua correção depende também da saída do parser LX-Parser

- **Modificadores por sintagmas nominais:** Proporção média de modificadores por sintagma nominal nas sentenças. Considera como modificadores adjetivos, advérbios e artigos que participam de um sintagma⁶;
- **Pronomes por sintagmas nominais:** Proporção média de pronomes por sintagma nominal nas sentenças;
- **Número de sentenças:** Número de sentenças no texto;
- **Palavras por sentenças:** Número de palavras dividido pelo número de sentenças;

Em resumo, há várias dependências para a corretude do cálculo das métricas complexidade de Yngve (YNGVE, 1960), complexidade de Frazier (FRAZIER, 1985), média de cláusulas por sentença, incidência de sintagmas nominais, modificadores por sintagmas nominais, e pronomes por sintagmas nominais. As duas primeiras dependem do sucesso de se analisar a árvore como um todo; a terceira depende da identificação correta de sintagmas verbais; e as últimas três dependem na identificação correta de sintagmas nominais⁷. Palavras por sentenças e o número de sentenças são correlacionadas, já que um maior número de sentenças indica que elas possuem um tamanho menor.

Os resultados são apresentados na Tabela 50. As comparações foram conduzidas com o teste não paramétrica Wilcoxon rank-sum (também conhecido como Mann-Whitney-Wilcoxon), com um nível de significância de 5% ($p < 0.05$). A hipótese nula é a de que as métricas possuem médias iguais para as transcrições manuais e automáticas. Os valores mostrados na tabela são as médias (com desvio padrão), e o valor de p . Os resultados em negritos denotam diferenças estatísticas significativas entre as transcrições manuais e automáticas.

Podemos perceber que para pacientes de controle nenhuma métrica apresentou diferença estatística significativa. Para CCL apenas três métricas apresentaram: média de cláusulas por sentença, distância de dependência e palavras por sentença. Já para DA, além das três métricas anteriores, outras duas apresentaram diferença: complexidade de Yngve e complexidade de Frazier. Como esperado, a degradação do desempenho do método é evidente conforme nos movemos de uma fala mais comportada (CTL) para uma fala mais comprometida (CCL e DA). Entretanto, os resultados para CCL ainda podem ser considerados bons, já que apenas 3 métricas apresentaram diferença significativa. Focamos nossa análise nessas métricas para CCL, pois as transcrições de CCL passaram por todos os passos do DeepBonDD com o auxílio de informações prosódicas.

A métrica palavras por sentença provavelmente teve bastante diferença, pois nosso segmentador tem a tendência de colocar mais pontos finais do que a anotação manual, o que

⁶ Essa métrica está passando por uma revisão, pois a sua implementação correta deveria considerar somente adjetivos como modificadores de sintagmas nominais. Depende também da saída do parser LX-Parser.

⁷ .

Tabela 50 – Avaliação de métricas sintáticas no Coh-Metrix-Dementia.

Métrica	CTL			CCL			DA		
	Auto	Manual	<i>p</i>	Auto	Manual	<i>p</i>	Auto	Manual	<i>p</i>
Complexidade de Yngve	2.06 (0.10)	2.11 (0.16)	0.29	2.06 (0.14)	2.15 (0.15)	0.13	1.77 (0.24)	1.91 (0.20)	0.02
Complexidade de Frazier	7.14 (0.24)	7.20 (0.29)	0.68	7.07 (0.30)	7.25 (0.27)	0.10	6.44 (0.51)	6.82 (0.45)	0.02
Média de cláusulas por sent.	1.88 (0.26)	2.06 (0.37)	0.23	1.86 (0.28)	2.22 (0.41)	0.01	1.39 (0.45)	1.72 (0.36)	0.01
Incidência de SN	297.30 (74.18)	296.12 (76.64)	0.84	273.78 (93.44)	311.68 (29.24)	0.19	241.15 (146.29)	267.17 (118.46)	0.62
Modificadores por SN	0.39 (0.07)	0.40 (0.06)	0.95	0.40 (0.08)	0.40 (0.07)	0.87	0.36 (0.09)	0.36 (0.12)	0.81
Distância de dependência	34.01 (7.22)	40.26 (11.68)	0.16	34.29 (7.11)	48.01 (14.58)	0.00	23.52 (12.37)	31.53 (10.94)	0.00
Pronomes por SN	0.25 (0.08)	0.23 (0.07)	0.59	0.23 (0.07)	0.23 (0.07)	1.00	0.20 (0.08)	0.21 (0.10)	0.91
Palavras por sentença	11.25 (1.68)	12.53 (2.55)	0.17	11.47 (2.09)	14.05 (3.04)	0.01	8.23 (3.59)	10.18 (2.80)	0.01
Número de sentenças	32.70 (15.57)	30.00 (14.29)	0.47	30.90 (14.92)	26.90 (14.10)	0.24	39.70 (22.95)	33.90 (23.39)	0.22

Fonte: Elaborada pelo autor.

SN: Sintagma nominal

vimos anteriormente com a maior quantidade de falsos positivos em relação a falsos negativos. Além disso, a remoção de disfluências eliminou menos palavras do que o necessário, como também foi visto com o maior número de falsos negativos em relação ao de falsos positivos. Esses dois fenômenos fazem com que o número de palavras por sentenças das transcrições automática sejam diferentes do que os das manuais, e portanto, a métrica de palavras por sentenças retornou diferença significativa. E esse mesmo comportamento em relação a distribuição de sentenças acaba afetando também a média de cláusulas por sentença e distância de dependência, as quais também são calculadas de acordo com o número de sentenças em um texto.

Entretanto, as médias entre as transcrições automáticas e manuais de CCL para essa métrica mostram que os valores são próximos (de 11 a 14 palavras), o que significa que apesar de ter diferença significativa, essa remoção não afeta a estrutura sintática do método, já que as duas métricas sintáticas (complexidade de Yngve e Frazier) não apresentaram diferenças significativas. Por exemplo, mostramos no [Apêndice C](#) uma transcrição inteira de CCL manual e após passar por passos do DeepBonDD, e mostramos abaixo apenas um trecho manual seguido pelo mesmo trecho com SS e DD automáticas:

Manual: e quando chegou na casa da madraستا todas tentaram colocar o sapato mas em vão o pé muito grande e nenhuma entrou . quando chegou a vez da cinderela o sapato coube direitinho . ai todos ficaram felizes no palácio . e ela contente casou e ficaram felizes para sempre .

Auto: e quando chegou na casa da madrasta todas tentaram colocar o sapato . mas em vão o pé muito grande . e nenhuma entrou . quando chegou a vez da branca de neve não da cinderela . o sapato coube direitinho . ai todos ficaram felizes no palácio . e ela contente casou e ficou e ficaram felizes para sempre .

É possível perceber que o trecho automático possui mais limites de sentenças que o manual, geralmente após conjunções (*mas*, *e*), resultando em sentenças bem formadas, porém não de acordo com a anotação manual. Além disso, podemos ver que a disfluência de edição “*da branca de neve não da cinderela*” não foi eliminada, o que além de mudar o tamanho da sentença, acabou criando uma sentença nova: “*o sapato coube direitinho*”. Ou seja, dependendo de como as sentenças são segmentadas, as métricas suscetíveis ao número de sentenças podem apresentar pequenas variações em relação a transcrição original. A remoção de disfluências também acaba afetando essas métricas, principalmente se forem disfluências longas, eliminando ou deixando mais palavras na sentença e consequentemente afetando ainda mais a distribuição delas. Analisando especificamente os casos de divergência, é possível observar que, apesar das variações de limites de sentença e de disfluências, o DeepBonDD é capaz de gerar uma transcrição que pode ser considerada inteligível. No trecho automático acima, apenas a sentença “quando chegou a vez da branca de neve não da cinderela” não é bem formada sintaticamente.

6.4 Considerações Finais

Primeiramente, avaliamos a nossa RCNN para a tarefa de detecção de preenchimentos (pausas preenchidas e marcadores discursivos) com várias abordagens diferentes.

Para apoiar a escolha de um modelo baseado em redes neurais recorrentes convolucionais, analisamos o desempenho de diferentes métodos de classificação baseados em redes neurais, sendo que nossa RCNN obteve os melhores resultados na detecção de marcadores discursivos para CTL e CCL, e empatou com o CRF para detecção de pausas preenchidas de CCL. Assim como aconteceu nas avaliações de SS, a nossa RCNN atingiu em média os melhores resultados, implicando que a união de um modelo convolucional com um modelo recorrente foi uma boa escolha para detecção de preenchimento em fala comprometida. Com a análise de erros, mostramos que nossa RCNN baseia-se fortemente na identidade das palavras para a detecção de ambos preenchimentos. Por fim, mostramos que a melhor escolha para a detecção de preenchimentos é combinar lista de palavras para a detecção de pausas preenchidas e nosso modelo de RCNN para detecção de marcadores discursivos, e caso deseja-se obter um maior desempenho, é possível treinar outro modelo apenas para decidir o caso da pausa preenchida “é”, pois ela é muito ambígua e ocorre frequentemente em fala comprometida. Utilizando essa estratégia e ao revisar a anotação para a pausa preenchida “é”, obtemos em média um F_1 entre

0.91 e 0.92 para detecção de preenchimentos tanto para transcrições de CTL quanto para de CCL.

Para a detecção de disfluências de edição propusemos duas alterações na RCNN:

1. adicionamos *features* manuais informando se unigramas/bigramas de palavras e etiquetas morfossintáticas são iguais a unigramas/bigramas vizinhos, adicionamos também *feautres* que informam se uma palavra é prefixo de palavras posteriores, e se uma palavra está próxima de uma final de sentença;
2. a segunda modificação foi a inclusão de um modelo CRF de cadeia linear com o algoritmo de Viterbi na última camada da nossa RCNN (formando assim o modelo RCNN-CRF). Com os *baselines* verificamos que informações lexicais são bastante impactantes para detecção de repetições.

Assim como fizemos para as tarefas anteriores, para apoiar a escolha de um modelo baseado em redes neurais recorrentes convolucionais, analisamos o desempenho de diferentes métodos de classificação baseados em redes neurais. Com essa análise ficou evidente que a introdução do modelo de CRF em nosso modelo de RCNN original foi uma boa decisão, já que a RCNN-CRF obteve os melhores resultados para detecção de repetições e revisões para todos os grupos de pacientes. Com os experimentos, os verificamos que os resultados oscilaram bastante para cada tipo de disfluência: repetições foram identificadas com $F_1 = 0.85 - 0.89$, já revisões ficaram na faixa de $F_1 = 0.35 - 0.37$, e recomeços ficaram na faixa de $F_1 = 0.05$, muito mais abaixo que os outros.

CONCLUSÕES

Este capítulo retoma os objetivos deste trabalho, apresentando uma visão geral sobre o trabalho desenvolvido e apresenta as contribuições, limitações e os trabalhos futuros.

7.1 Contribuições

Para permitir a funcionalidade correta de ferramentas de PLN que analisam transcrições de fala potencialmente comprometida, o projeto DeepBonDD foi concebido para tratar de duas tarefas: processar a saída de sistemas de RAV e gerar textos adequados ao processamento de ferramentas para diagnóstico de demências. Como não temos ainda um sistema RAV para trabalhar com fala comprometida em português, neste trabalho as transcrições foram feitas manualmente e um procedimento de alinhamento forçado foi realizado entre o áudio e o texto para a extração de *features* prosódicas.

O DeepBonDD é responsável pela segmentação de sentenças (SS) e detecção de disfluências (DD) de transcrições de fala comprometida, avaliando ambas as tarefas intrinsecamente, e também extrinsecamente no Coh-Metrix-Dementia após recuperar limites de sentenças e remover as disfluências das transcrições.

Antes de propor um método para as duas tarefas, buscamos a definição das tarefas na literatura e mostramos que há diferenças entre sentenças delimitadas acusticamente e sentenças linguísticas. E, então, apresentamos a tarefa de SS em conjunto com outras tarefas similares, que buscam recuperar todos sinais de pontuação ou desambiguar os já existentes. Para a tarefa de DD, apontamos primeiramente as diferenças entre fluência e disfluência, e então ilustramos com exemplos a categorização de disfluências mais aceita na literatura ([SHRIBERG, 1994](#)), a qual é dividida em:

- preenchimentos: pausas preenchidas (*ahn, eh*); marcadores discursivos (*né, bom*), e termos de edição explícitos (*não, digo*); e

- disfluências de edição: repetições (<pra ela> pra ela ir ao castela); revisões (<uma filha> duas filhas); recomeços (<ela tá> aqui é uma noiva); e disfluências complexas, que abrangem as três anteriores contidas em outras.

Buscamos também a descrição de *corpus* utilizados frequentemente em trabalhos relacionados da língua inglesa, expondo propriedades como número de *tokens* e horas de duração das gravações. Para não deixar dúvidas acerca do método proposto, explicamos sucintamente as representações lexicais e prosódicas utilizadas. E, por fim, fundamentamos nosso método ao trazer definições de redes neurais, progredindo de arquiteturas mais simples, como MLPs, para mais complexas, como recorrentes e convolucionais.

Fizemos uma vasta revisão bibliográfica para ambas as tarefas. Decidimos agrupar os trabalhos da literatura de SS quanto ao tipo de fala, enfatizando três tipos de fala: (1) fala preparada; (2) fala espontânea; (3) fala comprometida. O último tipo é a fala usada no cenário deste trabalho de pesquisa. Como disfluências ocorrem raramente em fala preparada, não dividimos os trabalhos da literatura de DD quanto ao tipo de fala. Conforme a literatura (LIU, 2004), é possível categorizar as disfluências em preenchimentos e disfluências de edição e, portanto, dividimos os trabalhos relacionados usando essa categorização.

Antes de desenvolver métodos para SS e DD, foi necessário uma ampla análise sobre cada fenômeno. Para tanto, foram utilizados dois *corpus* da área clínica coletados pela FMUSP e PUCRS: *corpus* Cinderela e o *corpus* da bateria BALE. Eles são compostos de narrativas de pacientes de histórias contidas num livro de figuras, com cenas em sequência, sem palavras. Para verificar diferenças entre fala comprometida e fala preparada, coletamos o *corpus* da Constituição Brasileira de 1988 do Projeto FalaBrasil, que é um *corpus* de fala preparada modificado para ser usado em nossos experimentos.

O principal *corpus* desta pesquisa é o *corpus* Cinderela, que é composto por um conjunto de narrativas recontadas e transcritas manualmente de fala comprometida e normal. Ele foi anotado primeiramente com limites de sentenças, levando em consideração apenas informações textuais presentes na transcrição e, então, ajustado manualmente para que as transcrições estivessem fielmente alinhadas palavra-a-palavra com os áudios. Após essa anotação, outros dois anotadores levaram em consideração pistas prosódicas para anotar novamente as transcrições ajustadas. Para o processo de anotação de disfluência foi criado um manual, no qual uma categorização bem definida foi explicada com exemplificações, visando evitar ambiguidades. Dois anotadores anotaram preenchimentos de acordo com seus tipos, e para disfluências de edição apenas um anotador realizou a anotação específica de cada tipo para todas as transcrições. Todas as versões do *corpus* Cinderela com as devidas anotações estão disponíveis a outros pesquisadores, via pedidos por e-mail.

O método desenvolvido para o DeepBonDD para as tarefas de SS e DD foi inspirado no sucesso de trabalhos relacionados recentes (TILK; ALUMÄE, 2016; CHE *et al.*, 2016; HOUGH;

SCHLANGEN, 2017) e consiste em um modelo de Redes Neurais Recorrentes Convolucionais (RCNN) que utiliza dois tipos de *features*: textuais e prosódicas. As *features* textuais foram divididas em etiquetas morfossintáticas e *word embeddings*. Já as *features* prosódicas foram divididas em duração, pitch, energia e pausa. Praticamente a mesma arquitetura da RCNN pode ser utilizada para SS como para DD, com a exceção do número de classes definidas para cada tarefa, o qual corresponde ao número de neurônios na camada de saída. Nosso modelo final consiste na combinação linear de dois modelos independentes. O primeiro é responsável por tratar apenas informações textuais, enquanto o segundo trata apenas informações prosódicas. Em DD, para detecção de disfluências de edição, nos baseamos em trabalhos relacionados recentes (ZAYATS; OSTENDORF; HAJISHIRZI, 2016; HOUGH; SCHLANGEN, 2017) e introduzimos um modelo de CRF na saída da nossa RCNN e também um conjunto específico de *features* linguísticas.

Avaliamos o método proposto para a tarefa de SS com várias abordagens diferentes, e descobrimos que as *embeddings* de palavras induzidas pelo *word2vec* são mais eficientes para a nossa RCNN. Ao adicionar informações prosódicas em conjunto com essas *embeddings* e etiquetas morfossintáticas, verificamos que a combinação entre todas essas informações geralmente resulta em melhores resultados em termos de F_1 . Analisando em termos de *SER*, verificamos que apenas as *embeddings* sozinhas apresentam um resultado melhor do que usando todas as *features*, indicando uma possível linha de investigação quanto a qualidade das informações prosódicas utilizadas.

Justificamos a escolha de um modelo baseado em redes neurais recorrentes convolucionais ao verificar o desempenho de diferentes métodos de classificação baseados em redes neurais, sendo que nossa RCNN obteve melhores resultados em todos os *corpus* testados: $F_1 = 0.77$ para CTL, $F_1 = 0.74$ para CCL, $F_1 = 0.79 - 0.80$ para Constituição. Verificamos também que nosso modelo conseguiu bons resultados quando testamos com narrativas cuja história é diferente da história usada para as narrativas de treinamento. Por fim, com a análise de erros, mostramos que nossa RCNN conseguiu aprender evidências lexicais, sintáticas e semânticas no *corpus* de narrativas transcritas de testes neuropsicológicos base desta pesquisa.

Avaliamos a nossa RCNN para a tarefa de detecção de preenchimentos (pausas preenchidas e marcadores discursivos) com várias abordagens diferentes. Logo nos primeiros experimentos, descobrimos que as pausas preenchidas são fortemente relacionadas com as identidades de palavras. Com a avaliação de *word embeddings* descobrimos que a melhor técnica foi o *fastText* Skip-gram com 600 dimensões. Ao variar o conjunto de *features*, percebemos que informações prosódicas têm um baixo impacto na classificação, e que as *features* textuais de *embeddings* e PoS são, respectivamente, mais impactantes na detecção de ambos os tipos de preenchimentos. Entretanto, os melhores resultados foram obtidos quando todas as *features* foram utilizadas em conjunto.

Para apoiar a escolha de um modelo baseado em redes neurais recorrentes convolucionais,

analisamos o desempenho de diferentes métodos de classificação baseados em redes neurais, sendo que nossa RCNN obteve os melhores resultados na detecção de marcadores discursivos para CTL e CCL, e empatou com o CRF para detecção de pausas preenchidas de CCL. Com a análise de erros, mostramos que nossa RCNN baseia-se fortemente na identidade das palavras para a detecção de ambos preenchimentos. Por fim, mostramos que a melhor escolha para a detecção de preenchimentos é combinar lista de palavras para a detecção de pausas preenchidas e nosso modelo de RCNN para detecção de marcadores discursivos.

Em busca de maior desempenho, verificamos que é possível treinar outro modelo para decidir o caso da pausa preenchida “é”, pois ela é muito ambígua e ocorre frequentemente em fala comprometida. Utilizando essa estratégia e ao revisar a anotação para a pausa preenchida “é”, obtemos em média um F_1 entre 0.91 e 0.92 para detecção de preenchimentos tanto para transcrições de CTL quanto para de CCL.

Para a detecção de disfluências de edição, realizamos duas alterações na RCNN: (i) adicionamos novas *features* linguísticas; (ii) incluímos um modelo CRF de cadeia linear com o algoritmo de Viterbi na última camada da nossa RCNN (formando assim o modelo RCNN-CRF). Com os *baselines*, verificamos que informações lexicais são bastante impactantes para detecção de repetições. Com a avaliação de *word embeddings* descobrimos que a melhor técnica foi o *wang2vec*, com Skip-gram de 600 dimensões. Mas ainda assim, os resultados utilizando apenas *word embeddings* ficaram abaixo dos *baselines* para a detecção de repetições e recomeços. Observamos que os melhores resultados foram obtidos utilizando apenas as novas *features* linguísticas e etiquetas morfossintáticas simultaneamente, ultrapassando o *baseline* por uma margem larga. Entretanto, nossos resultados mostraram que informações prosódicas e de *word embeddings* não contribuem para detecção de repetições e revisões.

Assim como fizemos para as tarefas anteriores, para apoiar a escolha de um modelo baseado em redes neurais recorrentes convolucionais, analisamos o desempenho de diferentes métodos de classificação baseados em redes neurais. Com essa análise ficou evidente que a introdução do modelo de CRF em nosso modelo de RCNN original foi uma boa decisão, já que a RCNN-CRF obteve os melhores resultados a detecção de repetições e revisões para todos os grupos de pacientes. Com os experimentos, verificamos que os resultados oscilaram bastante para cada tipo de disfluência: repetições foram identificadas com $F_1 = 0.85 - 0.89$, já revisões ficaram na faixa de $F_1 = 0.35 - 0.37$, e recomeços ficaram na faixa de $F_1 = 0.05$, muito mais abaixo que os outros.

Analisamos os principais erros e acertos da nossa RCNN-CRF para cada tipo de disfluência de edição e com isso descobrimos que repetições são usualmente identificáveis ao analisar sequências de palavras duplicadas, principalmente para repetições pequenas, e que os principais erros são devidos a prefixos ou enganos com revisões.

Para revisões verificamos que os principais acertos foram para tamanhos maiores que 1, e que geralmente esses acertos possuem uma estrutura lexical na qual a primeira palavra

do enunciado original é igual a primeira palavra da correção, e as últimas geralmente refletem uma mudança de verbo ou substantivo; já os principais erros foram devido à ambiguidade do fenômeno ou porque as dicas lexicais estavam muito distantes, indicando um possível aumento no tamanho do filtro de convolução da RCNN para 9 ou 11 (usamos 7 em nossos experimentos) e um aumento do número de palavras vizinhas que devem ser consideradas nas *features* manuais.

Os recomeços foram claramente a categoria na qual a nossa RCNN-CRF obteve os piores resultados, sendo que nenhum deles passou do limiar de 5% de F_1 .

Avaliamos também se a segmentação e remoção de disfluências automáticas do DeepBonDD influenciaram positivamente ou negativamente o cálculo das métricas sintáticas do Coh-Metrix-Dementia. Para isso, selecionamos 9 métricas sintáticas para verificar se houve diferença significativa entre as transcrições manuais e as geradas automaticamente pelo DeepBonDD. As métricas extraídas dependem do sucesso de analisar a árvore sintática como um todo; e também dependem da identificação correta de sintagmas verbais e sintagmas nominais.

Com um nível de significância de 5% ($p < 0.05$), mostramos que os resultados obtidos para transcrições de CTLs não possuem diferença significativa em relação as transcrições manuais. E, como esperado, a degradação do desempenho do método é evidente conforme movemos de uma fala mais comportada (CTL) para uma fala mais comprometida (CCL e DA), sendo que para CCLs três métricas apresentaram diferença significativa, e para DAs cinco. Entretanto, os resultados para CCL ainda podem ser considerados bons, já que apenas três dentre as nove métricas analisadas apresentaram diferença significativa. Por fim, analisando especificamente os casos de divergência, foi possível observar que, apesar das variações de limites de sentença e de disfluências, o DeepBonDD é capaz de gerar transcrições que podem ser usadas por ferramentas de PLN.

Desse modo, podemos resumir as principais contribuições deste trabalho como:

- Criação de um segmentador de sentenças para transcrições de fala comprometida;
- Criação de um removedor de preenchimentos (pausas preenchidas e marcadores discursivos) para transcrições de fala comprometida;
- Criação de um removedor de disfluências de edição (repetições e revisões) para transcrições de fala comprometida;
- Revisão extensa da literatura para as tarefas de SS e DD;
- Elaboração de manual de anotação para a tarefa de DD, assim como o corpus anotado;
- Realização de análises estatísticas de preenchimentos e disfluências de edição em relação ao tamanho das sentenças e ao trecho de ocorrência na transcrição para pacientes idosos.

- Definição de um modelo neural recorrente convolucional para SS e detecção de preenchimentos, e de um modelo que usa o modelo neural recorrente convolucional em conjunto com um modelo de CRF para detecção de disfluências de edição;

Os passos de processamento da abordagem DeepBonDD para segmentar uma transcrição em sentenças e remover as disfluências consistem em: passar a transcrição paralelamente para um segmentador de sentenças e um removedor de pausas preenchidas e marcadores discursivos; combinar a saída desses dois processos; aplicar o removedor de disfluências de repetições e revisões, gerando uma transcrição final segmentada e livre de disfluências. Na ferramenta desenvolvida, é possível utilizar cada um desses passos separadamente. Ou seja, caso desejado, é possível apenas segmentar um documento em sentenças ou apenas remover disfluências desse documento. Além disso, embora o DeepBonDD tenha sido proposto para lidar com transcrições de fala comprometida, ele foi desenvolvido de uma maneira que é possível estendê-lo para lidar com outros tipos de textos, como textos escritos. Para isso basta prover um corpus anotado e treinar o modelo de RCNN novamente com esse tipo de corpus. O DeepBonDD está livremente disponível e documentado em <<https://github.com/mtreviso/deepbondd>>.

7.2 Limitações

O projeto original no qual este trabalho de mestrado está incluído visa a utilização de um sistema de RAV, o qual não estava disponível durante os experimentos deste trabalho. Ao lidar com transcrições automáticas, esperamos uma degradação em nossos resultados, já que erros prévios do sistema de RAV serão passados adiante para o DeepBonDD. Esse comportamento já foi demonstrado em vários trabalhos de SS e DD (BATISTA *et al.*, 2012; CHO *et al.*, 2015; TILK; ALUMÄE, 2016; HOUGH; SCHLANGEN, 2017).

O tamanho do nosso corpus base (Cinderela v2) está muito abaixo do tamanho dos corpus utilizados para língua inglesa, e o fato de métodos baseados em redes neurais profundas terem obtido bons desempenhos foi surpreendente. Para uma simples comparação, nosso corpus possui apenas ~24K palavras obtidas a partir de ~4h de recontos, enquanto o Switchboard-2 possui ~3M palavras de transcrições manuais e automáticas provindas de ~303h de conversações telefônicas.

Além disso, a anotação do nosso corpus em relação a disfluências de edição foi feita com apenas dois anotadores para preenchimentos, e com apenas um anotador para as disfluências de edição. Portanto, não foi realizada a análise de concordância entre anotadores para criar o *corpus gold standard*. Este trabalho é, assim, um primeiro estágio na construção de corpus anotados para as tarefas de SS e DD a partir de transcrições vindas de fala espontânea e comprometida.

Analisar trabalhos relacionados nessa área é difícil, principalmente pela grande variedade de corpus e metodologias de avaliação utilizadas. Além disso, no melhor de nosso conhecimento, não há trabalhos para SS e DD baseados em corpus de fala para a língua portuguesa brasileira.

Entre os trabalhos relacionados apresentados para outras línguas, há principalmente uma baixa quantidade de trabalhos que tratam de transcrições de fala comprometida.

Em relação ao método proposto, a nossa segmentação de sentenças busca apenas encontrar os limites das sentenças, não subcategorizando os sinais de pontuação. Para a detecção de preenchimentos o nosso método não inclui termos de edição explícitos, pois eles ocorreram com baixa frequência em nosso corpus, para os quais decidimos não realizar experimentos. Já para a detecção de disfluências de edição a nossa RCNN-CRF mostrou-se ineficaz para a detecção de recomenços e, portanto, removemos a detecção desse tipo de disfluência do método final. Além disso, nosso método não foi modelado para detectar disfluências complexas, as quais possuem uma estrutura recursiva de disfluências, entretanto, é possível reaplicar o passo de detecção de repetições e revisões para eliminá-las.

7.3 Trabalhos Futuros

Os principais esforços após este projeto de mestrado devem ser na coleta de mais dados de fala espontânea e comprometida. Dados desse tipo de fala possuem características diferenciadas importantes, como sentenças fragmentadas e um grande número de disfluências. O processamento correto desses fenômenos são essenciais para que futuras ferramentas de PLN possam ser utilizadas de maneira eficaz.

Trabalhos futuros para SS incluem outras investigações de análise de erros, entre elas analisar o espaço vetorial das *embeddings* após o processo de treinamento para verificar evidências semânticas, e também avaliar a perplexidade de uma modelo de língua simples treinado nas predições do nosso modelo para verificar evidências sintáticas. Finalmente, com um maior número de dados é possível anotar diferentes tipos de sinais de pontuação, para o que bastaria aumentar o número de neurônios na última camada da rede neural para lidar com essa estratégia.

Com a avaliação das *features* abrimos uma linha de investigação interessante para detecção de disfluências em relação a meios de representar eficientemente a identidade das palavras usando *word embeddings*. Para a detecção de preenchimentos sugerimos uma possível estratégia em separar a detecção da pausa preenchida “é” de outras pausas preenchidas, para a qual já mostramos que a RCNN produz bons resultados, porém a utilização de recursos extras como modelos de língua treinados sobre textos bem escritos podem ajudar. Para a detecção de disfluências de edição, acreditamos que mais trabalhos devem ser feitos em relação às *features* linguísticas, como a inclusão de *features* que indicam o gênero e o grau das palavras, as quais podem beneficiar a identificação repetições e revisões pequenas - que representam os erros mais frequentes cometidos pela RCNN-CRF.

As *features* prosódicas mostraram-se benéficas para as tarefas de SS e detecção de preenchimentos para transcrições de CTL e CCL. Para transcrições de DA não foi possível obter informações prosódicas, pois infelizmente não conseguimos realizar o alinhamento forçado

entre as transcrições e o áudio. No entanto, para todas as tarefas observamos que o uso *features* prosódicas acrescentaram muito pouco ao desempenho da RCNN, ou até mesmo pioraram os resultados como visto para detecção de disfluências de edição. Isso abre uma linha de investigação quanto as informações prosódicas utilizadas, levando em consideração as peculiaridades em relação à qualidade dos áudios extraídos, ou até mesmo em relação à insuficiência discriminativa das *features* extraídas para a natureza inerente dos pacientes.

Por fim, acreditamos que um nicho ainda maior deve ser preenchido para trabalhar com fala espontânea do português brasileiro. Por exemplo, há poucas ferramentas de processamento de fala disponíveis, como sistemas de RAV, sistemas para alinhamento forçado entre transcrições manuais e o respectivo áudio, e até mesmo modelos acústicos e modelos de língua. Para isso, trabalhos futuros podem se beneficiar do projeto NURC¹, no qual o corpus levantado no país perfaz hoje um total de mais de 1500 horas de registros magnetofônicos - se consideramos todas as cidades do projeto - coletados na década de 70. Esse material representa o desempenho linguístico de participantes em aulas e conferências, diálogos informais, e entrevistas. E, portanto, caracteriza uma grande quantidade de dados bem anotados para fala espontânea.

Artigos Publicados e Submetidos

Artigos longos publicados:

- Treviso, M. V., Shulby, C., and Aluísio, S. M. (2017). Sentence Segmentation in Narrative Transcripts from Neuropsychological Tests using Recurrent Convolutional Neural Networks. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL).
- Treviso, M. V., Shulby, C., and Aluísio, S. M. (2017). Evaluating Word Embeddings for Sentence Boundary Detection in Speech Transcripts. Proceedings of Symposium in Information and Human Language Technology (STIL). * Segundo melhor artigo da conferência.

Artigos longos submetidos:

- Treviso, M. V.; dos Santos, L. B.; Shulby, C.; Hübner, L. C.; Mansur, L. L.; Aluísio, S. M. (2017). Detecção de comprometimento cognitivo leve em narrativas em Português Brasileiro: primeiros passos para um sistema automatizado. Revista Letras de Hoje.

¹ <<http://www.letras.ufrj.br/nurc-rj/>>

Artigos influenciados por este trabalho:

- Hübner, L. C.; Loureiro F.; Smidarle A. D.; Pedro J. R.; Monticelli V.; Treviso M. V.; dos Santos L. B.; Schilling L. P.; Mansur L. L.; Aluísio S. M. (2017). Automatically distinguishing Mild Cognitive Impairment, Alzheimer’s Disease and education effect in healthy aging in narratives in Brazilian Portuguese. Science of Aphasia Conference.
- Hartmann, N. S.; Fonseca E. R.; Shulby, C.; Treviso, M. V.; Rodrigues, J. S.; Aluísio, S. M. (2017). Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. Symposium in Information and Human Language Technology (STIL).
- Corrêa Jr, E. A.; Marinho V. Q.; dos Santos, L. B.; Bertaglia, T. F. C.; Treviso, M. V.; Brum H. B. (2017). PELESent: Cross-domain polarity classification using distant supervision. Brazilian Conference on Intelligent Systems (BRACIS).

REFERÊNCIAS

ALABISO, J.; MACINTYRE, R.; GRAFF, D. English broadcast news transcripts (hub4-1997). In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC98T28**. [S.l.], 1998. Citado na página 29.

ALMEIDA, V. B. de. Pausas preenchidas e domínios prosódicos: evidências para a validação do descritor fluência em um teste de proficiência oral em língua estrangeira. **ALFA: Revista de Linguística**, 2009. Citado na página 21.

ALUÍSIO, S.; CUNHA, A.; SCARTON, C. Evaluating progression of alzheimer's disease by regression and classification methods in a narrative language test in portuguese. In: **PROPOR**. [S.l.: s.n.], 2016. p. 109–114. Citado nas páginas 3 e 89.

ANCHIÊTA, R. T.; NETO, F. A. R.; SOUSA, R. F. de; MOURA, R. S. Using stylometric features for sentiment classification. In: **CICLing**. [S.l.: s.n.], 2015. p. 189–200. Citado na página 12.

ANDREETTA, S.; CANTAGALLO, A.; MARINI, A. Narrative discourse in anomie aphasia. **Neuropsychologia**, p. 1787–1793, 2012. Citado na página 2.

AVANÇO, L. V.; BRUM, H. B.; NUNES, M. d. G. V. Improving opinion classifiers by combining different methods and resources. In: **ENIAC**. [S.l.: s.n.], 2016. Citado na página 64.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **ICLR**, 2015. Citado na página 66.

BALDWIN, T.; JOSEPH, M. P. A. K. Restoring punctuation and casing in english text. **Australasian Joint Conference on Artificial Intelligence**, p. 547–556, 2009. ISSN 03029743. Citado nas páginas 13 e 53.

BARROS, D. L. P. de. Procedimentos de reformulação: a correção. In: PRETI, D. (Ed.). **Análise de Textos Oraís**. fourth. [S.l.]: Humanitas, 1999, (Coleção Projetos Paralelos, v. 1). p. 129–156. Citado nas páginas 147 e 148.

BATISTA, F.; CASEIRO, D.; MAMEDE, N.; TRANCOSO, I. Recovering capitalization and punctuation marks for automatic speech recognition: Case study for portuguese broadcast news. **Speech Communication**, p. 847–862, 2008. Citado na página 58.

BATISTA, F.; CASEIRO, D.; MAMEDE, N. J.; TRANCOSO, I. Recovering punctuation marks for automatic speech recognition. **INTERSPEECH**, p. 2153–2156, 2007. Citado nas páginas 35 e 58.

BATISTA, F.; MONIZ, H.; TRANCOSO, I.; MEINEDO, H.; MATA, A.; MAMEDE, N. Extending the punctuation module for european portuguese. **INTERSPEECH**, p. 1509–1512, 2010. Citado na página 58.

- BATISTA, F.; MONIZ, H.; TRANCOSO, I.; MAMEDE, N. Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. **IEEE Transactions on Audio, Speech, and Language Processing**, p. 474–485, 2012. ISSN 1558-7916. Citado nas páginas 5, 6, 15, 16, 58, 61, 71, 93, 121 e 168.
- BATISTA, F.; TRANCOSO, I.; MAMEDE, N. J. Comparing automatic rich transcription for portuguese, spanish and english broadcast news. In: **ASRU Workshop**. [S.l.: s.n.], 2009. p. 540–545. Citado na página 58.
- BATISTA, F. M. M.; MAMEDE, D. N. J. N. **Recovering Capitalization and Punctuation Marks on Speech Transcriptions**. 171 p. Tese (Doutorado) — Instituto Superior Técnico - Universidade de Lisboa, 2011. Citado nas páginas 4, 16, 32, 53, 55, 58 e 128.
- BATISTA, P. d. S. *et al.* **Avanços em reconhecimento de fala para português brasileiro e aplicações: ditado no libreoffice e unidade de resposta audível com asterisk**. Dissertação (Mestrado) — Universidade Federal do Pará, 2013. Citado nas páginas 27 e 101.
- BAYLES, K.; TOMOEDA, C. Abcd: Arizona battery for communication disorders of dementia. **Tucson, AZ: Canyonlands Publishing**, 1993. Citado nas páginas 2 e 7.
- BEAR, J.; DOWDING, J.; SHRIBERG, E. Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. In: **ACL**. [S.l.: s.n.], 1992. p. 56–63. Citado nas páginas 5, 6, 20, 77, 84 e 85.
- BEEFERMAN, D.; BERGER, A.; LAFFERTY, J. Cyberpunch: A lightweight punctuation annotation system for speech. In: **ICASSP**. [S.l.: s.n.], 1998. p. 689–692. ISBN 0780344286. Citado nas páginas 16, 55, 56 e 71.
- BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **JMLR**, p. 1137–1155, 2003. Citado na página 36.
- BERTAGLIA, T. F.; NUNES, M. G. Exploring word embeddings for unsupervised textual user-generated content normalization. In: **COLING. Workshop WNUT**. [S.l.], 2016. p. 74–82. Citado na página 13.
- BERTERO, D.; WANG, L.; CHAN, H. Y.; FUNG, P. A comparison between a dnn and a crf disfluency detection and reconstruction system. In: **INTERSPEECH**. [S.l.: s.n.], 2015. p. 844–848. Citado nas páginas 83, 135, 139, 140, 141, 148 e 156.
- BOERSMA, P. P. G. *et al.* Praat, a system for doing phonetics by computer. **Glott international**, 2002. Citado na página 110.
- BOEUF, C. L. **Raconte: 55 historiettes en images**. [S.l.]: L'École, 1971. Citado na página 100.
- BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **arXiv preprint arXiv:1607.04606**, 2016. Citado nas páginas 38 e 39.
- BREIMAN, L. Bagging predictors. **JMLR**, p. 123–140, 1996. Citado na página 62.
- BROWN, P. F.; DESOUZA, P. V.; MERCER, R. L.; PIETRA, V. J. D.; LAI, J. C. Class-based n-gram models of natural language. **Computational linguistics**, p. 467–479, 1992. Citado na página 36.

- BRUCKSCHEN, M.; MUNIZ, F.; SOUZA, J.; FUCHS, J.; INFANTE, K.; MUNIZ, M.; GONÇALVES, P.; VIEIRA, R.; ALUISIO, S. Anotação lingüística em xml do corpus pln-br. **Série de relatórios do NILC, ICMC-USP**, 2008. Citado nas páginas 116, 132 e 143.
- BRUM, H.; ARAUJO, F.; KEPLER, F. Sentiment analysis for brazilian portuguese over a skewed class corpora. In: **PROPOR**. [S.l.: s.n.], 2016. p. 134–138. Citado na página 12.
- CALHOUN, S.; CARLETTA, J.; BRENIER, J. M.; MAYO, N.; JURAFSKY, D.; STEEDMAN, M.; BEAVER, D. The nlt-format switchboard corpus: A rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. **LREC**, p. 387–419, 2010. Citado na página 28.
- CANAVAN, A.; GRAFF, D.; ZIPPERLEN, G. Callhome american english speech. In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC97S42**. [S.l.], 1997. Citado na página 30.
- CARLETTA, J. Assessing agreement on classification tasks: the kappa statistic. **Computational linguistics**, p. 249–254, 1996. Citado na página 93.
- CARLETTA, J.; EVERT, S.; HEID, U.; KILGOUR, J. The nite xml toolkit: data model and query language. **LREC**, p. 313–334, 2005. Citado na página 28.
- CAUCHY, A. Méthode générale pour la résolution des systemes d'équations simultanées. **Comp. Rend. Sci. Paris**, p. 536–538, 1847. Citado na página 44.
- CHE, X.; WANG, C.; YANG, H.; MEINEL, C. Punctuation prediction for unsegmented transcript based on word vector. **LREC**, p. 654–658, 2016. Citado nas páginas 9, 6, 16, 34, 35, 59, 66, 71, 72, 105, 122, 123, 164 e 165.
- CHEN, J. C. Speech recognition with automatic punctuation. **EUROSPEECH**, p. 6–9, 1999. Citado nas páginas 6, 18, 55, 56, 57 e 71.
- CHEN, S. F.; KINGSBURY, B.; MANGU, L.; POVEY, D.; SAON, G.; SOLTAU, H.; ZWEIG, G. Advances in speech transcription at ibm under the darpa ears program. **IEEE Transactions on Audio, Speech, and Language Processing**, p. 1596–1608, 2006. Citado na página 55.
- CHENG, J.; LAPATA, M. Neural summarization by extracting sentences and words. In: **ACL**. [S.l.: s.n.], 2016. Citado na página 42.
- CHO, E.; KILGOUR, K.; NIEHUES, J.; WAIBEL, A. Combination of nn and crf models for joint detection of punctuation and disfluencies. p. 3650–3654, 2015. ISSN 19909772. Citado nas páginas 9, 35, 66, 78, 82, 84 e 168.
- CHO, K.; MERRIENBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: **EMNLP**. [S.l.]: Association for Computational Linguistics, 2014. p. 1724–1734. Citado nas páginas 49 e 81.
- CHRISTENSEN, H.; GOTOH, Y.; RENALS, S. Punctuation annotation using statistical prosody models. **ISCA Tutorial and Research**, 2006. Citado nas páginas 6, 16, 56, 61, 65 e 71.
- CIERI, C.; GRAFF, D.; KIMBALL, O.; MILLER, D.; WALKER, K. Fisher english training speech part 1 speech. In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC2004S13**. [S.l.], 2004. Citado nas páginas 32 e 83.

- CLARK, H. H.; WASOW, T. Repeating words in spontaneous speech. **Cognitive psychology**, v. 37, p. 201–242, 1998. Citado nas páginas 20 e 84.
- CLEMENTE, R.; RIBEIRO-FILHO, S. Comprometimento cognitivo leve: aspectos conceituais, abordagem clínica e diagnóstica. **Revista Hospital Universitário Pedro Ernesto**, 2008. Citado na página 1.
- COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: **ICML**. [S.l.: s.n.], 2008. p. 160–167. Citado na página 37.
- COLLOBERT, R.; WESTON, J.; BOTTOU, L.; KARLEN, M.; KAVUKCUOGLU, K.; KUKSA, P. Natural language processing (almost) from scratch. **JMLR**, p. 2493–2537, 2011. Citado nas páginas 37, 49, 59 e 83.
- CORE, M. G.; SCHUBERT, L. K. A syntactic framework for speech repairs and other disruptions. In: **ACL**. [S.l.: s.n.], 1999. p. 413–420. Citado na página 79.
- COTES, C. O uso das pausas nos diferentes estilos de televisão. **Rev. CEFAC**, p. 228–237, 2007. Citado na página 40.
- CUENDET, S.; HAKKANI-TÜR, D.; SHRIBERG, E.; FUNG, J.; FAVRE, B. Cross-genre feature comparisons for spoken sentence segmentation. **ICSC**, p. 265–271, 2007. ISSN 1793-351X. Citado na página 125.
- CUNHA, A. L. da. **Coh-Metrix-Dementia: Análise Automática de Distúrbios de Linguagem nas Demências utilizando Processamento de Línguas Naturais**. Master's thesis. Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. 2015. Citado na página 158.
- CUNHA, A. L. V. da; SOUSA, L. B. de; MANSUR, L. L.; ALUÍSIO, S. M. Automatic proposition extraction from dependency trees: Helping early prediction of alzheimer's disease from narratives. In: IEEE. **Computer-Based Medical Systems (CBMS), 2015 IEEE 28th International Symposium on**. [S.l.], 2015. p. 127–130. Citado na página 90.
- DESHMUKH, N.; GANAPATHIRAJU, A.; GLEESON, A.; HAMAKER, J.; PICONE, J. Resegmentation of switchboard. In: **ICSLP**. [S.l.: s.n.], 1998. Citado na página 28.
- DEVLIN, J.; ZBIB, R.; HUANG, Z.; LAMAR, T.; SCHWARTZ, R. M.; MAKHOUL, J. Fast and robust neural network joint models for statistical machine translation. In: CITESEER. **ACL**. [S.l.], 2014. p. 1370–1380. Citado na página 42.
- DÜBOIS, J. *et al.* **Dicionário de linguística**. [S.l.]: São Paulo: Cultrix, 1973. Citado nas páginas 40 e 41.
- DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; DEERWESTER, S.; HARSHMAN, R. Using latent semantic analysis to improve access to textual information. In: ACM. **SIGCHI**. [S.l.], 1988. p. 281–285. Citado na página 36.
- FAVRE, B.; HAKKANI-TÜR, D.; PETROV, S.; KLEIN, D. Efficient sentence segmentation using syntactic features. **Spoken Language Technology Workshop**, 2008. Citado nas páginas 56, 57, 61 e 71.

- FERGUSON, J.; DURRETT, G.; KLEIN, D. Disfluency detection with a semi-markov model and prosodic features. **NAACL**, p. 257–262, 2015. Citado nas páginas 5, 6, 39, 80, 84, 85, 140, 146 e 149.
- FONSECA, E. R.; ALUÍSIO, S. M. Improving pos tagging across portuguese variants with word embeddings. In: **PROPOR**. [S.l.: s.n.], 2016. p. 227–232. Citado na página 12.
- FONSECA, E. R.; ALUISIO, S. M. *et al.* A deep architecture for non-projective dependency parsing. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS-ACL. **NAACL; Workshop on Vector Space Modeling**. [S.l.], 2015b. Citado nas páginas 42 e 49.
- FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. **Journal of the Brazilian Computer Society**, Springer London, p. 13, 2015a. Citado nas páginas 69 e 110.
- FORMAN, G.; SCHOLZ, M. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. **ACM SIGKDD Explorations Newsletter**, p. 49–57, 2010. Citado na página 19.
- FRASER, K. C.; BEN-DAVID, N.; HIRST, G.; GRAHAM, N. L.; ROCHON, E. Sentence segmentation of aphasic speech. **NAACL**, p. 862–871, 2015. Citado nas páginas 11, 16, 67, 68, 70, 71, 75, 120, 122 e 123.
- FRASER, K. C.; MELTZER, J. A.; RUDZICZ, F. Linguistic features identify alzheimer’s disease in narrative speech. **Journal of Alzheimer’s Disease**, p. 407–422, 2016. Citado na página 3.
- FRAZIER, L. Syntactic complexity. **Natural language parsing: Psychological, computational, and theoretical perspectives**, 1985. Citado nas páginas 158 e 159.
- GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: **Aistats**. [S.l.: s.n.], 2010. p. 249–256. Citado na página 106.
- GODFREY, J.; HOLLIMAN, E. Switchboard-1 release 2. In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC97S62**. [S.l.], 1993. Citado na página 28.
- GODFREY, J. J.; HOLLIMAN, E. C.; MCDANIEL, J. Switchboard: Telephone speech corpus for research and development. In: IEEE. **ICASSP**. [S.l.], 1992. p. 517–520. Citado nas páginas 20, 28, 55 e 75.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. Book in preparation for MIT Press. 2016. Disponível em: <<http://www.deeplearningbook.org>>. Citado nas páginas 37, 42, 47 e 51.
- GOTOH, Y.; RENALS, S. Sentence boundary detection in broadcast speech transcripts. **ISCA Workshop**, p. 228–235, 2000. Citado nas páginas 6, 15, 35, 56, 61 e 71.
- GRAFF, D.; ALABISO, J. English broadcast news transcripts (hub4-1996). In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC97T22**. [S.l.], 1997. Citado na página 29.
- GRAVANO, A.; JANSCHKE, M.; BACCHIANI, M. Restoring punctuation and capitalization in transcribed speech. **ICASSP**, p. 4741–4744, 2009. ISSN 15206149. Citado na página 56.

- GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. **ICASSP**. [S.l.], 2013. p. 6645–6649. Citado nas páginas 42 e 47.
- GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. Lstm: A search space odyssey. **IEEE transactions on neural networks and learning systems**, 2016. Citado na página 47.
- GROUP, N. M. I. 1997 hub5 english evaluation. In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC2002S23**. [S.l.], 2002. Citado na página 30.
- GÜZ, Ü.; CUENDET, S.; HAKKANI-TÜR, D. Z.; TÜR, G. Co-training using prosodic and lexical information for sentence segmentation. p. 2597–2600, 2007. Citado nas páginas 63, 73 e 121.
- HARKINS, D. **Switchboard resegmentation project**. 2003. Disponível em: <<http://www.cavs.msstate.edu/hse/ies/projects/switchboard>>. Citado na página 28.
- HASSAN, H.; SCHWARTZ, L.; HAKKANI-TÜR, D.; TUR, G. Segmentation and disfluency removal for conversational speech translation. In: **INTERSPEECH**. [S.l.: s.n.], 2014. p. 318–322. Citado nas páginas 5 e 78.
- HEEMAN, P.; ALLEN, J. Detecting and correcting speech repairs. In: **ACL**. [S.l.: s.n.], 1994. p. 1–8. Citado nas páginas 6, 77, 79, 84 e 85.
- HINDLE, D. Deterministic parsing of syntactic non-fluencies. In: **ACL**. [S.l.: s.n.], 1983. p. 123–128. Citado na página 39.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, p. 1735–1780, 1997. Citado nas páginas 47 e 64.
- HOUGH, J.; SCHLANGEN, D. Recurrent neural networks for incremental disfluency detection. In: **INTERSPEECH**. [S.l.: s.n.], 2015. p. 849–853. Citado nas páginas 7, 76, 81, 83, 84, 85, 135, 141, 147 e 148.
- _____. Joint, incremental disfluency detection and utterance segmentation from speech. **EACL**, p. 326–336, 2017. Citado nas páginas 7, 81, 83, 84, 85, 86, 135, 136, 139, 141, 142, 146, 147, 148, 153, 155, 156, 164, 165 e 168.
- HUANG, G.; XU, C.; XIAO, X.; XIE, L.; CHNG, E. S.; LI, H. Multi-view features in a dnn-crf model for improved sentence unit detection on english broadcast news. In: **APSIPA**. [S.l.: s.n.], 2014. ISBN 9786163618238. Citado nas páginas 6, 16, 39, 58, 64, 71, 73 e 121.
- HUANG, J.; ZWEIG, G. Maximum entropy model for punctuation annotation from speech. **INTERSPEECH**, p. 917–920, 2002. Citado nas páginas 54 e 61.
- JARROLD, W. L.; PEINTNER, B.; YEH, E.; KRASNOW, R.; JAVITZ, H. S.; SWAN, G. E. Language analytics for assessing brain health: Cognitive impairment, depression and pre-symptomatic alzheimer’s disease. In: **International Conference on Brain Informatics**. [S.l.: s.n.], 2010. p. 299–307. Citado na página 2.
- JERÔNIMO, G. M. *et al.* **Produção de narrativas orais no envelhecimento sadio, no comprometimento cognitivo leve e na doença de Alzheimer e sua relação com construtos cognitivos e escolaridade**. Tese (Doutorado), 2016. Citado na página 100.

- JOHNSON, M. Pcfg models of linguistic tree representations. **Computational Linguistics**, p. 613–632, 1998. Citado na página 57.
- JOHNSON, M.; CHARNIAK, E. A tag-based noisy channel model of speech repairs. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **ACL**. [S.l.], 2004. p. 33. Citado na página 80.
- JONES, D. A.; WOLF, F.; GIBSON, E.; WILLIAMS, E.; FEDORENKO, E.; REYNOLDS, D. A.; ZISSMAN, M. A. Measuring the readability of automatic speech-to-text transcripts. In: **INTERSPEECH**. [S.l.: s.n.], 2003. Citado na página 4.
- JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. **arXiv preprint arXiv:1607.01759**, 2016. Citado na página 38.
- JR, C. N. S.; KAESTNER, C. A. An analysis of sentence boundary detection systems for english and portuguese documents. In: **CICLing**. [S.l.: s.n.], 2004. p. 135–141. Citado na página 53.
- JURAFSKY, D.; MARTIN, J. H. **Speech and language processing**. [S.l.]: Pearson, 2014. v. 3. Citado nas páginas 3, 12, 14, 21, 27 e 40.
- KEPLER, F. N.; FINGER, M. Variable-length markov models and ambiguous words in portuguese. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **NAACL**. [S.l.], 2010. p. 15–23. Citado nas páginas 12 e 127.
- KHOMITSEVICH, O.; CHISTIKOV, P.; KRIVOSHEEVA, T.; EPIMAKHOVA, N.; CHERNYKH, I. Combining prosodic and lexical classifiers for two-pass punctuation detection in a russian asr system. p. 161–169, 2015. Citado nas páginas 64, 71, 72, 73 e 119.
- KIM, J.-H.; WOODLAND, P. C. The use of prosody in a combined system for punctuation generation and speech recognition. **INTERSPEECH**, p. 2757–2760, 2001. ISSN 1098-6596. Citado nas páginas 57 e 61.
- _____. A combined punctuation generation and speech recognition system and its performance enhancement using prosody. **Speech Communication**, p. 563–577, 2003. ISSN 01676393. Citado nas páginas 6, 16, 29, 57, 65, 71 e 72.
- KIM, Y. Convolutional neural networks for sentence classification. **EMNLP**, 2014. Citado na página 107.
- KOLÁR, J.; LIU, Y.; SHRIBERG, E. Genre effects on automatic sentence segmentation of speech: A comparison of broadcast news and broadcast conversations. In: **ICASSP**. [S.l.: s.n.], 2009. p. 4701—4704. Citado nas páginas 4, 53, 65, 74 e 120.
- LAI, S.; XU, L.; LIU, K.; ZHAO, J. Recurrent convolutional neural networks for text classification. In: **AAAI**. [S.l.: s.n.], 2015. p. 2267–2273. Citado nas páginas 36 e 105.
- LANDAUER, T. K.; FOLTZ, P. W.; LAHAM, D. An introduction to latent semantic analysis. **Discourse processes**, Taylor & Francis, p. 259–284, 1998. Citado na página 36.
- LANG, K. J.; WAIBEL, A. H.; HINTON, G. E. A time-delay neural network architecture for isolated word recognition. **Neural networks**, p. 23–43, 1990. Citado na página 45.
- LECUN, Y.; BENGIO, Y. *et al.* Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, p. 10, 1995. Citado na página 49.

- LEHR, M.; PRUD'HOMMEAU, E. T.; SHAFRAN, I.; ROARK, B. Fully automated neuropsychological assessment for detecting mild cognitive impairment. In: **INTERSPEECH**. [S.l.: s.n.], 2012. p. 1039–1042. Citado na página 3.
- LENNON, P. The lexical element in spoken second language fluency. In: **Perspectives on fluency**. [S.l.: s.n.], 2000. p. 25–42. Citado na página 20.
- LEVELT, W. Speaking: from intention to articulation mit press. **Cambridge, MA**, 1989. Citado na página 23.
- LING, W.; DYER, C.; BLACK, A.; TRANCOSO, I. Two/too simple adaptations of word2vec for syntax problems. In: **NAACL**. [S.l.: s.n.], 2015. Citado nas páginas 36 e 38.
- LIU, Y. **STRUCTURAL EVENT DETECTION FOR RICH TRANSCRIPTION OF SPEECH**. Tese (Doutorado) — Purdue University, 2004. Citado nas páginas 6, 11, 12, 15, 20, 21, 22, 23, 24, 61, 76, 86, 98, 164 e 195.
- LIU, Y.; CHAWLA, N. V.; HARPER, M. P.; SHRIBERG, E.; STOLCKE, A. A study in machine learning from imbalanced data for sentence boundary detection in speech. **Computer Speech and Language**, 2005b. ISSN 08852308. Citado nas páginas 53, 62, 71, 73 e 74.
- LIU, Y.; SHRIBERG, E. Comparing evaluation metrics for sentence boundary detection. In: **ICASSP**. [S.l.: s.n.], 2007. Citado nas páginas 18 e 19.
- LIU, Y.; SHRIBERG, E.; STOLCKE, A.; HARPER, M. P. Comparing hmm, maximum entropy, and conditional random fields for disfluency detection. In: **INTERSPEECH**. [S.l.: s.n.], 2005. p. 3313–3316. Citado nas páginas 6, 79, 82, 83, 84, 85, 95, 135, 140, 148 e 156.
- LIU, Y.; SHRIBERG, E.; STOLCKE, A.; HILLARD, D.; OSTENDORF, M.; HARPER, M. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. **IEEE Transactions on audio, speech, and language processing**, p. 1526–1540, 2006. Citado nas páginas 16, 40, 53, 58, 59, 64, 78 e 121.
- LIU, Y.; SHRIBERG, E.; STOLCKE, A. Automatic disfluency identification in conversational speech using multiple knowledge sources. In: **INTERSPEECH**. [S.l.: s.n.], 2003. p. 957–960. Citado na página 79.
- LIU, Y.; STOLCKE, A.; SHRIBERG, E.; HARPER, M. Using conditional random fields for sentence boundary detection in speech. **ACL**, p. 451–458, 2005a. Citado nas páginas 9, 6, 18, 53, 61, 62, 68, 71, 72, 73, 74 e 128.
- LÓPEZ, R.; PARDO, T. A. S. Experiments on sentence boundary detection in user-generated web content. In: **CICLing**. [S.l.: s.n.], 2015. p. 227–237. ISBN 9783319181103. ISSN 16113349. Citado nas páginas 12, 13 e 53.
- LUENGO, I.; NAVAS, E.; HERNÁEZ, I.; SÁNCHEZ, J. Automatic emotion recognition using prosodic parameters. In: **INTERSPEECH**. [S.l.: s.n.], 2005. p. 493–496. Citado na página 64.
- LUND, K.; BURGESS, C. Producing high-dimensional semantic spaces from lexical co-occurrence. **Behavior Research Methods, Instruments, & Computers**, p. 203–208, 1996. Citado na página 37.
- MA, X.; HOVY, E. H. End-to-end sequence labeling via bi-directional lstm-cnns-crf. **CoRR**, 2016. Citado na página 141.

- MAAS, A. L.; DALY, R. E.; PHAM, P. T.; HUANG, D.; NG, A. Y.; POTTS, C. Learning word vectors for sentiment analysis. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **ACL: Human Language Technologies**. [S.l.], 2011. p. 142–150. Citado na página [59](#).
- MAKHOUL, J.; KUBALA, F.; SCHWARTZ, R.; WEISCHEDEL, R. *et al.* Performance measures for information extraction. In: **Proceedings of DARPA broadcast news workshop**. [S.l.: s.n.], 1999. p. 249–252. Citado na página [19](#).
- MAPSTONE, M.; CHEEMA, A. K.; FIANDACA, M. S.; ZHONG, X.; MHYRE, T. R.; MACARTHUR, L. H.; HALL, W. J.; FISHER, S. G.; PETERSON, D. R.; HALEY, J. M. *et al.* Plasma phospholipids identify antecedent memory impairment in older adults. **Nature medicine**, p. 415–418, 2014. Citado na página [2](#).
- MARCUS, M.; SANTORINI, B.; MARCINKIEWICZ, M.; TAYLOR, A. Treebank-3. In: LINGUISTIC DATA CONSORTIUM (LDC). **Web Download, Catalog LDC99T42**. [S.l.], 1999. Citado na página [28](#).
- MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of english: The penn treebank. **Computational Linguistics**, MIT Press, p. 313–330, 1993. Citado nas páginas [28](#) e [56](#).
- MARINHO, V. Q.; HIRST, G.; AMANCIO, D. R. Authorship attribution via network motifs identification. **arXiv preprint arXiv:1607.06961**, 2016. Citado na página [12](#).
- MATUSOV, E.; MAUSER, A.; NEY, H. Automatic sentence segmentation and punctuation prediction for spoken language translation. **IWSLT**, p. 158–165, 2006. ISSN 15206149. Citado na página [5](#).
- MAURER, A.; PONTIL, M.; ROMERA-PAREDES, B. The benefit of multitask representation learning. **Journal of Machine Learning Research**, p. 1–32, 2016. Citado na página [78](#).
- MCKHANN, G. M.; KNOPMAN, D. S.; CHERTKOW, H.; HYMAN, B. T.; JACK, C. R.; KAWAS, C. H.; KLUNK, W. E.; KOROSHETZ, W. J.; MANLY, J. J.; MAYEUX, R. *et al.* The diagnosis of dementia due to alzheimer’s disease: Recommendations from the national institute on aging-alzheimer’s association workgroups on diagnostic guidelines for alzheimer’s disease. **Alzheimer’s & dementia**, p. 263–269, 2011. Citado na página [2](#).
- MEDEIROS, H.; MONIZ, H.; BATISTA, F.; TRANCOSO, I.; NUNES, L. Disfluency detection based on prosodic features for university lectures. **INTERSPEECH**, p. 2629–2633, 2013. Citado nas páginas [79](#), [84](#), [85](#) e [156](#).
- MESULAM, M. *et al.* Primary progressive aphasia. **Annals of neurology**, Wiley Online Library, p. 425–432, 2001. Citado na página [67](#).
- MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013b. Citado nas páginas [36](#), [37](#) e [38](#).
- MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **NAACL**. [S.l.: s.n.], 2013. p. 746–751. Citado na página [36](#).
- MONIZ, H. **Processing Disfluencies in European Portuguese**. 176 p. Tese (Doutorado) — Instituto Superior Técnico - Universidade de Lisboa, 2013. Citado nas páginas [21](#), [40](#) e [41](#).

- MONIZ, H.; BATISTA, F.; MATA, A. I.; TRANCOSO, I. Speaking style effects in the production of disfluencies. **Speech Communication**, p. 20–35, 2014. Citado nas páginas 76 e 79.
- MONIZ, H.; BATISTA, F.; MEINEDO, H.; ABAD, A.; TRANCOSO, I.; MATA a.I.; MAMEDE, N. Prosodically-based automatic segmentation and punctuation. **Speech Prosody**, p. 3–6, 2010. Citado nas páginas 20 e 79.
- MONIZ, H.; FERREIRA, J.; BATISTA, F.; TRANCOSO, I.; LISBOA. Disfluency detection across domains. p. 1–6, 2015. Citado na página 79.
- MONIZ, H.; ISABEL, A.; TRANCOSO, I. A classificação das disfluências como mecanismos de (dis) fluência e os seus contextos prosódicos. **ENAPL**, p. 420–432, 2011. Citado nas páginas 79 e 146.
- MORRIS, J. C.; WEINTRAUB, S.; CHUI, H. C.; CUMMINGS, J.; DECARLI, C.; FERRIS, S.; FOSTER, N. L.; GALASKO, D.; GRAFF-RADFORD, N.; PESKIND, E. R. *et al.* The uniform data set (uds): clinical and cognitive variables and descriptive data from alzheimer disease centers. **Alzheimer Disease & Associated Disorders**, p. 210–216, 2006. Citado na página 2.
- MROZINSKI, J.; WHITTAKER, E. W.; CHATAIN, P.; FURUI, S. Automatic sentence segmentation of speech for automatic summarization. In: **ICASSP**. [S.l.: s.n.], 2006. Citado nas páginas 5, 61 e 128.
- MUANGPAISAN, W.; PETCHARAT, C.; SRINONPRASERT, V. Prevalence of potentially reversible conditions in dementia and mild cognitive impairment in a geriatric clinic. **Geriatrics & gerontology international**, p. 59–64, 2012. Citado na página 2.
- NAKATANI, C. H.; HIRSCHBERG, J. A corpus-based study of repair cues in spontaneous speech. **The Journal of the Acoustical Society of America**, p. 1603–1616, 1994. Citado na página 23.
- NIVRE, J.; HALL, J.; NILSSON, J. Maltparser: A data-driven parser-generator for dependency parsing. In: **LREC**. [S.l.: s.n.], 2006. Citado na página 158.
- NÓBREGA, F. A. A.; PARDO, T. A. S. Investigating machine learning approaches for sentence compression in different application contexts for portuguese. In: **PROPOR**. [S.l.: s.n.], 2016. p. 245–250. Citado na página 12.
- PALMER, D. D.; HEARST, M. A. Adaptive sentence boundary disambiguation. In: **ANLP**. [S.l.]: ACL, 1994. ISSN 0891-2017. Citado nas páginas 13 e 124.
- PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. **ICML**, p. 1310–1318, 2013. Citado na página 47.
- PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **EMNLP**. [S.l.: s.n.], 2014. p. 1532–1543. Citado nas páginas 37 e 59.
- PHAM, V.; BLUCHE, T.; KERMORVANT, C.; LOURADOUD, J. Dropout improves recurrent neural networks for handwriting recognition. In: **ICFHR**. [S.l.: s.n.], 2014. p. 285–290. Citado na página 108.
- PRETI, D. *et al.* **Análise de textos orais**. [S.l.]: Hvmánitas, 1997. Citado nas páginas 3 e 21.

- PRUD'HOMMEAUX, E.; ROARK, B. Graph-based word alignment for clinical language evaluation. **Computational Linguistics**, MIT Press, 2016. Citado na página 3.
- PRUD'HOMMEAUX, E.; MITCHELL, M.; ROARK, B. Using patterns of narrative recall for improved detection of mild cognitive impairment. **Age**, 2011. Citado na página 3.
- PRZYBOCKI, M. A.; MARTIN, A. F.; LE, A. N. Nist speaker recognition evaluation chronicles. In: IEEE. **Speaker and Language Recognition Workshop**. [S.l.], 2006. p. 1–6. Citado na página 30.
- QIAN, X.; LIU, Y. Disfluency detection using multi-step stacked learning. In: **ACL**. [S.l.: s.n.], 2013. p. 820–825. Citado nas páginas 6, 25, 80, 82, 84, 139, 140, 147 e 149.
- RAMÍREZ, J.; YÉLAMOS, P.; GÓRRIZ, J.; SEGURA, J. Svm-based speech endpoint detection using contextual speech features. **Electronics letters**, p. 426–428, 2006. Citado na página 64.
- READ, J.; DRIDAN, R.; OEPEN, S.; SOLBERG, L. J. Sentence boundary detection: A long solved problem? **COLING**, p. 985–994, 2012. Citado nas páginas 13 e 53.
- REYNAR, J. C.; RATNAPARKHI, A. A maximum entropy approach to identifying sentence boundaries. **ANLP**, p. 16–19, 1997. Citado na página 16.
- ROARK, B.; LIU, Y.; HARPER, M.; STEWART, R.; LEASE, M.; SNOVER, M.; SHAFRAN, I.; DORR, B.; HALE, J.; KRASNYANSKAYA, A. *et al.* Reranking for sentence boundary detection in conversational speech. In: IEEE. **ICASSP**. [S.l.], 2006. p. I–I. Citado na página 57.
- ROARK, B.; MITCHELL, M.; HOSOM, J.-P.; HOLLINGSHEAD, K.; KAYE, J. Spoken language derived measures for detecting mild cognitive impairment. **IEEE transactions on audio, speech, and language processing**, p. 2081–2090, 2011. Citado na página 3.
- RUDNICKY, A. I. Hub 4: Business broadcast news. In: **Proceedings of the DARPA Speech Recognition Workshop**. [S.l.: s.n.], 1996. p. 8–11. Citado na página 29.
- RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, P. R. *et al.* **Parallel distributed processing: Explorations in the microstructures of cognition. Volume 1: Foundations**. [S.l.]: The MIT Press, Cambridge, MA, 1986. Citado na página 42.
- SAFFRAN, E. M.; BERNDT, R. S.; SCHWARTZ, M. F. The quantitative analysis of agrammatic production: Procedure and data. **Brain and language**, p. 440–479, 1989. Citado nas páginas 67, 90 e 189.
- SARAWAGI, S.; COHEN, W. W. Semi-markov conditional random fields for information extraction. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2005. p. 1185–1192. Citado na página 80.
- SCHAPIRE, R. E.; SINGER, Y. Boostexter: A boosting-based system for text categorization. **Machine learning**, Springer, p. 135–168, 2000. Citado na página 63.
- SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, p. 2673–2681, 1997. Citado na página 46.
- SEARA, I. C. **Manual de Fonética Acústica**. 2013. Disponível em: <<https://moodle.ufsc.br/course/view.php?id=30043>>. Citado nas páginas 40 e 41.

- SERRANI, V. M. **Ambiente web de suporte à transcrição fonética automática de lemas em verbetes de dicionários do português do Brasil**. Tese (Doutorado), 2015. Citado na página [110](#).
- SHA, F.; PEREIRA, F. Shallow parsing with conditional random fields. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **NAACL**. [S.l.], 2003. p. 134–141. Citado na página [64](#).
- SHEN, D.; WEE, C.-Y.; ZHANG, D.; ZHOU, L.; YAP, P.-T. Machine learning techniques for ad/mci diagnosis and prognosis. In: **Machine Learning in Healthcare Informatics**. [S.l.: s.n.], 2014. p. 147–179. Citado na página [1](#).
- SHRIBERG, E. Spontaneous speech: how people really talk and why engineers should care. In: **INTERSPEECH**. [S.l.: s.n.], 2005. p. 1781–1784. Citado na página [20](#).
- SHRIBERG, E.; BATES, R. A.; STOLCKE, A. A prosody only decision-tree model for disfluency detection. In: **Eurospeech**. [S.l.: s.n.], 1997. p. 2383–2386. Citado nas páginas [6](#), [60](#), [77](#), [82](#), [85](#), [86](#), [134](#) e [139](#).
- SHRIBERG, E.; DHILLON, R.; BHAGAT, S.; ANG, J.; CARVEY, H. **The ICSI meeting recorder dialog act (MRDA) corpus**. [S.l.], 2004. Citado na página [34](#).
- SHRIBERG, E.; FAVRE, B.; FUNG, J.; HAKKANI-TÜR, D.; CUENDET, S. Prosodic similarities of dialog act boundaries across speaking styles. **Linguistic Patterns in Spontaneous Speech**, p. 213–239, 2009. Citado nas páginas [6](#), [16](#), [63](#), [71](#), [73](#), [119](#) e [121](#).
- SHRIBERG, E.; FERRER, L.; KAJAREKAR, S.; VENKATARAMAN, A.; STOLCKE, A. Modeling prosodic feature sequences for speaker recognition. **Speech Communication**, p. 455–472, 2005. Citado na página [64](#).
- SHRIBERG, E.; STOLCKE, A.; HAKKANI-TÜR, D.; TÜR, G. Prosody-based automatic segmentation of speech into sentences and topics. **Speech Communication**, p. 127–154, 2000. ISSN 01676393. Citado nas páginas [5](#), [6](#), [15](#), [18](#), [40](#), [53](#), [55](#), [56](#), [61](#), [62](#), [63](#), [65](#), [70](#), [71](#), [72](#), [73](#), [74](#) e [121](#).
- SHRIBERG, E. E. **Preliminaries to a theory of speech disfluencies**. Tese (Doutorado) — University of California at Berkeley, 1994. Citado nas páginas [20](#), [21](#), [23](#), [26](#), [75](#), [76](#), [82](#), [96](#), [99](#) e [163](#).
- SILVA, J. R.; BRANCO, A.; GONÇALVES, P. N. Top-performing robust constituency parsing of portuguese: Freely available in as many ways as you can get it. In: **LREC**. [S.l.: s.n.], 2010. Citado na página [158](#).
- SNOVER, M.; DORR, B.; SCHWARTZ, R. A lexically-driven algorithm for disfluency detection. **NAACL**, p. 157–160, 2004. Citado nas páginas [35](#), [82](#), [84](#) e [156](#).
- SOCHER, R. **Recursive deep learning for natural language processing and computer vision**. Tese (Doutorado) — Stanford University, 2014. Citado na página [42](#).
- SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, p. 427–437, 2009. Citado na página [25](#).

- SRIVASTAVA, N.; HINTON, G. E.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, p. 1929–1958, 2014. Citado na página [108](#).
- STAMATATOS, E.; FAKOTAKIS, N.; KOKKINAKIS, G. Automatic extraction of rules for sentence boundary disambiguation. In: **HLT-ML Workshop**. [S.l.: s.n.], 1999. p. 88–92. Citado na página [13](#).
- STERN, R. M. English broadcast news speech (hub4). In: **DARPA Speech Recognition Workshop**. [S.l.: s.n.], 1997. p. 7. Citado nas páginas [56](#) e [57](#).
- STEVENSON, M.; GAIZAUSKAS, R.; COURT, R.; STREET, P. Experiments on sentence boundary detection. **Information Retrieval**, p. 1–7, 1999. Citado nas páginas [13](#) e [93](#).
- STOLCKE, A.; SHRIBERG, E. Automatic linguistic segmentation of conversational speech. **ICSLP**, p. 1868–1871, 1996. Citado nas páginas [11](#), [15](#), [55](#), [58](#), [60](#), [72](#), [73](#), [74](#), [77](#) e [121](#).
- STOLCKE, A.; SHRIBERG, E.; BATES, R.; OSTENDORF, M.; HAKKANI-TÜR, D. Z.; PLACHE, M.; TÜR, G.; LU, Y. Automatic detection of sentence boundaries and disfluencies based on recognized words. **ICSLP**, 1998. Citado nas páginas [6](#), [55](#), [60](#), [61](#), [65](#), [77](#), [78](#), [82](#), [84](#), [85](#) e [95](#).
- STRASSEL, S. **Simple metadata annotation specification V5.0**. [S.l.]: Linguistic Data Consortium, 2003. Citado nas páginas [22](#), [23](#), [94](#), [98](#) e [195](#).
- _____. Linguistic resources for effective, affordable, reusable speech-to-text. In: **LREC**. [S.l.: s.n.], 2004a. Citado nas páginas [12](#), [15](#), [23](#) e [32](#).
- _____. **Simple metadata annotation specification V6. 2**. [S.l.]: Linguistic Data Consortium, 2004b. Citado na página [21](#).
- SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 3104–3112. Citado nas páginas [66](#) e [81](#).
- SUTTON, C.; MCCALLUM, A. *et al.* An introduction to conditional random fields. **Foundations and Trends® in Machine Learning**, p. 267–373, 2012. Citado nas páginas [58](#) e [141](#).
- TASKAR, B.; GUESTIN, C.; KOLLER, D. Max-margin markov networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2004. p. 25–32. Citado na página [80](#).
- TAYLOR, A.; MARCUS, M.; SANTORINI, B. The penn treebank: an overview. In: **Treebanks**. [S.l.: s.n.], 2003. p. 5–22. Citado na página [28](#).
- TEIXEIRA, C. V. L.; GOBBI, L. T. B.; CORAZZA, D. I.; STELLA, F.; COSTA, J. L. R.; GOBBI, S. Non-pharmacological interventions on cognitive functions in older people with mild cognitive impairment (mci). **Archives of gerontology and geriatrics**, p. 175–180, 2012. Citado na página [2](#).
- TIELEMAN, T.; HINTON, G. Rmsprop: Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural Networks for Machine Learning - Lecture 6.5**, 2012. Citado na página [110](#).

TILK, O.; ALUMÄE, T. Lstm for punctuation restoration in speech transcripts. 2015. ISSN 19909772. Citado nas páginas 9, 7, 64, 65, 66, 73, 74, 105, 119, 122 e 123.

_____. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. 2016. Citado nas páginas 7, 16, 66, 71, 72, 73, 108, 119, 164, 165 e 168.

TOLEDO, C. M. **Análise de aspectos micro e macrolinguísticos da narrativa de indivíduos com doença de Alzheimer, comprometimento cognitivo leve e sem comprometimentos cognitivos**. Tese (Doutorado) — Faculdade de Medicina da Universidade de São Paulo, 2017. Citado na página 89.

TREVISIO, M. V.; SHULBY, C.; ALUÍSIO, S. M. Sentence segmentation in narrative transcripts from neuropsychological tests using recurrent convolutional neural networks. **EACL**, p. 1–10, 2017. Citado nas páginas 11, 16, 7, 68, 69, 71, 72, 75, 90, 93, 116 e 120.

TUDINI, V. Repair and codeswitching for learning in online intercultural talk. **System**, p. 15–25, 2016. Citado na página 20.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: A simple and general method for semi-supervised learning. In: **ACL**. [S.l.: s.n.], 2010. p. 384–394. Citado nas páginas 36 e 37.

VASSIÈRE, J. Language-independent prosodic features. **Prosody: Models and measurements**, p. 53–66, 1983. Citado nas páginas 6, 39, 40 e 41.

WANG, S.; CHE, W.; LIU, T. A neural attention model for disfluency detection. **COLING**, p. 278–287, 2016. Citado nas páginas 81, 84, 86 e 147.

WANG, S.; CHE, W.; ZHANG, Y.; ZHANG, M.; LIU, T. Transition-based disfluency detection using lstms. **EMNLP**, p. 2775–2784, 2017. Citado nas páginas 7, 76, 82, 84, 86, 139, 140, 147, 149 e 156.

WANG, W.; TUR, G.; ZHENG, J.; AYAN, N. F. Automatic disfluency removal for improving spoken language translation. In: **ICASSP**. [S.l.: s.n.], 2010. p. 5214–5217. Citado nas páginas 5, 79, 83, 95, 135 e 148.

WANG, X.; NG, H. T.; SIM, K. C. Dynamic conditional random fields for joint sentence boundary and punctuation prediction. **INTERSPEECH**, p. 1382–1385, 2012. Citado na página 119.

_____. A beam-search decoder for disfluency detection. **COLING**, p. 1457–1467, 2014. Citado nas páginas 80, 81, 84, 85, 86, 139, 140, 141 e 147.

WAYNE, C. L. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In: **LREC**. [S.l.: s.n.], 2000. Citado na página 31.

WENGELIN, A. Disfluencies in writing-are they like in speaking? In: **ISCA Tutorial and Research Workshop**. [S.l.: s.n.], 2001. Citado na página 20.

WERBOS, P. **Beyond regression: New tools for prediction and analysis in the behavioral sciences**. Tese (Doutorado) — Harvard University, 1974. Citado na página 44.

WERBOS, P. J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, p. 1550–1560, 1990. Citado na página 46.

- XU, C.; XIE, L.; HUANG, G.; XIAO, X.; CHNG, E. S.; LI, H. A deep neural network approach for sentence boundary detection in broadcast news. **INTERSPEECH**, p. 2887–2891, 2014. Citado nas páginas 58 e 122.
- YANCHEVA, M.; FRASER, K.; RUDZICZ, F. Using linguistic features longitudinally to predict clinical scores for alzheimer’s disease and related dementias. In: **6th Workshop SLPAT**. [S.l.: s.n.], 2015. p. 134–140. Citado na página 3.
- YNGVE, V. H. A model and an hypothesis for language structure. **Proceedings of the American philosophical society**, 1960. Citado nas páginas 158 e 159.
- YOUNG, S.; EVERMANN, G.; GALES, M.; HAIN, T.; KERSHAW, D.; LIU, X.; MOORE, G.; ODELL, J.; OLLASON, D.; POVEY, D. *et al.* The htk book. **Cambridge university engineering department**, p. 175, 2002. Citado na página 110.
- ZAYATS, V.; OSTENDORF, M.; HAJISHIRZI, H. Disfluency detection using a bidirectional lstm. **CoRR**, 2016. Citado nas páginas 7, 81, 84, 85, 139, 140, 141, 142, 147, 156 e 165.
- ZHANG, X.; ZHAO, J.; LECUN, Y. Character-level convolutional networks for text classification. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 649–657. Citado na página 49.
- ZHOU, L.; ZHUANG, Z. Recovering punctuations in instant messages-towards the prosody norm in im. **AMCIS**, p. 408–417, 2004. Citado na página 16.
- ZIMMERMAN, M.; HAKKANI-TÜR, D.; FUNG, J.; MIRGHAFORI, N.; GOTTLIEB, L.; SHRIBERG, E.; LIU, Y. **The ICSI+ multilingual sentence segmentation system**. [S.l.], 2006. Citado na página 57.
- ZORRAQUINO, M. A. M.; PORTOLÉS, J. Los marcadores del discurso. **Gramática descriptiva de la lengua española**, Espasa Madrid, p. 4051–4213, 1999. Citado na página 21.

MANUAL DE ANOTAÇÃO DE PROPOSIÇÕES

O manual de anotação segue o que foi proposto por [Saffran, Berndt e Schwartz \(1989\)](#). As narrativas foram transcritas e anotadas previamente usando as marcações definidas na tabela abaixo por pesquisadores da Faculdade de Medicina da USP.

Marcação	Objetivo
...	para indicar pausa curta
((pausa <i>x</i> segundos))	para indicar pausa longa com cerca de <i>x</i> segundos
::	para indicar prolongamento de vogais
<disf> <i>texto</i> </disf>	para indicar disfluências
<empty> <i>texto</i> </empty>	para indicar trechos chamados de <i>fillers</i> ou comentários do paciente sobre a tarefa ou narrativa

Exemplo para a tabela acima:

1 a cinderela era uma moça que::: <empty>eu penso assim né</empty> que vivia numa fazenda mas ela era pobre e::: e ai uma fada resolveu ajudá-la e:: ai ...((pausa 3 segundos)) um príncipe ...((pausa 5 segundos)) tinha um sapato aonde ele achava que <disf>a pessoa a pessoa</disf> a moça que <empty>como se diz</empty> calçasse aquele calçado seria a esposa dele e assim ele tava procurando e essa moça o sapato serviu no pé dela e ai ela casou com o príncipe não era príncipe acho <empty>eu não me lembro exatamente</empty> e assim um final feliz aconteceu <empty>é isso mesmo? </empty> ((risos))

A nova anotação realizada nesta dissertação teve o objetivo de segmentar de maneira uniforme uma oração para que a segmentação sentencial fosse anotada em 1 de 22 proposições da estória da Cinderela. A anotação consistia em 3 passos, cada qual com suas etiquetas:

1. Remoção (via anotação) de trechos de textos chamados de *non-words*.

2. Segmentação de sentenças e julgamento se (+) ou (-), isto é, se é prosódica, sintática e semanticamente bem formada ou não.
3. Anotação de 22 proposições sobre as sentenças bem formadas.

A seguir anexamos o manual de anotação utilizado em cada um dos três passos.

Passo 1: Anotação de non-words

Cada trecho que corresponde a *non-words* deve ser marcado como:

1. **Neologismos:** invenção de palavras.
2. **Empty:** comentários dos pacientes que ainda não foram marcados.
3. **Falsos começos:** “ai”, “e então”, “bom”, etc.
4. **Conjunções coordenativas:** que une duas orações completas (e, mas, ou).
Anotar: Cinderela limpou o chão “e” ela lavou a roupa.
Não anotar: Cinderela limpou o chão “e” lavou a roupa.
5. **Marcadores do discurso direto:** “Ele disse, X”; “O príncipe disse, Y”. Não anotar os indiretos, como “Ela disse que ela gostaria de ir ao baile”.

A seguir colchetes indicam o que deve ser anotado.

6. **Repetições:** [Cinderela queria] Cinderela queria. Mas não anotar casos de ênfase como em: Cinderela disse por favor por favor me deixe ir.
7. **Interrupções:** [O pai odeia Romeo] o pai odeia o pai de Romeo
8. **Alterações:** O pai odeia [a mãe de Romeo] o pai de Romeo
9. **Elaborações:** O pai odeia [pai] o pai de Romeo
10. **Dêiticos:**
 - os advérbios com valor locativo (de lugar): aqui, ali, além, cá, lá, etc.
 - as locuções adverbiais com valor locativo: aqui perto, lá de cima, etc.

Exemplos: [aqui] é um senhor; [isso aqui] ..., [aqui] é o cavalo; [aqui] ... a menina ...

Exemplo de anotação:

1 ela morava com a madrasta as irmã <empty>né</empty> e ela era diferenciada das três era maltratada ela tinha que fazer limpeza na casa toda no castelo
 alias é as irmãs não faziam nada...((pausa 3 segundos)) até que um dia chegou um convite do rei ele ia fazer um baile...((pausa 3 segundos)) é
 a madrasta então <empty>é::: </empty> colocou que todas as filhas elas iam menos a cinderela... ((pausa 6 segundos)) bom como ela não tinha o vestido sapato as coisas
 tudo então ela mesmo teve que fazer a roupa dela começou a fazer as irmãs não queriam que ela fosse de jeito nenhum... <empty>eu sou muito resumista nas histórias
 ((risos))</empty> ... aí ela ficou triste aí apareceu uma fada madrinha que falou não você vai na festa sim ajudou ela a fazer o vestido dela arrumou uma carruagem
 feita de uma abobora tudo feito <disf>pela</disf> por essa fada madrinha... ((pausa 3 segundos)) e...explicou para ela <empty>né</empty> <empty>to no final da
 história</empty> já que ela fosse nesta festa mas a meia noite ela teria que vir embora porque tudo ia voltar ao normal...foi na festa o príncipe gostou muito dela
 dançaram bastante é quando chegou meia noite ela teve que sair só que aí perdeu o sapato na escadaria <disf>da</disf> do castelo mesmo o príncipe correndo
 atrás não deu tempo aí quando ele achou o sapato ele começou a procurar de casa em casa quem era dona desse sapatinho... bateu na casa da...masdrasda cinderela
 todas as meninas experimentaram o sapato em nenhuma serviu até que que ela resolveu experimentar na cinderela é era o sapato dela.... ((pausa 3
 segundos)) aí ele contou e casou com o príncipe <empty>acho que é isso né? </empty>

Passo 2: Anotação de sentenças e julgamento

Segmentar e marcar com (+) sentenças simples, subordinadas, com cláusulas relativas que sejam sintaticamente bem formadas. Sentenças com sujeito omissos são marcadas com (+), pois no português conseguimos recuperar o sujeito pela conjugação verbal.

Não há necessidade de serem semanticamente bem formadas a não ser que faltem argumentos necessários aos verbos. Nem sempre uma pausa é uma indicação de fim de sentença. Pontue com (-) caso contrário. Violações de restrições seletivas não ferem a boa formação da sentença. Assim, anotar com (+): O homem voa. Palavras com problemas ortográficos também não são problemas aqui.

Exemplo de anotação de sentenças e julgamento:

1 ela morava com a madrasta as irmã ela era diferenciada das três era maltratada ela tinha que fazer limpeza no castelo as irmãs não faziam nada...((pausa 3 segundos))
 um dia chegou um convite do rei ele ia fazer um baile...((pausa 3 segundos)) a madrasta colocou que elas iam menos a cinderela... ((pausa 6 segundos))
 como ela não tinha tudo então ela mesmo teve que fazer a roupa dela começou a fazer as irmãs não queriam que ela fosse de jeito nenhum... ... ela ficou triste
 apareceu uma fada madrinha que falou não você vai na festa sim ajudou ela a fazer o vestido dela arrumou uma carruagem feita de uma abobora
 tudo feito por essa fada madrinha... ((pausa 3 segundos))
 ...explicou para ela já que ela fosse nesta festa a meia noite ela teria que vir embora porque tudo ia voltar ao normal...foi na festa o príncipe gostou muito dela
 dançaram bastante quando chegou meia noite ela teve que sair só que perdeu o sapato na escadaria do castelo mesmo o príncipe correndo atrás não deu tempo
 quando ele achou o sapato ele começou a procurar de casa em casa quem era dona desse sapatinho... bateu na casa da...masdrasda todas as meninas experimentaram o sapato
 em nenhuma serviu ela resolveu experimentar na cinderela era o sapato dela.... ((pausa 3 segundos)) ele contou e casou com o príncipe

Passo 3: Anotação das proposições

Anotar sobre as sentenças bem formadas (+) uma das 28 proposições abaixo:

1. Mãe da Cinderela morre
2. Pai da Cinderela se casa novamente
3. Pai da Cinderela morre
4. Menina rica
5. Inveja (Madrasta e Filhas)
6. Limpando o sótão/serviçal
7. Deboche e malvadeza
8. Convite para o baile
9. Não permite Cinderela ir ao baile pois não tinha vestido para ir
10. Animais ajudam a fazer o vestido
11. Cinderela feliz com o vestido
12. Filhas da madrasta rasgam vestido da Cinderela
13. Refúgio para a floresta/chorando
14. Fada madrinha aparece
15. Fada madrinha medindo Cinderela para novo vestido
16. Momento da transformação/abobora-carruagem
17. Fada madrinha dá um vestido para a Cinderela
18. Fada madrinha avisa a Cinderela para voltar antes da meia noite
19. Cinderela foi para festa dançar
20. Príncipe conhece a Cinderela
21. Príncipe dança com Cinderela
22. Meia noite/Cinderela perde sapato na escada
23. Príncipe pega sapato e procura Cinderela
24. Madrasta prende Cinderela no sótão

- 25. Filha da madrasta experimenta o sapato e não serve
- 26. Animais soltam Cinderela/Experimenta sapato
- 27. Sapato serve no pé de Cinderela
- 28. Casamento/Feliz para sempre

Sendo que há a possibilidade de haver mais de uma proposição de mesmo tipo no texto.

MANUAL DE ANOTAÇÃO DE DISFLUÊNCIAS

Nosso manual de anotação de disfluências é baseado no manual utilizado por [Liu \(2004\)](#), que é o mesmo manual criado para as avaliações do NIST-RT 2004 ([STRASSEL, 2003](#)).

B.1 Primeira etapa: Preenchimentos

Os preenchimentos podem ser classificados como: **pausas preenchidas**; **marcadores discursivos** e **termos de edição explícitos**.

Pausas preenchidas

Pausas preenchidas são não-lexemas (*non-words*) que falantes empregam para indicar hesitação ou para manter controle de uma conversa enquanto pensam o que falar a seguir.

No português brasileiro as pausas preenchidas podem ser: “é”, “éh”, “ah”, “ahn”, “mm”, “umm”, etc. No exemplo abaixo temos duas delas.

ah a cinderela é a história de uma princesa ahn cuja mãe faleceu . e e o pai se casou novamente com ahn uma uma outra mulher que era viúva

Marcadores discursivos

Marcadores discursivos (MDs) são uma ou mais palavras que funcionam primariamente como uma unidade estrutural da língua falada. Geralmente, aparecem no início ou no fim de um enunciado, marcando a mudança do discurso.

Expressões comuns podem ser:

agora	então	bom
você sabe	olha	certo
tá?	né?	ok?

Por exemplo:

e ela tinha assim uma vida muito livre né . agora acontece que ela ela
perdeu o pai dela num acidente . **então** ela era adotada

É importante enfatizar que as palavras destacadas na lista acima podem aparecer como não sendo marcadores discursivos.

Os MDs “então” e “agora” são realmente difíceis de distinguir se são mecanismos de coesão textual, ou se cumprem alguma função relativa a interação dos interlocutores - de modo a serem sintaticamente independentes das unidades adjacentes. Portanto, deve-se anotar todos como sendo MDs a menos que hajam evidências fortes de que estão se comportando como mecanismos de coesão textual.

Por fim, O MD “olha” não deve ser confundido com a ação do interlocutor solicitar que o outro de fato olhe para o livro de figuras, como em: “olha aqui o cavalinho tomando água”.

Termo de edição explícito

São expressões que remetem explicitamente a uma mensagem de edição dentro de uma disfluência de edição. Ou seja, o interlocutor produz um enunciado seguidamente de uma expressão que representa um termo de edição explícito (podem ser uma ou mais palavras) e então a correção do enunciado original. Por exemplo:

ela foi pra casa digo pra festa

A palavra “digo” é um termo de edição explícito. **Importante:** pausas preenchidas e marcadores discursivos (subcategorias anteriores) também podem funcionar como um término explícito numa disfluência de edição.

Como anotar

No BRAT terão três classes para anotação:

- (pp) pausas preenchidas
- (md) marcadores discursivos
- (te) termo de edição explícito

Basta selecionar a parte do texto remetente a cada uma das três classes. Lembrando que a classe de (te) termo de edição explícito pode co-existir com as outras duas.

No caso de aparecer dois preenchimentos um após o outro, sejam eles do mesmo tipo ou não, deve-se selecionar uma etiqueta para cada um. Por exemplo, a marcação correta é: bom uhn a cinderela perdeu o pai né

Isso porque cada palavra é tratada como uma unidade básica de processamento pela ferramenta de detecção.

B.2 Segunda etapa: Disfluências de Edição

Disfluências de edição ocorrem quando o interlocutor faz um enunciado que não é completo e fluente, de modo que ele próprio corrige ou altera seu enunciado. A estrutura proposta é composta por:

[(enunciado original) * <termo de edição> correção]

- **Enunciado original:** também chamado de “reparando”, é a região do enunciado que é reparada por completo;
- **Ponto de interrupção (*):** é o ponto em que o interlocutor quebra o enunciado original com, por exemplo, um termo de edição explícito, repetição, revisão ou recomeço;
- **Termo de edição:** é opcional, e consiste de uma pausa preenchida, um marcador discursivo, ou um termo de edição explícito;
- **Correção:** é o trecho que corrige o enunciado original, sendo o que de fato o interlocutor quis enunciar.

A estrutura interna das disfluências de edição pode ser dividida em quatro subtipos:

- **Repetições:** O interlocutor repete parte do enunciado. Exemplo:

[(pra ela) * <ahn> pra ela] ir pro castelo

- **Revisões:** O interlocutor modifica o enunciado original com uma estrutura sintaticamente similar. Exemplo:

que já tinha [(uma filha) * duas filhas]

- **Recomeços:** O interlocutor abandona completamente o enunciado e começa de novo. Em recomeços, o trecho de correção pode ser vazio, significando que o enunciado original não foi repetido e nem revisto. Exemplo:

[(ela tá) *] aqui é uma noiva

- **Disfluências Complexas:** O interlocutor produz uma série de disfluências seguidas, ou recursivas. Exemplo de uma repetição dentro de uma revisão:

[(não tô (achando) * achando) * <ahn> não tô vendo] o sapatinho

Como anotar

No BRAT as transcrições já terão as anotações da etapa anterior. As classes dessa etapa serão divididas em:

- (eo-rep) enunciado original de uma repetição
- (eo-rev) enunciado original de uma revisão
- (eo-rec) enunciado original de um recomeço
- (pi) ponto de interrupção

Novamente, basta selecionar o texto e escolher a classe apropriadamente. Não há uma classe para termo de edição, pois ele já deve ter sido anotado na etapa anterior.

Nesse caso deve-se selecionar uma etiqueta para todo o enunciado original. Já para os pontos de interrupção, nenhuma etiqueta precisa ser selecionada, pois é possível obtê-la automaticamente a partir do fim do enunciado original. Por exemplo, a marcação correta é:

pra ela ahn pra ela
 eo-rep pi
 o pai morreu ahn pra ela
 eo-rep pi

Preenchimentos

Os preenchimentos na etapa anterior vão estar marcados, mas eles não foram removidos. Caso eles aparecerem no meio de uma disfluência, deve-se selecionar a etiqueta de EO por cima do preenchimento. Por exemplo:

se é pavariando se vangloriando
 eo-rev
 um dia o pai morreu não a mãe morreu
 eo-rev

Sequência de disfluências

Muitas vezes os pacientes produzem uma sequência de disfluências, portanto é importante anotar a região correta do enunciado original de cada uma. Por exemplo, veja que é anotada uma repetição e uma revisão em sequência, porém em regiões diferentes:

por isso que ela que ela que ele beijou ela
 eo-rep eo-rev

Repetições não triviais

Repetições podem ser bastante ambíguas de anotar. Deve-se prestar atenção em casos onde repete-se palavras para exaltar intensidade. Por exemplo:

a cinderela era muito muito bonita

e ela foi na festa dançou dançou dançou dançou

A fala produzida pelos pacientes é muitas vezes interrompida por eles mesmos, deixando palavras má formadas antes de a repetirem. Esses casos devem ser marcados como repetições. Por exemplo:

então a madri a madrinha
 eo-rep

as irmãs ado adotivas
 eo-rep

Mas perceba que casos em que a palavra anterior é um preenchimento (“uhm”) não caracterizam repetição:

então *uhm* um certo dia

Revisões não triviais

Na maioria dos casos, as revisões são alterações de uma palavra por outra ou possuem uma dica lexical após o ponto de interrupção que corresponde ao início do enunciado original (“do cachorro do cavalo”). Entretanto, podem ocorrer casos mais complicados, como por exemplo:

faleceu o pai dela a mãe faleceu
 eo-rev

Há casos também em que parece que o paciente trocou um adjetivo por outra, quando na verdade apenas adicionou mais adjetivos, o que não caracteriza uma revisão. Por exemplo:

uma pessoa tranquila serena

Disfluências complexas

Se aparecer uma disfluência complexa, deve-se ignorar a mais abrangente - mantendo a mais interna. O exemplo anterior deve ser anotado assim:

não tô achando achando *ahn* não tô vendo
eo-rep

Ou seja, foi marcada a repetição mas não a revisão, embora a marcação do “ahn” permaneceu, pois é uma pausa preenchida anotada na etapa anterior.

Outros exemplos:

com a a com a cinderela
eo-rep

fizeram teceram o teci o a o tecido lá da da roupa
eo-rev eo-rev eo-rev eo-rep

TRANSCRIÇÃO DE CCL APÓS PASSOS DO DEEPBONDD

Exemplo de uma transcrição de CCL após passar por passos do DeepBonDD.

Transcrição original

num país muito distante muito bonito é ahn nasceu uma linda é menina que recebeu o nome de cinderela aqui nesta gravura ela está junto com o pai num jardim acaraci acariciando um um um belo cavalo ai ela cresceu e continuou linda e e sempre muito feliz mas acontece que a mãe dela ah faleceu e o pai pegou e casou-se casou-se com uma uma moça que já tinha duas filhas ahn duas filhas essa madrasta muito muito má colocou é colocou-a para fazer os serviços domésticos judiando muito dela e e e também as as irmãs néentre aspas ahn também sempre que podiam ahn caçoava e judiava enfim até que um dia chegou um dia o príncipe ia dar um uma grande festa num palácio e resolveu mandar cartas pra todas as pessoas ahn da redondeza ai até que chegou na nas mãos da cinderela um convite e ela foi entregar para a a sua madrasta não sim ela ela a madrasta soube mas assim mesmo ela resolveu ahn ahn achando que ela poderia ir também ahn nessa festa porque ela também havia recebido o convite e os amiguinhos dela os ratinhos lá da vizinhança todo mundo se pôs a executar o vestido que ficou muito bonito mas as irmãs vendo isso é imediatamente rasgou toda a sua o seu vestido ahn arrebentou o seu colar enfim dizendo pra que ela não fazendo pra que ela não fosse não pudesse ir ao baile ela muito chorosa foi prum uma floresta e num cantinho num banco pôs-se a chorar nisso apareceu a fada madrinha que ahn ficando

sabendo do ocorrido resolveu ajudá-la e imediatamente ahn ti tirou suas medidas e apresentou uma um uma um vestido mais bonito ainda que o ou que o outro e verificando que não havia como ela chegar lá fez uma transformou uma abóbora em uma carruagem e mas dizendo a ela que pra ela tomar cuidado que o o o efeito da da o efeito da mágica iria ahn desfazer quando o relógio tocasse meia noite ai ela chegando no baile conheceu o príncipe dançou bastante com ele e as e as a a as suas irmãs muito invejosas olhando tudo ai quando o relógio deu meia noite to começou a tocar ela saiu correndo e na escadaria perdeu o sapatinho o príncipe saiu correndo atrás dela e só conseguiu ficar com o sapato ai o príncipe resolveu ahn saber quem era a moça e e e como não sabia lembrou que ahn o sapatinho iria talvez trazê-la de volta ai foi em todos os as casas do povoado e quando chegou na casa da da madrasta todas tentaram colocar o sapato mas em vão o pé muito grande e nenhuma entrou quando chegou a vez da branca de neve não da cinderela hanhan o o sapato coube direitinho ai todos ficaram felizes no palácio e ela contente casou e ahn ficou e ficaram felizes para sempre

Transcrição segmentada manualmente

num país muito distante muito bonito é ahn nasceu uma linda é menina que recebeu o nome de cinderela . aqui nesta gravura ela está junto com o pai num jardim acaraci acariciando um um um belo cavalo . ai ela cresceu e continuou linda e e sempre muito feliz . mas acontece que a mãe dela ah faleceu . e o pai pegou e casou-se casou-se com uma uma moça que já tinha duas filhas ahn duas filhas . essa madrasta muito muito má colocou é colocou-a para fazer os serviços domésticos judiando muito dela . e e e também as as irmãs né entre aspas ahn também sempre que podiam ahn caçoava e judiava . enfim até que um dia chegou um dia o príncipe ia dar um uma grande festa num palácio e resolveu mandar cartas pra todas as pessoas ahn da redondeza . ai até que chegou na nas mãos da cinderela um convite . e ela foi entregar para a a sua madrasta não sim . ela ela a madrasta soube mas assim mesmo ela resolveu ahn ahn achando que ela poderia ir também ahn nessa festa porque ela também havia recebido o convite . e os amiguinhos dela os ratinhos lá da vizinhança todo mundo se pôs a executar o vestido que ficou muito bonito . mas as irmãs vendo isso é imediatamente rasgou toda a sua o seu vestido ahn arreventou o seu colar enfim dizendo pra que ela não . fazendo pra que ela não fosse não pudesse ir ao baile . ela muito chorosa foi prum uma floresta e num

cantinho num banco pôs-se a chorar . nisso apareceu a fada madrinha que ahn ficando sabendo do ocorrido resolveu ajudá-la e imediatamente ahn tirou suas medidas e apresentou uma um uma um vestido mais bonito ainda que o ou que o outro . e verificando que não havia como ela chegar lá fez uma transformou uma abóbora em uma carruagem . e mas dizendo a ela que pra ela tomar cuidado que o o o efeito da da o efeito da mágica iria ahn desfazer quando o relógio tocasse meia noite . ai ela chegando no baile conheceu o príncipe dançou bastante com ele . e as e as a a as suas irmãs muito invejosas olhando tudo . ai quando o relógio deu meia noite to começou a tocar ela saiu correndo e na escadaria perdeu o sapatinho . o príncipe saiu correndo atrás dela e só conseguiu ficar com o sapato . ai o príncipe resolveu ahn saber quem era a moça . e e e como não sabia lembrou que ahn o sapatinho iria talvez trazê-la de volta . ai foi em todos os as casas do povoado . e quando chegou na casa da da madrastra todas tentaram colocar o sapato mas em vão o pé muito grande e nenhuma entrou . quando chegou a vez da branca de neve não da cinderela hanhan o o sapato coube direitinho . ai todos ficaram felizes no palácio . e ela contente casou e ahn ficou e ficaram felizes para sempre .

Transcrição segmentada automaticamente

num país muito distante muito bonito . é ahn nasceu uma linda é menina que recebeu o nome de cinderela . aqui nesta gravura ela está junto com o pai num jardim acaraci acariciando . um um um belo cavalo . ai ela cresceu e continuou linda e e sempre muito feliz . mas acontece que a mãe dela ah faleceu e o pai pegou e casou-se casou-se com uma uma moça que já tinha duas filhas ahn duas filhas . essa madrastra muito muito má colocou é colocou-a para fazer os serviços domésticos judiando muito dela . e e e também as as irmãs né entre aspas ahn também sempre que podiam ahn caçoava e judiava enfim até que um dia chegou um dia o príncipe ia dar um uma grande festa num palácio e resolveu mandar cartas pra todas as pessoas ahn da redondeza . ai até que chegou na nas mãos da cinderela um convite e ela foi entregar para a a sua madrastra não sim . ela ela a madrastra soube mas assim mesmo ela resolveu ahn ahn achando que ela poderia ir também ahn nessa festa porque ela também havia recebido o convite . e os amiguinhos dela os ratinhos lá da vizinhança todo mundo se pôs a executar o vestido que ficou muito bonito . mas as irmãs vendo isso é imediatamente rasgou toda a sua o seu vestido ahn arrebitou o seu colar . enfim dizendo pra que ela não fazendo pra que ela não fosse não

pudesse ir ao baile . ela muito chorosa foi prum uma floresta e num cantinho num banco pôs-se a chorar nisso apareceu a fada madrinha que ahn ficando sabendo do ocorrido resolveu ajudá-la . e imediatamente ahn ti tirou suas medidas . e apresentou uma um uma um vestido mais bonito . ainda que o ou que o outro . e verificando que não havia . como ela chegar lá fez uma transformou uma abóbora em uma carruagem . e mas dizendo a ela que pra ela tomar cuidado que o o o efeito da da o efeito da mágica iria ahn desfazer quando o relógio tocasse meia noite . ai ela chegando no baile conheceu o príncipe dançou bastante com ele . e as e as a a as suas irmãs muito invejosas olhando tudo . ai quando o relógio deu meia noite to começou a tocar ela saiu correndo e na escadaria perdeu o sapatinho . o príncipe saiu correndo atrás dela . e só conseguiu ficar com o sapato . ai o príncipe resolveu ahn saber quem era a moça . e e e como não sabia lembrou que ahn o sapatinho iria talvez trazê-la de volta . ai foi em todos os as casas do povoado . e quando chegou na casa da da madrastra todas tentaram colocar o sapato . mas em vão o pé muito grande . e nenhuma entrou . quando chegou a vez da branca de neve não da cinderela . hanhan o o sapato coube direitinho . ai todos ficaram felizes no palácio . e ela contente casou e ahn ficou e ficaram felizes para sempre .

Transcrição segmentada com disfluências removidas, ambas automaticamente

num país muito distante muito bonito . nasceu uma linda menina que recebeu o nome de cinderela . aqui nesta gravura ela está junto com o pai num jardim acaraci acariciando . um belo cavalo . ai ela cresceu e continuou linda e sempre muito feliz . mas acontece que a mãe dela faleceu e o pai pegou e casou-se casou-se com uma moça que já tinha duas filhas . essa madrastra muito má colocou colocou-a para fazer os serviços domésticos judiando muito dela . e também as irmãs entre aspas também sempre que podiam caçoava e judiava enfim até que um dia chegou um dia o príncipe ia dar uma grande festa num palácio e resolveu mandar cartas pra todas as pessoas da redondeza . ai até que chegou nas mãos da cinderela um convite e ela foi entregar para a sua madrastra não sim . ela a madrastra soube mas assim mesmo ela resolveu achando que ela poderia ir também nessa festa porque ela também havia recebido o convite . e os amiguinhos dela os ratinhos lá da vizinhança todo mundo se pôs a executar o vestido que ficou muito bonito . mas as irmãs vendo isso imediatamente rasgou toda seu vestido arreventou o seu colar . enfim dizendo pra que

ela não fazendo pra que ela não fosse não pudesse ir ao baile . ela muito chorosa foi prum uma floresta e num banco pôs-se a chorar nisso apareceu a fada madrinha que ficando sabendo do ocorrido resolveu ajudá-la . e imediatamente tirou suas medidas . e apresentou uma um vestido mais bonito . ainda que o ou que o outro . e verificando que não havia . como ela chegar lá fez uma transformou uma abóbora em uma carruagem . e mas dizendo a ela que pra ela tomar cuidado que da o efeito da mágica iria desfazer quando o relógio tocasse meia noite . ai ela chegando no baile conheceu o príncipe dançou bastante com ele . a as suas irmãs muito invejosas olhando tudo . ai quando o relógio deu meia noite to começou a tocar ela saiu correndo e na escadaria perdeu o sapatinho . o príncipe saiu correndo atrás dela . e só conseguiu ficar com o sapato . ai o príncipe resolveu saber quem era a moça . e como não sabia lembrou que o sapatinho iria talvez trazê-la de volta . ai foi em todos os as casas do povoado . e quando chegou na casa da madrastra todas tentaram colocar o sapato . mas em vão o pé muito grande . e nenhuma entrou . quando chegou a vez da branca de neve não da cinderela . o sapato coube direitinho . ai todos ficaram felizes no palácio . e ela contente casou e ficou e ficaram felizes para sempre .

NORMAS DE TRANSCRIÇÃO NURC

Normas Urbanas Cultas: NURC/SP n 338 EF, 331 D2 e 153 D2.

Ocorrências	Sinais	Exemplificação
Incompreensão de palavras ou segmentos	()	Do nível de renda... () Nível de renda nominal...
Hipótese do que se ouviu	(hipótese)	(estou) meio preocupado (com o gravador)
Truncamento (havendo homografia, usa-se acento indicativo da tônica e/ ou timbre)	/	E come/ e reinicia
Entonação enfática	maiúsculas	Porque as pessoas reTÊM moeda
Alongamento de vogal ou consoante (como s, r)	:: podendo aumentar para ::: ou mais	Ao emprestarem os... éh::: ... o dinheiro
Silabação	-	Por motivo tran-as-ção
Interrogação	?	E o banco... Central... certo?
Qualquer pausa	...	São três motivos... ou três razões... que fazem com que se retenha moeda... existe uma... retenção
Comentários descritivos do transcritor	((minúsculas))	((tossiu))
Comentário que quebram a seqüência temática da exposição; exposição; desvio temático	--	... a demanda da moeda – vamos dar essa notação – demanda de moeda por motivo
Superposição, simultaneidade de voz	ligando as linhas	A. na casa da sua irmã B. sexta-feira? A. fizeram lá... B. cozinham lá?
Indicação de que a fala foi tomada ou interrompida em determinado ponto. Não no seu início, por exemplo	(...)	(...) nós vimos que existem...
Citações literais, reprodução de discurso direto ou leituras de textos, durante a gravação	“ ”	Pedro Lima...ah escreve na ocasião... “O cinema falado em língua estrangeira não precisa de nenhuma baRREIra entre nós”...

Observações

- Iniciais maiúsculas: não se usam em início de períodos, turno e frases;
- Fáticos: ah, éh, eh, ahn, ehn, uhn, tá (não pôr está: tá? Você está brava?);
- Nomes de obras ou nomes comuns estrangeiros são grifados;
- Números por extenso;
- Não se indica o ponto de exclamação (frase exclamativa);
- Não anota o cadenciamento da frase;
- Podem-se combinar sinais. Por exemplo: oh::: ... (alongamento e pausa)
- Não se utilizam sinais de pausa, típicos da língua escrita, como ponto-e-vírgula, ponto final, dois pontos, vírgula. As reticências marcam qualquer tipo de pausa.