Análise retórica com base em grande quantidade de dados

Erick Galani Maziero

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:_____

Erick Galani Maziero

Análise retórica com base em grande quantidade de dados

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. VERSÃO REVISADA
Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP - São Carlos

Dezembro de 2016

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

Maziero, Erick Galani
M475a Análise retórica com base em grande quantidade de dados / Erick Galani Maziero; orientador Thiago Alexandre Salgueiro Pardo. -- São Carlos, 2016.
193 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

1. Processamento de Linguagem Natural. 2. Rhetorical Structure Theory. 3. Aprendizado semissupervisionado sem fim. 4. Grande quantidade de dados. I. Pardo, Thiago Alexandre Salgueiro, orient. II. Título.

Erick Galani Maziero

Rhetorical analysis based on large amount of data

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. FINAL VERSION
Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP - São Carlos

December 2016

...tudo o que for verdadeiro, tudo o que for nobre, tudo o que for correto, tudo o que for puro, tudo o que for amável, tudo o que for de boa fama, se houver algo de excelente ou digno de louvor, pensem nessas coisas.

Filipenses 4.8

Agradecimentos

Agradeço ao Deus Todo-Poderoso, Onipresente e Onisciente. Ao Senhor Jesus, meu Amigo. Ao Espírito Santo, meu Consolador.

A minha amada esposa Gesana, que abnegou de tantas coisas para me apoiar em minha formação. Eu te amo!

Aos meus amados pais Cláudio e Odete, que nunca mediram esforços para que eu pudesse chegar até aqui. Eles me ensinaram a valorizar o que é bom.

Ao meu querido irmão Henrique, amigo de todas as horas e meu incentivador.

Ao meu orientador, Dr. Thiago A. S. Pardo, por seu tempo e atenção, os quais foram fundamentais para minha formação. Por seu trabalho, ao qual admiro e tento imitar.

Aos meus amigos, que me ajudaram de todas as formas possíveis.

Thanks to Dr. Graeme Hirst, who supervised me during my research at University of Toronto. An admirable person and researcher.

À FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo (processo no 2011/23323-4), pelo apoio e suporte financeiro.

Conteúdo

1 Introdução **1**

2 Teorias Discursivas **13**

2.1 Grosz e Sidner Discourse Theory 14

2.2 Rhetorical Structure Theory 15

2.3 Relações semânticas de Jordan 20

2.4 (Segmented) Discourse Representation Theory 24

2.5 Relevance Theory 25

2.6 Relações semânticas de Kehler 25

2.7 Reestruturação de Wolf e Gibson 26

2.8 Outros modelos e representações discursivas 26

3 Trabalhos Correlatos **29**

3.1 Identificação das relações discursivas 29

3.1.1 Uso de padrões lexicais 29

3.1.2 Uso de Aprendizado de Máquina 34

3.1.2.1 Modelo probabilístico 34

3.1.2.2 Aprendizado supervisionado 37

3.1.2.3 Aprendizado semissupervisionado e não supervisionado . . 46

3.1.3 Mapa de abordagens e conclusões 51

3.2 Segmentação discursiva 52
3.2.1 Soricut & Marcu (2003) 52
3.2.2 Pardo (2005) 53
3.2.3 Hernault *et al* (2010) 54
3.2.4 Joty *et al* (2015) 54
3.3 Recursos e Ferramentas 55
3.3.1 Recursos 56
3.3.1.1 CSTNews 56
3.3.1.2 Rhetalho 58
3.3.1.3 Summ-it 59
3.3.1.4 CorpusTCC 60
3.3.1.5 Thesaurus para o Português 61
3.3.1.6 Listas de Palavras 62
3.3.2 Ferramentas 63
3.3.2.1 RSTTool 63
3.3.2.2 *Parser* Palavras 63
3.3.2.3 LX-Parser 65
3.3.2.4 MXPOST 67
3.3.2.5 Weka 67
3.3.2.6 Lematizador 67
3.3.2.7 Cadeias Lexicais 69
3.3.2.8 RSTEval 72
3.3.2.9 Crawler 72

4 Noções Básicas de AM 75

4.1 Paradigmas de aprendizado automático 77
4.1.1 Supervisionado 78
4.1.2 Não supervisionado 79
4.1.3 Semissupervisionado 79
4.2 Grandes quantidade de dados 82

4.2.1 Fluxos contínuos - *never-ending learning* 83
4.2.2 Mudança de conceito - *Concept drift* 85

5 Segmentação Discursiva 87

5.1 Segmentador baseado em regras lexico-sintáticas 87
5.1.1 As regras 88
5.1.2 Resultados 94

6 Identificação de Relações Discursivas e suas Nuclearidades 97

6.1 Aprendizado supervisionado 98
6.1.1 Adaptação do *parser* SPADE 102
6.1.2 Adaptação do *parser* HILDA 103
6.1.3 Novos atributos 105
6.1.4 Resultados 108
6.2 Aprendizado semissupervisionado sem fim 113
6.2.1 Arquitetura do *framework* de Semissupervisão Sem Fim 115
6.2.1.1 Aprendizado Automático 116
6.2.1.2 A grande quantidade de dados 118
6.2.1.3 Padrões Lexicais - *LPS* 119
6.2.1.4 Regras manuais 121
6.2.1.5 Controle de mudança de conceito (*concept drift*) 122
6.2.2 Resultados 123
6.2.2.1 Validação cruzada de 10 pastas 124
6.2.2.2 Conjunto de teste fixo 125
6.2.2.3 Experimento para o Inglês 129
6.2.2.4 Conclusões dos resultados 131
6.3 Nuclearidade 132
6.3.1 Ordem Canônica 133
6.3.2 Classificação com aprendizado supervisionado 135

7 Conclusões **139**

7.1 Contribuições 139

7.1.1 Teóricas 140

7.1.2 Práticas 140

7.1.3 Limitações e Dificuldades 141

7.2 Trabalhos futuros 141

A Conjunto das relações discursivas **173**

A.1 Descrição das relações retóricas 173

B Listas de Palavras **187**

B.1 Verbos de atribuição 187

B.2 Conjunções 189

Lista de Figuras

1.1 Níveis linguísticos 2

1.2 Exemplo de texto a ser analisado discursivamente. 3

1.3 Estrutura discursiva do texto exemplo da Figura 1.2. 6

1.4 Principais etapas da análise discursiva. 6

2.1 Exemplo da relação *Antithesis* 16

2.2 Exemplo da relação *Contrast* 17

2.3 Duas possíveis escolhas de nuclearidade para um mesmo par de segmentos, dependendo do que se quer enfatizar, uma causa (*Volitional-Cause*) ou um resultado (*Volitional-Result*) 18

2.4 Exemplo das relações estruturais *Parenthetical* e *Same-unit* 20

2.5 Exemplo de estruturação discursiva não arbórea (Wolf & Gibson (2005) - p. 266) 27

3.1 Exemplo simplificado de aplicação de padrão lexical no DiZer 33

3.2 Exemplo de árvore sintática lexicalizada. Os círculos indicam os nós internos mais informativos para realização da análise discursiva. Exemplo extraído de Soricut & Marcu (2003). 35

3.3 Exemplo de árvore discursiva. Exemplo extraído de Soricut & Marcu (2003) 36

3.4 Mapa das abordagens utilizadas no *parsing* discursivo. 51

3.5 Frequência das relações retóricas no córpus CSTNews 57

3.6 Frequência das relações retóricas no córpus Rhetalho 59
3.7 Frequência das relações retóricas no córpus Summ-it 60
3.8 Frequência das relações retóricas no córpus CorpusTCC 61
3.9 Ferrramenta de anotação RSTTool - (O'Donnell, 2000) 64
3.10 Exemplo de análise sintática pelo Palavras 66
3.11 Exemplo de análise sintática pelo LX-Parser 66
3.12 Interface do Weka, para treinamento de classificadores. 68

4.1 Etapas simplificadas para o uso do AM no tratamento de um problema. . . 76
4.2 Organização dos paradigmas de aprendizado automático. 78

6.1 Esquema do aprendizado supervisionado explorado. 98
6.2 Esquema do aprendizado semissupervisionado explorado. 114
6.3 Resultado do uso do *framework* de SSNEL para a adaptação HILDA-PT avaliado com validação estratificada cruzada de 10 pastas. 124
6.4 Resultado para SSNEL com $threshold > 0,3$ 126
6.5 Resultado para SSNEL com $threshold > 0,5$ 126
6.6 Resultado para SSNEL com $threshold > 0,6$ 127
6.7 Resultado para SSNEL com $threshold > 0,7$ 127
6.8 Resultado para SSNEL com $threshold > 0,8$ 128
6.9 Resultado para SSNEL com seleção de relações menos frequentes e $threshold > 0,7$ 128
6.10 Resultado para SSNEL utilizando apenas classificadores e $threshold > 0,7$ 129
6.11 Resultado para SSNEL utilizando apenas classificadores e $threshold > 0,7$ para o Inglês 131

Lista de Tabelas

2.1 Definição da relação *Antithesis* 15

2.2 Definição da relação *Contrast* 16

2.3 Conjunto original de relações propostas na RST 19

2.4 Lista das relações semânticas de Jordan. 21

2.4 Lista das relações semânticas de Jordan. 22

2.4 Lista das relações semânticas de Jordan. 23

2.4 Lista das relações semânticas de Jordan. 24

3.1 Conjunto de atributos adaptados do *parser* HILDA. Os atributos são agrupados em dois conjuntos: Organização Textual e Sintáticos (relacionados aos conjuntos de dominância). 43

3.2 Pistas lexicais utilizadas por Marcu & Echihabi (2002) para obtenção automática de instâncias de treinamento para a criação de classificadores. . . 47

3.3 Estatísticas do córpus CSTNews 56

3.4 Concordância da anotação RST no CSTNews 58

3.5 Estatísticas do córpus Rhetalho 58

3.6 Estatísticas do córpus Summ-it 59

3.7 Estatísticas do córpus CorpusTCC 61

3.8 Estatísticas do TeP2 - Thesaurus para o Português do Brasil 62

5.1 Regra de segmentação 1. 89

5.2 Regra de segmentação 2. 90
5.3 Regra de segmentação 3. 90
5.4 Regra de segmentação 4. 90
5.5 Regra de segmentação 5. 90
5.6 Regra de segmentação 6. 91
5.7 Regra de segmentação 7. 91
5.8 Regra de segmentação 8. 91
5.9 Regra de segmentação 9. 92
5.10 Regra de segmentação 10. 92
5.11 Regra de segmentação 11. 92
5.12 Regra de segmentação 12. 92
5.13 Regra de segmentação 13. 93
5.14 Regra auxiliar na segmentação 14. 93
5.15 Regra auxiliar na segmentação 15. 93
5.16 Regra auxiliar na segmentação 16. 93
5.17 Regra de restrição 17. 94
5.18 Regra de restrição 18. 94
5.19 Avaliação do segmentador baseado em regras. 95
5.20 Matriz de confusão da segmentação realizada pelo SegRegras 96

6.1 Número de documentos e palavras do conjunto de corpora RST para o Português (composto dos corpora CSTNews, Rhetalho, Summ-it e CorpusTCC). 99
6.2 Agrupamento das 29 relações retóricas do nível intrassentencial disponíveis nos corpora RST para o Português em 16 grupos e suas frequências. 100
6.3 Agrupamento das 29 relações retóricas do nível inter-sentencial disponíveis nos corpora RST para o Português em 17 grupos e suas frequências. 101
6.4 Atributos da organização textual. 106
6.5 Atributos da morfossintaxe. 106
6.6 Atributos da sintaxe. 107

6.7 Atributos da semântica e discurso. 108

6.8 Resultados das adaptações (SPADE-PT e HILDA-PT) em comparação com os *baselines* (Relação *Elaboration* e DiZer) 109

6.9 Medida-F para cada relação tratada pelo HILDA-PT, considerando o agrupamento de relações (Tabela 6.2). A medida-F foi ponderada de acordo com a frequência de cada relação/grupo de relações no conjunto de teste. 110

6.10 Matriz de confusão do HILDA-PT 111

6.11 Resultados dos experimentos com aprendizado supervisionado para a identificação de relações discursivas no nível intrassentencial, considerando diversos grupos de atributos, em que OT é Organização Textual, M é Morfossintaxe, S é Semântica e D é Discurso. 112

6.12 Resultados dos experimentos com aprendizado supervisionado para a identificação de relações discursivas no nível inter-sentencial, considerando diversos grupos de atributos, em que M é Morfossintaxe, S é Semântica e D é Discurso 112

6.13 Exemplo de padrão lexical utilizado para identificação do grupo de relações de causa/resultado. 120

6.14 Exemplo de padrão lexical utilizado para identificação do grupo de relações de causa/resultado (*Cause-Result*) e *Evidence-Justify-explanation*. 120

6.15 Regra Attribution 1. 121

6.16 Regra Attribution 2. 121

6.17 Regra Attribution 3. 121

6.18 Regra Parenthetical 4. 121

6.19 Valor máximo da medida-F obtido em cada experimento SSNEL. Também é apresentada a quantidade de iterações necessárias para se chegar ao valor e a quantidade de iterações executadas. 130

6.20 Tamanho dos corpora RST-PT para o Português, seus componentes, e do córpus para o Inglês (EN), o RST-DT. 130

6.21 Ordem canônica da nuclearidade nas relações retóricas. O valor em negrito representa a maior ocorrência nos corpora RST para o nível intrassentencial. 134
6.22 Ordem canônica da nuclearidade nas relações retóricas. O valor em negrito representa a maior ocorrência nos corpora RST para o nível inter-sentencial. 136
6.23 Resultados dos experimentos com aprendizado supervisionado para a identificação de nuclearidade no nível intrassentencial, considerando diversos grupos de atributos, em que OT é Organização Textual, M é Morfossintaxe, S é Semântica e D é Discursos. 137
6.24 Resultados dos experimentos com aprendizado supervisionado para a identificação de nuclearidade no nível inter-sentencial, considerando diversos grupos de atributos, em que M é Morfossintaxe, S é Semântica e D é Discurso. 138

A.1 Agrupamento das relações retóricas utilizadas nesta pesquisa 173
A.2 Definição da relação *Antithesis* 175
A.3 Definição da relação *Attribution* 175
A.4 Definição da relação *Circumstance* 175
A.5 Definição da relação *Comparison* 176
A.6 Definição da relação *Concession* 176
A.7 Definição da relação *Elaboration* 176
A.8 Definição da relação *Conclusion* 177
A.9 Definição da relação *Condition* 177
A.10 Definição da relação *Enablement* 177
A.11 Definição da relação *Evidence* 178
A.12 Definição da relação *Justify* 178
A.13 Definição da relação *Evaluation* 178
A.14 Definição da relação *Explanation* 179
A.15 Definição da relação *Interpretation* 179
A.16 Definição da relação *Means* 179
A.17 Definição da relação *Motivation* 180

A.18 Definição da relação *Non-Volitional Cause* 180
A.19 Definição da relação *Non-Volitional Result* 181
A.20 Definição da relação *Otherwise* 181
A.21 Definição da relação *Parenthetical* 182
A.22 Definição da relação *Purpose* 182
A.23 Definição da relação *Restatement* 182
A.24 Definição da relação *Solutionhood* 183
A.25 Definição da relação *Summary* 183
A.26 Definição da relação *Volitional Cause* 183
A.27 Definição da relação *Volitional Result* 184
A.28 Definição da relação *Contrast* 185
A.29 Definição da relação *Joint* 185
A.30 Definição da relação *List* 185
A.31 Definição da relação *Same-unit* 186

B.1 Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas. 189
B.1 Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas. 190
B.1 Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas. 191
B.1 Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas. 192
B.1 Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas. 193

Lista de Algoritmos

1 Algoritmo do lematizador 68

2 Algoritmo para identificar cadeias lexicais 70

3 Algoritmo do web crawler 73

4 Algoritmo simplificado do *self-training* 80

5 Algoritmo simplificado do *co-training* 81

6 Algoritmo do segmentador discursivo 89

7 Algoritmo do SSNEL utilizando dois modelos de identificação de relações retóricas entre cada $PairUBDs$. 117

Resumo

Maziero, E. G. (2016). *Análise retórica com base em grande quantidade de dados.* Tese (Doutorado em Ciências - Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

Com uma quantidade quase incontável de informação textual disponível na web, a automatização de diversas tarefas referentes ao processamento automático de textos é uma necessidade inegável. Em abordagens superficiais do PLN (Processamento da Linguagem Natural), importantes propriedades do texto são perdidas, como posição, ordem, adjacência e contexto dos segmentos textuais. Uma análise textual mais profunda, como a realizada no nível do discurso, ocupa-se da busca e identificação da organização retórica do texto, gerando uma estrutura hierárquica em que as intenções do autor são explicitadas e relacionadas entre si. Para a automatização dessa tarefa, tem-se utilizado técnicas de aprendizado automático, predominantemente do paradigma supervisionado. Nesse paradigma, são necessários dados rotulados manualmente para a geração dos modelos de classificação. Como a anotação para essa tarefa é algo custoso, os resultados obtidos no aprendizado são insatisfatórios, pois estão bem aquém do desempenho humano na mesma tarefa. Nesta tese, o uso massivo de dados não rotulados no aprendizado semissupervisionado sem fim foi empregado na tarefa de identificação das relações retóricas. Foi proposto um *framework* que utiliza textos obtidos continuamente da web. No *framework*, realiza-se a monitoração da mudança de conceito, que pode ocorrer durante o aprendizado contínuo, e emprega-se uma variação dos algoritmos tradicionais de semissupervisão. Além disso, foram adaptados para o Português técnicas do estado da arte. Sem a necessidade de anotação humana, a medida-F melhorou, por enquanto, em 0,144 (de 0,543 para 0,621). Esse resultado consiste no estado da arte da análise discursiva automática para o Português.

Palavras-chave: *Rhetorical Structure Theory, aprendizado semissupervisionado sem fim, grande quantidade de dados*

Abstract

Maziero, E. G. (2016). *Análise retórica com base em grande quantidade de dados.* Tese (Doutorado em Ciências - Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

Considering the almost uncountable textual information available on the web, the automatization of several tasks related to the automatic text processing is an undeniable need. In superficial approaches of NLP (Natural Language Processing), important properties of the text are lost, as position, order, adjacency and context of textual segments. A deeper analysis, as carried out in the discursive level, deals with the identification of the rhetoric organization of the text, generating a hierarchical structure. In this structure, the intentions of the author are identified and related among them. To the automatization of this task, most of the works have used machine learning techniques, mainly from the supervised paradigm. In this paradigm, manually labeled data is required to obtain classification models, specially to identify the rhetorical relations. As the manual annotation is a costly process, the obtained results in the task are unsatisfactory, because they are below the human perfomance. In this thesis, the massive use of unlabeled data was applied in a semi-supervised never-ending learning to identify the rhetorical relations. In this exploration, a framework was proposed, which uses texts continuously obtained from the web. In the framework, a variation of traditional semi-supervised algorithms was employed, and it uses a concept-drift monitoring strategy. Besides that, state of the art techniques for English were adapted to Portuguese. Without the human intervention, the F-measure increased, for while, 0.144 (from 0.543 to 0.621). This result consists in the state-of-the-art for Discourse Analysis in Portuguese.

Keywords: *Rhetorical Structure Theory, semi-supervised never-ending learning, large amount of data*

1 Introdução

Mesmo antes de 1950, pesquisas relacionadas à linguagem natural têm mostrado cunho prático, especificamente na tarefa de tradução automática (Hutchins, 2010), do Russo para o Inglês, no cenário da Guerra Fria, especialmente entre Estados Unidos e Rússia. Hoje, com uma quantidade quase incontável de informação textual disponível na web, a automatização de diversas tarefas referentes ao processamento automático de textos, além da tradução automática, é uma necessidade inegável.

O processamento de dados em formato textual pode ser realizado de diversas maneiras, desde o uso de *bag-of-words* até o uso de conhecimentos linguísticos profundos. Em abordagens como *bag-of-words*, muitas propriedades textuais úteis são perdidas, tais como *posição*, *ordem*, *adjacência* e *contexto* das palavras (Webber & Joshi, 2012). Essas propriedades são obtidas no nível discursivo de um texto e ao longo dos anos tem sido amplamente utilizadas em tarefas do Processamento de Linguagem Natural (PLN).

O processamento da linguagem natural ocupa-se desde a classificação das palavras em suas classes morfossintáticas (como verbos, substantivos, conjunções, etc.), passando pela estruturação sintática do texto (gerando uma estrutura arbórea para cada sentença, explicitando os substantivos, verbos, predicados, objetos, etc.), indo pela identificação dos significados (semântica) das palavras, expressões e orações, e atinge o nível discursivo. Os níveis linguísticos são exibidos na Figura 1.1 (Jurafsky e Martin, 2009) e o discursivo ocupa-se da estrutura retórica do texto e trata diversos fenômenos, como anáforas e correferências, aspectos textuais, identificação dos tópicos e relações retóricas entre as

proposições textuais.

Figura 1.1: **Níveis linguísticos**

Na análise do discurso, especificamente, há o estudo das relações entre as proposições textuais, que podem variar de palavras a segmentos maiores que uma sentença. Dentre as diversas teorias discursivas, a *Rhetorical Structure Theory* - RST (Mann & Thompson, 1987) é a mais proeminente no PLN e trata desse tipo de análise relacional. A proeminência dessa teoria pode ser confirmada pela quantidade de trabalhos que a utilizam em diversas tarefas do processamento da linguagem natural. O rol de relações, que pode ser expandido, e o conceito de nuclearidade dos segmentos tornam a RST uma teoria bem informativa e passível de tratamento computacional.

O texto exemplo, da Figura 1.2, tem diversas orações e sentenças que são apresentadas de forma coerente, e não aleatória, com a finalidade de atender ao intento do autor na produção do texto. Discurso é, portanto, esse conjunto de partes textuais, ou segmentos discursivos, apresentados de forma coerente.

Na sentença "Mesmo estando muito doente, não faltou à aula.", é apresentada uma concessão feita pelo escritor, pois se espera que o aluno falte à aula quando está muito doente. Essa sentença também serve de justificativa para o leitor do texto aceitar melhor que "Henrique é um aluno muito dedicado". A sentença em que se apresenta que o aluno é dedicado é mais importante que a sentença que evidencia tal informação. Concessão e

justificativa são duas relações presentes no texto e são chamadas de relações discursivas, ou de coerência, por relacionarem porções de texto, estruturando a coerência entre as mesmas, assim como definindo a importância dessas porções textuais.

HENRIQUE É UM ALUNO MUITO DEDICADO.
MESMO ESTANDO MUITO DOENTE,
NÃO FALTOU À AULA.
QUANDO INTERROGADO POR
UM DE SEUS PROFESSORES SOBRE SEU
EMPENHO, ELE DISSE SER
UM ENTRETENIMENTO O APRENDER.

Figura 1.2: **Exemplo de texto a ser analisado discursivamente.**

Considerando as informações provenientes desse tipo de análise, a sumarização automática pode se beneficiar da categorização de segmentos textuais em importância, podendo desprezar, na composição do sumário, os segmentos menos importantes do texto. Por exemplo, em sumarização automática multidocumento, esse conhecimento pode ser utilizado em conjunto com outros conhecimentos, como a CST (uma teoria de estruturação multi-documento (Radev, 2000)), na composição do sumário final (Cardoso *et al*, 2011b).

Um sistema de perguntas e respostas pode, com conhecimento discursivo, responder perguntas baseando-se na estrutura discursiva dos textos. Bosma (2005) utiliza a estrutura discursiva dos textos para estender as respostas dadas automaticamente, provendo informação adicional e possibilitando melhor acurácia do sistema. Verberne *et al* (2007) utilizam o conhecimento discursivo em perguntas do tipo *wh-questions* na extração da resposta, considerando que o tópico da pergunta corresponde a algum segmento do texto e que esse segmento está relacionado, por algum relação RST, a outro segmento que contenha a resposta.

Com relação à tradução automática, trabalhos mostram que a estrutura discursiva de um texto na língua de origem e na língua alvo pode ser distinta. Desta forma, Mitkov (1993) e Marcu *et al* (2000) exploraram como o discurso pode auxiliar na tradução automática, principalmente no tocante à tradução de parágrafos e não sentença por sentença. Os autores não buscaram manter a mesma estrutura discursiva da língua alvo, mas criaram metodologias de mapeamento da estrutura discursiva do texto de origem em outra estrutura discursiva da língua alvo. Recentemente, Iruskieta *et al* (2015) estudaram um córpus anotado com discurso paralelamente em Inglês, Espanhol e Basco para verificar as diferenças linguísticas entre as estruturas discursivas dos textos. Essa análise gerou estratégias de tradução entre as línguas, a partir das estruturas discursivas.

A análise de sentimentos é melhorada pelo uso de conhecimento discursivo, como evidenciado por Balage F. (2012). O autor trata a análise de sentimentos como um problema de classificação de textos, de acordo com o sentimento que ele representa. O discurso é utilizado para aumentar ou diminuir a importância de sentenças do texto e melhorar a classificação do mesmo. Taboada (2016) utiliza o discurso para guiar a orientação semântica de palavras, a fim de definir sua polaridade. O discurso ajuda fornecendo informação contextual às palavras.

As pesquisas realizadas com discurso para o Português do Brasil valem-se da disponibilidade de recursos anotados segundo a RST. Em especial, pela disponibilidade de quatro corpora com anotação RST, a saber, CSTNews (Cardoso *et al*, 2011a), CorpusTCC (Pardo & Nunes, 2004), Rhetalho (Pardo & Seno, 2005) e Summ-It (Collovini *et al*, 2007). Tais informações possibilitam o desenvolvimento de trabalhos baseados na informação discursiva e diversas pesquisas têm sido assim embasadas com o uso desses corpora (Cardoso *et al*, 2013; Jorge *et al*, 2014; Dias & Pardo, 2015; Cardoso *et al*, 2015; Cardoso & Pardo, 2016).

Além das aplicações citadas, o conhecimento discursivo pode ser utilizado para estruturar apresentações de conteúdos da web em forma de histórias (Nakasone *et al*, 2006), modelar o diálogo de interação homem-robô (Makatchev *et al*, 2009), melhorar a compressão de sentenças (Clarke & Lapata, 2010), avaliar a coerência de um texto (Lin & Kan,

2011), melhorar a tarefa de Extração de Informação (Lioma *et al*, 2012) e de sistemas de busca na web (Inam *et al*, 2012) pela reformulação do texto utilizado na busca.

O foco deste trabalho é a identificação das relações retóricas entre porções de texto, levando a uma estrutura do discurso (estrutura discursiva), em que as porções textuais são relacionadas, iterativamente, até que todo o texto forme uma estrutura única. No exemplo, poderíamos segmentar o texto em suas orações, que algumas vezes correspondem a sentenças:

1. Henrique é um aluno muito dedicado.

2. Mesmo estando muito doente,

3. não faltou à aula.

4. Quando interrogado por um de seus professores sobre seu empenho,

5. ele disse ser um entretenimento o aprender.

Considerando a RST - *Rhetorical Structure Theory* (Mann & Thompson, 1987), que foi proposta inicialmente com vistas ao estudo da geração textual, gera-se uma estrutura conforme ilustrada na Figura 1.3. Nessa estrutura, o texto exemplo foi segmentado em cinco porções textuais que são chamadas, na teoria, de Unidades Básicas do Discurso (UBDs), numeradas de 1 a 5.

O relacionamento entre os segmentos 2 e 3 mostra uma concessão (*Concession*) feita pelo escritor, em que, mesmo o aluno estando doente, foi à escola. Isso é explicado (*Explanation*) pelos segmentos 4 e 5, que por sua vez são relacionados pela relação *Circumstance*, pois o segmento 4 apresenta a circunstância na qual o aluno disse ser o aprendizado um entretenimento para ele. Os segmentos 2 a 5 servem como uma justificativa (*Justify*) para o segmento 1, em que se afirma que Henrique é um aluno muito dedicado.

Cada segmento, em uma dada relação, pode ser considerado mais importante (núcleo) ou menos (satélite) de acordo com as intenções do autor, capturadas pelas relações retóricas. A essa noção de importância, dá-se o nome de nuclearidade. Por exemplo, o

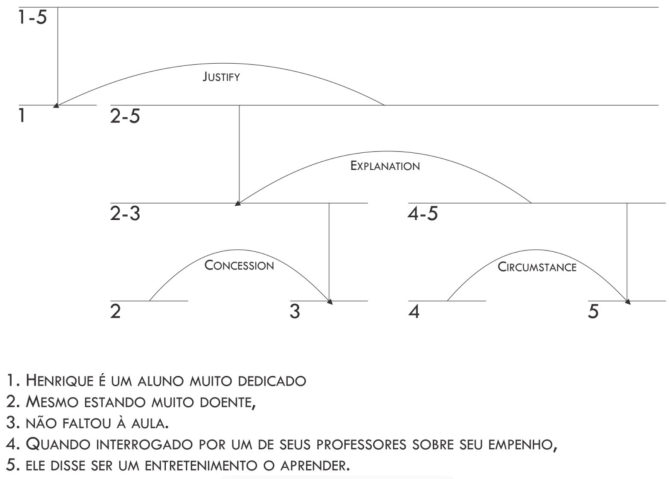


Figura 1.3: **Estrutura discursiva do texto exemplo da Figura 1.2.**

segmento 4 apresenta uma circunstância (*Circumstance*) para o segmento 5. Nessa relação, o segmento 5 é o segmento nuclear, enquanto que o segmento 4 é satélite. Na ilustração, a nuclearidade é indicada pela direção da seta que relaciona os segmentos, partindo do satélite para o núcleo. A cada relação entre um par de segmentos (UBDs, em inglês, *Elementary Discourse Units* - EDUs), forma-se um *span*, que, por sua vez, poderá ser relacionado a outro segmento ou *span*. Entende-se *span* como um segmento textual formado por uma ou mais UBDs.

Formou-se, assim, uma estrutura hierárquica que abrangeu todo o texto, estruturando sua coerência. Nessa estrutura, os segmentos textuais são as folhas e os nós internos são as relações retóricas entre os segmentos textuais. Esse tipo de representação abstrata é referido como uma estrutura discursiva do texto (Figura 1.3).

Figura 1.4: **Principais etapas da análise discursiva.**

Em termos gerais, a análise discursiva segundo a RST pode ser realizada em três

passos principais: segmentação do texto, identificação das relações entre os segmentos e suas nuclearidades e construção da árvore discursiva, conforme é esquematizado na Figura 1.4.

Diversas técnicas, conhecimentos e metodologias têm sido empregados na tarefa de identificar automaticamente as relações retóricas presentes em um texto, possibilitando montar sua estrutura discursiva. Para o Inglês, inicialmente utilizou-se uma metodologia baseada em padrões discursivos (Marcu, 1997, 2000; Corston-Oliver, 1998), com a restrição de identificar relações com marcas explícitas no texto, por exemplo, a presença de marcadores discursivos. Posteriormente, empregou-se técnicas de aprendizado automático (Reitter, 2003; Soricut & Marcu, 2003; Mahmud & Ramsay, 2005; Wellner *et al*, 2009; duVerle & Prendinger, 2009; Hernault *et al*, 2010; Feng & Hirst, 2012; Joty *et al*, 2015), técnicas de recuperação de informação (Schilder, 2002), informações semânticas e lógica de primeira ordem (Subba & Di Eugenio, 2009) e algoritmos de *shift-reduce* sobre informação sintática (Sagae, 2009).

Marcu & Echihabi (2002) e Sporleder & Lascarides (2008) realizaram um processo em que, antes de aplicar técnicas de aprendizado automático, aumentaram o conjunto de dados de treinamento pela rotulação automática de exemplos com o uso de padrões textuais. Chiarcos (2012) realizou experimentos de aprendizado semissupervisionado, buscando mapear probabilisticamente a ocorrência de marcadores discursivos e verbos a relações retóricas. Esses trabalhos buscaram aumentar, de forma não supervisionada, o conjunto de dados rotulados, tendo em vista que rotular dados manualmente é uma tarefa custosa.

Para o Português do Brasil, tem-se apenas uma abordagem na tarefa automática de análise discursiva (Pardo & Nunes, 2008). Essa abordagem baseia-se exclusivamente no uso de padrões textuais para a identificação das relações retóricas entre as porções de um texto e obtém resultado limitado por essa metodologia, ou seja, essa metodologia consegue captar apenas as relações retóricas que são indicadas por marcas explícitas, como marcadores discursivos. Posteriormente, esse trabalho foi disponibilizado online por Maziero *et al* (2011), permitindo a criação de repositórios de padrões discursivos em

diversas línguas, além do Português, como o Espanhol (Cunha *et al*, 2012) e o Inglês.

Tanto a utilização de padrões textuais quanto o uso de aprendizado supervisionado são muito dependentes de exemplos anotados segundo a teoria discursiva em uso, além de serem abordagens muito dependentes do domínio dos textos utilizados. Assim, sente-se a necessidade de desenvolver uma metodologia de análise do discurso que vá além da utilização de padrões textuais e que não dependa apenas de dados rotulados, produzindo uma metodologia de análise discursiva mais robusta e com melhores resultados, visto que os resultados ainda estão distantes da capacidade humana de análise.

Exceto os trabalhos de Marcu & Echihabi (2002), Hernault *et al* (2010) e Chiarcos (2012) para a língua inglesa, os trabalhos citados para ambas as línguas dependem de dados rotulados dos quais se possa extrair conhecimento discursivo, seja para a definição de padrões textuais, seja para gerar modelos por aprendizado automático.

Como uma abordagem que independe de apenas dados rotulados, tem-se o aprendizado semissupervisionado, que é utilizado em cenários com disponibilidade de dados não rotulados e a rotulação é uma tarefa custosa (Zhu, 2008). Uma ideia relacionada ao aprendizado semissupervisionado é a aplicação do mesmo em várias iterações, sempre buscando a melhoria dos resultados. Carlson *et al* (2010) desenvolveram uma arquitetura que é executada continuamente em busca de informações semânticas. Os resultados indicam a potencialidade desse tipo de abordagem.

Um fato que tem tornado cada vez mais vantajoso o uso de métodos semissupervisionados sem fim no aprendizado automático nas diversas análises computacionais é a disponibilidade de muito conteúdo na web. Tradutores automáticos, como o Google Translate[1], têm se tornado um dos melhores na tarefa por se utilizar desse enorme córpus chamado web. A análise discursiva pode se beneficiar amplamente, se souber extrair informações discursivas das páginas web, melhorando muito os resultados da análise discursiva automática (Marcu & Echihabi, 2002; Sporleder & Lascarides, 2008; Chiarcos, 2012).

Nesse trabalho, explorou-se o uso de abordagens de aprendizado semissupervisionado sem fim (do inglês, *semi-supervised never-ending learning* - SSNEL), de forma que não

[1]http://translate.google.com

houve limitação aos dados anotados manualmente. Informações sintáticas, semânticas e até discursivas foram codificadas em atributos para o aprendizado automático buscando superar as lacunas dos trabalhos apresentados anteriormente.

Foram adaptados para o Português dois importantes trabalhos de *parsing* discursivo do Inglês (Soricut & Marcu, 2003; Hernault *et al*, 2010). Soricut & Marcu (2003) introduziram o conceito, a ser definido no decorrer desta tese, de conjunto de dominâncias, e Hernault *et al* (2010) expandiram o trabalho anterior com mais atributos para um aprendizado automático supervisionado, além de realizarem o *parsing* discursivo para todo o texto, diferentemente de Soricut & Marcu (2003), que realizaram a análise individualmente para cada sentença do texto.

Contou-se com a disponibilidade de diversos corpora anotados de acordo com a teoria RST para a realização do aprendizado automático. A maior parte dos textos dos corpora são jornalísticos, extraídos de diversos jornais. Trabalha-se com textos do gênero jornalístico, por estarem amplamente disponíveis em diversas fontes na web, e por serem, geralmente, textos bem escritos. Essa disponibilidade possibilita a exploração de abordagens semissupervisionadas.

Para o aprendizado automático, foram gerados modelos de identificação partindo dos corpora anotado segundo a teoria RST, consistindo no aprendizado supervisionado. No aprendizado semissupervisionado, esses modelos foram utilizados para rotular novos textos, que serviram de entrada para a geração de novos modelos para identificação automática das relações RST. Esse processo de rotular novos textos e criar novos modelos é repetido iterativamente até se obter bons classificadores, segundo algum critério de parada.

A tese desta pesquisa é que a análise discursiva, mais especificamente a identificação das relações retóricas, pode ser beneficiada, atingindo o estado-da-arte para o Português, com o uso de aprendizado supervisionado seguido do uso massivo de dados não rotulados numa abordagem de aprendizado semissupervisionado sem fim. A tese foi confirmada, dado que a abordagem semissupervisionada sem fim obteve melhores resultados no *parsing* discursivo. O conhecimento discursivo obtido inicialmente com dados rotulados, ainda que

em pequena quantidade e desbalanceado, pôde ser generalizado de forma a aumentar a performance na identificação das relações retóricas, que é a principal etapa na análise discursiva.

Hipóteses secundárias também foram confirmadas. A adaptação de modelos do estado-da-arte para o Inglês foram adaptados ao Português e, já nessa etapa, atingiram o estado-da-arte para esta língua. Atributos linguísticos de diversos níveis foram explorados e o subconjunto que produziu os melhores resultados foi selecionado, melhorando ainda mais os resultados obtidos pela adaptação de modelos da língua inglesa. O uso de regras, criadas manualmente, melhoraram a tarefa em casos de relações que são facilmente identificadas por regras, como as relações *Parenthetical* e *Same-unit* (definidas no Apêndice A). Dado que essas relações foram tratadas por regras e não pelos classificadores, diminuiu-se a quantidade de classes no aprendizado automático, aumentando a performance do mesmo. A conjugação entre técnicas de aprendizado automático e regras manuais, numa abordagem híbrida produziu melhores resultados que uma abordagem puramente baseada em aprendizado automático.

Por fim, a metodologia desenvolvida pode ser aplicada a outras línguas que tenham um córpus anotado com alguma teoria discursiva que relacione segmentos com relações discursivas, tal como a língua inglesa, a exemplo de um experimento realizado.

Neste trabalho, além dos objetivos relacionados à tese e hipóteses, objetivou-se a exploração dos conjuntos de dominâncias, definidos por (Soricut & Marcu, 2003). Os atributos obtidos a partir dos conjuntos de dominâncias, foram amplamente utilizados em trabalhos de *parsing* discursivo. Objetivou-se, também, a criação de um *framework* de aprendizado semissupervisionado sem fim, aplicado à análise discursiva.

Como contribuição teórica, o aprendizado semissupervisionado sem fim (SSNEL) foi organizado e um *framework* foi desenvolvido de forma a permitir o aprendizado automático em grande quantidade de dados, a partir dos corpora anotados. Embora a metodologia seja instanciada para a análise discursiva, espera-se que possa ser aplicada a outras tarefas de PLN, dada a grande disponibilidade de textos na web.

Como contribuição prática, trabalha-se na disponibilização de um *parser* discursivo

online com bom desempenho, que possa ser utilizado facilmente. Também será disponibilizada uma versão *offline*, possibilitando total automatização de aplicações que utilizem o conhecimento discursivo.

Embora a análise discursiva seja sujeita a ambiguidades, subjetividades e falta de marcação textual explícita, sua informatividade é de grande interesse em outras tarefas do PLN. Assim, um *parser* discursivo com boa acurácia contribuirá com a automatização da extração do conhecimento discursivo de textos em Português para uso em outras aplicações.

O ineditismo desta pesquisa consiste na forma como o aprendizado semissupervisionado foi conjugado com o aprendizado sem fim e aplicado à análise discursiva. Nessa conjugação, estão relacionadas questões como o controle de mudança de conceito (*concept drift*) e o hibridismo (aprendizado automático e regras desenvolvidas manualmente).

Os resultados comprovaram a potencialidade dessa proposta, visto que a medida-F da classificação das relações retóricas teve ganho, até o momento, de 14,36% (atingindo medida-F de 0,621) e consiste no estado-da-arte para o *parsing* discursivo do Português.

Em linhas gerais, esta tese está organizada como segue. No Capítulo 2, são apresentadas algumas teorias discursivas, com ênfase na RST. No Capítulo 3, os principais trabalhos de *parsing* discursivos automáticos são apresentados, assim como os recursos e ferramentas utilizados nesta pesquisa. No Capítulo 4, importantes conceitos de aprendizado de máquina são introduzidos, com a finalidade de facilitar o entendimento da metodologia empregada. No Capítulo 5, a investigação conduzida sobre segmentação discursiva é apresentada, focando-se nas regras léxico-sintáticas. No Capítulo 6, apresenta-se a metodologia de identificação das relações retóricas e suas nuclearidades. Inicia-se pelo aprendizado supervisionado, com a exploração de diversos atributos, e apresenta-se o aprendizado semissupervisionado sem fim. Nesse capítulo, também são apresentadas regras criadas manualmente para a identificação de duas relações retóricas. Por fim, no Capítulo 7, apresentam-se as conclusões, contribuições teóricas e práticas e algumas limitações, assim como trabalhos futuros. Há, após as referências bibliográficas, o Apêndice A, com a descrição das relações retóricas consideradas e, no Apêndice B, algumas listas

de palavras utilizadas.

2

Teorias Discursivas

Diversos modelos têm sido propostos para a representação e o tratamento do discurso, que pode ser analisado e estruturado em diversas formas:

i) Estrutura linguística (manifestação linguística do discurso, coesão léxica, marcadores discursivos, anáforas);

ii) Estrutura intencional (em que cada segmento discursivo cumpre uma intenção do autor do texto);

iii) Estrutura informacional (identificando como os segmentos relacionam-se uns com os outros, pelas relações discursivas) e;

iv) Estrutura de foco (ou *atencional*, identificando quais entidades são salientes em determinados pontos do discurso).

Com relação à estruturação intencional/informacional, alguns modelos são propostos, como a RST (Mann & Thompson, 1987), que e é o foco deste trabalho. A razão desta escolha é seu frequente e diverso uso em tarefas do PLN. Todavia, a RST não é a única que trata da estruturação intencional/informacional de um texto. Outras pesquisas foram conduzidas por Grosz & Sidner (1986), Jordan (1992), Kamp & Reyle (1993), Lascarides & Asher (1993), Sperber & Wilson (1995), Kehler (2002), Asher & Lascarides (2003), Wolf & Gibson (2005), dentre outros. Esses trabalhos serão brevemente apresentados, com maior foco na RST. Posteriormente, as citadas teorias são apresentadas em ordem cronológica.

2.1 Grosz e Sidner Discourse Theory

Na teoria de Grosz & Sidner (1986), também chamada de GSDT (*Grosz and Sidner Discourse Theory*), o discurso é produzido a fim de satisfazer intenções do escritor ao produzir o texto. Essa teoria busca relacionar as infinitas intenções do autor do texto por meio de relações de satisfação e contribuição entre elas. As autoras definem dois tipos de intenções: as primárias e as subjacentes aos segmentos do discurso e que devem satisfazer as intenções primárias. Define-se, nesta teoria, relações, a saber: *Dominance*, *Satisfaction-Precedence*, *Supports* e *Generates*.

A relação *Dominance* ocorre quando a intenção subjacente a um segmento A contribui para a intenção subjacente de um segmento B. Assim, A *dominates* B ($DOM(A, B)$). A relação *Satisfaction-Precedence* ocorre quando a intenção subjacente a um segmento A deve ser satisfeita antes da intenção subjacente a um segmento B, assim $SP(A, B)$)

Já as relações *Supports* e *Generates* ocupam-se dos conteúdos informacionais dos segmentos. A primeira ocorre se a aceitação de um segmento B fornece subsídios para a aceitação do segmento A ($SUP(B, A)$). A segunda ocorre se a ação descrita em B contribui para a ação descrita em um segmento A ($GEN(B, A)$).

Por exemplo, considere as sentenças abaixo:

1. A teoria RST é bem informativa para muitas tarefas de PLN que necessitam de conhecimento discursivo, e conta com diversos *parsers* disponíveis.

2. Seu uso, portanto, é uma ótima alternativa quando se deseja automatizar totalmente uma tarefa de processamento da linguagem natural.

Pode-se verificar que a intenção do autor do texto é convencer o leitor que o uso da RST é uma ótima alternativa no PLN (segmento 2). Para isso, ele argumenta a favor da RST (segmento 1). Podemos dizer, portanto, que $DOM(2, 1)$ e $SUP(1, 2)$.

Há outras questões abordadas na teoria, como os focos de atenção e a estrutura linguística, que não são tratados aqui dado o interesse específico nas relações intencionais/informacionais. A GSDT diferencia-se da RST por utilizar um conjunto de relações

que não diz qual a intenção do autor do texto, mas que traça relacionamentos entre as intenções. Alem disso, a estrutura hierárquica construída pela RST não é predita na GSDT.

2.2 Rhetorical Structure Theory

A RST (Mann & Thompson, 1987) é uma teoria discursiva que surgiu nos anos 80 e tem sido amplamente aplicada na linguística computacional. Ela busca estruturar a organização do texto pelo uso de relações retóricas entre as suas partes. Cada uma dessas partes tem uma função a desempenhar em relação às outras partes textuais.

A RST tem uma sistemática para a realização da análise de um texto, que consiste nas principais etapas da análise e suas restrições, tais como i) a segmentação e em que consistem os segmentos discursivos, ii) a identificação das relações e as restrições dessas relações, iii) a definição do núcleo e do satélite e iv) a construção da estrutura retórica. A análise discursiva gera uma estrutura semelhante à da Figura 1.3, da página 6.

As relações RST são definidas em termos de i) restrições sobre o núcleo; ii) restrições sobre o satélite; iii) restrições sobre a combinação entre núcleo e satélite; e iv) efeito no receptor do texto.

As Tabelas 2.1 e 2.2 apresentam dois exemplos de definições de relações RST.

Tabela 2.1: **Definição da relação *Antithesis***

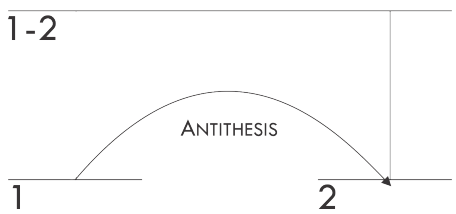
Relação *Antithesis*	
Restrição sobre núcleo (N)	O escritor julga N válido
Restrição sobre o satélite (S)	Nenhuma
Restrição sobre a combinação entre núcleo e satélite	N e S se contrastam e, por esse motivo, não podem ser válidos simultaneamente. Compreendendo-se S inválido levará o leitor a aceitar melhor N
Efeito no receptor	O leitor aceita melhor N

A Figura 2.1 apresenta um exemplo da relação *Antithesis*, em que os segmentos 1 e 2 não podem ser válidos ao mesmo tempo, pois, ou a "detonação"foi "acidental"ou

Tabela 2.2: **Definição da relação *Contrast***

Relação *Contrast*	
Restrição sobre os núcleos (Ns):	Não mais do que dois Ns; as situações nos Ns são (a) compreendidas como similares em vários aspectos, (b) compreendidas como diferentes em vários aspectos e (c) comparadas em relação a uma ou mais dessas diferenças
Efeito no receptor:	O leitor reconhece as similaridades e diferenças resultantes da comparação sendo feita

"proposital". Neste exemplo o segmento nuclear é o segmento 2. Para que a crença do leitor no segmento 2 seja melhor aceita, o segmento 1 deve ser inválido.



1. POUCO DEPOIS DA DETONAÇÃO, OCORRIDA ÀS 10H40 (3H40 EM BRASÍLIA), FONTES DA POLÍCIA MOSCOVITA ADIANTARAM QUE ELA TERIA ACONTECIDO PROVAVELMENTE POR CAUSA DA EXPLOSÃO ACIDENTAL DE UM BUJÃO DE GÁS.
2. - ISTO NÃO FOI UM ACIDENTE, FOI PROPOSITAL - DISSE RESIN, CITADO PELA AGÊNCIA OFICIAL RUSSA "ITAR-TASS".

Figura 2.1: **Exemplo da relação *Antithesis***

Um segmento discursivo, chamado de UBD (unidade básica do discurso, ou *elementary discourse unit*, em Inglês), é a menor unidade para a composição da árvore discursiva. Nas UBDs encerram-se as ideias básicas, ou proposições do texto. Essas unidades podem corresponder às orações ou sentenças do texto, ou ainda a unidade menores. Em (Carlson & Marcu, 2001) são definidas regras para a delimitação das UBDs de textos.

Na RST, os segmentos são classificados de acordo com a importância que desempenham

na relação estabelecida entre os mesmos. Um segmento pode ser núcleo se for considerado mais importante, enquanto que o satélite contribui com o núcleo e é secundário na relação. Dessa forma, em uma relação em que os dois segmentos são igualmente importantes, tem-se uma relação multinuclear, caso contrário, a relação tem um segmento mais importante que o outro e caracteriza uma relação mononuclear. A Figura 2.1 apresentou um exemplo de relação mononuclear (*Antithesis*) e, na Figura 2.2, apresenta-se um exemplo de relação multinuclear (*Contrast*). Por restrições da RST, as relações são sempre estabelecidas entre segmentos adjacentes no texto.

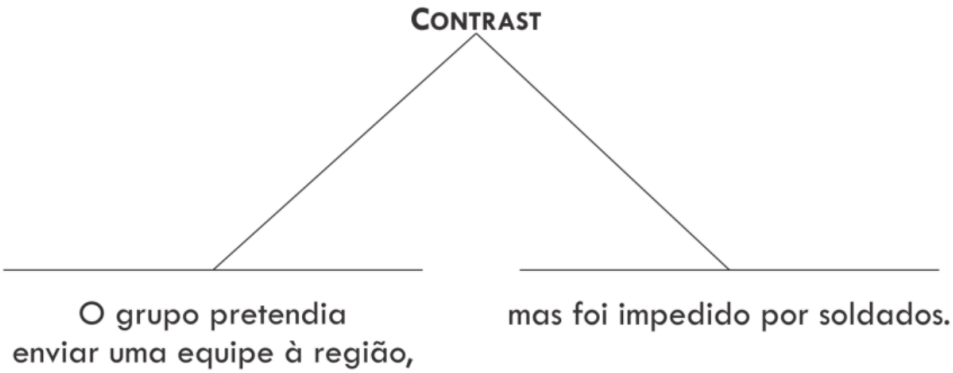


Figura 2.2: **Exemplo da relação *Contrast***

A definição da nuclearidade é dependente da relação estabelecida entre dois segmentos. Isso torna-se um desafio para a identificação automática, pois depende da interpretação do analista, e é, portanto, algo subjetivo e ambíguo. A Figura 2.3 mostra um exemplo de uma sentença segmentada em duas UBDs que podem ser relacionadas por duas relações RST, dependendo do que se quer enfatizar, uma causa ou um resultado, respectivamente *Volitional-Cause* ou *Volitional-Result*. Caso a primeira relação seja escolhida, tem-se que o segmento 2 é o núcleo, caso contrário, o segmento 1 será nuclear.

Não há um conjunto fechado de relações. O rol original é composto por 24 relações, apresentadas na Tabela 2.3. Outros trabalhos estenderam e outros agruparam as relações. O conjunto de relações utilizadas neste trabalho é apresentado no Apêndice A. Esse

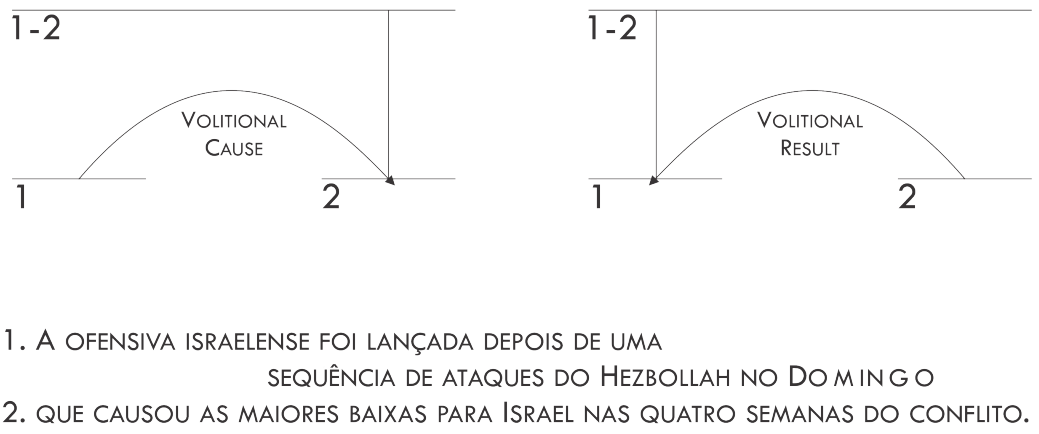


Figura 2.3: **Duas possíveis escolhas de nuclearidade para um mesmo par de segmentos, dependendo do que se quer enfatizar, uma causa (*Volitional-Cause*) ou um resultado (*Volitional-Result*)**

conjunto de relações foi escolhido por conter todas as relações presentes nos corpora RST empregados neste trabalho.

Um grande desafio encontrado na análise RST é a definição da relação retórica entre dois segmentos textuais. Se esse contexto for expandido para um texto inteiro, há diversas possíveis árvores discursivas para um mesmo texto, com segmentos, relações e nuclearidades diferentes. Por exemplo, um analista RST pode identificar que há uma oposição entre duas UBDs e, assim, relações como *Antithesis* e *Concession* poderiam ser úteis na análise, gerando diferentes árvores discursivas.

Além disso, as relações RST podem ser semânticas ou retóricas (intencionais). As relações semânticas se ocupam mais do relacionamento entre os conteúdos das UBDs, enquanto que as retóricas se ocupam das intenções do autor do texto. Assim, é possível que haja duas possíveis relações RST para um mesmo par de segmentos, uma semântica e outra retórica. Dentre as relações apresentadas na Tabela 2.3, as seguintes são intencionais: *Antithesis*, *Background*, *Concession*, *Enablement*, *Evidence*, *Justify* e *Motivation*.

A ambiguidade é um grande desafio da análise RST, seja por foco do analista (como no exemplo da Figura 2.3), seja pela escolha de relação semântica ou retórica, em alguns possíveis casos.

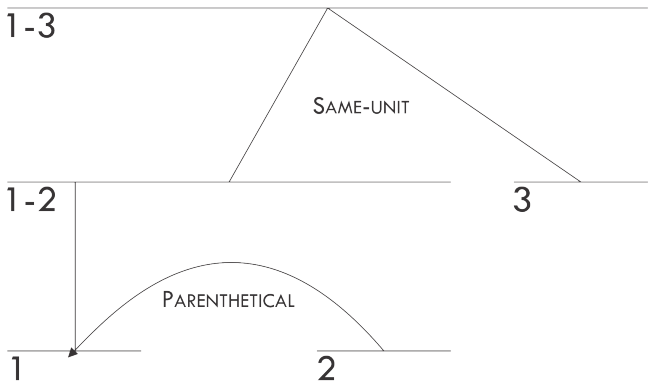
Tabela 2.3: **Conjunto original de relações propostas na RST**

Relações
Antithesis
Background
Circumstance
Concession
Condition
Contrast
Elaboration
Enablement
Evaluation
Evidence
Interpretation
Joint
Justify
Motivation
Non-Volitional-Cause
Non-Volitional-Result
Otherwise
Purpose
Restatement
Sequence
Solutionhood
Summary
Volitional-Cause
Volitional-Result

Alguns autores, como Marcu (1997), adicionaram relações consideradas estruturais, que não relacionam o conteúdo das UBDs e nem identificam as intenções, mas que servem para estruturar melhor a árvore retórica. Como relações estruturais, estão as relações *Parenthetical* e *Same-unit*.

A relação *Parenthetical* serve para separar textos com informações adicionais que estejam bem marcados, como entre parênteses ou colchetes, por exemplo. Já a relação *Same-unit* foi inserida para suprir a necessidade de relacionar dois segmentos que não estão adjacentes no texto, mas formam uma mesma UBD, separada, por exemplo, por um texto que caracterize a relação *Parenthetical*.

Na Figura 2.4, o texto é segmentado em três segmentos, sendo um deles, o segmento 2, um texto entre parênteses "(ONU)". O segmento 2 realiza a quebra da UBD formada



1. O LÍBANO APRESENTOU UM PEDIDO FORMAL AO CONSELHO DE SEGURANÇA DA ORGANIZAÇÃO DAS NAÇÕES UNIDAS
2. (ONU)
3. DE UMA REVISÃO DA PROPOSTA DE RESOLUÇÃO PARA O FIM DO CONFLITO NO ORIENTE MÉDIO.

Figura 2.4: **Exemplo das relações estruturais *Parenthetical* e *Same-unit***

por toda a sentença (veja que o segmento 3 não contém um verbo). Assim, a relação *Same-unit* é utilizada para garantir que a UBD seja toda ligada em um mesmo *span*.

Convenciona-se relacionar, primeiramente, as UBDs de uma mesma sentença. Posteriormente, as sentenças de um mesmo parágrafo e, por fim, os parágrafos até formar uma estrutura completa do texto.

É importante salientar que uma UBD deve consistir numa proposição completa, assim, costuma-se considerar a frase (o verbo e seus argumentos) como uma proposição completa. Já o segmento formado pelo texto "(ONU)", entre parênteses, é segmentado com vistas a tornar a análise discursiva mais informacional.

2.3 Relações semânticas de Jordan

Jordan (1992) propõe um conjunto de relações semânticas que se definem entre o conteúdo de quaisquer duas partes do texto, desvinculadas das intenções do escritor. Esta é a principal diferença para com a RST, que busca capturar as intenções do autor do texto, expressas no campo "efeito"das definições. Outra diferença é a não definição da ideia de

nuclearidade entre os segmentos envolvidos em uma relação.

As 69 relações de Jordan são agrupadas em alguns tipos de relações, a saber, *Detail*, *General*, *Logical*, *Modal*, *Time*, *Text Manipulation*, *Special* e *Other*. As relações de Jordan são uma compilação de relações propostas em diversos outros trabalhos e são exibidas na Tabela 2.4.

Tabela 2.4: **Lista das relações semânticas de Jordan.**

Relação	Grupo
Identification	*Detail*
Classification	
Specification	
Appearance	
Characteristics	
Function	
Material	
Parts	
Active	*General*
Passive	
Agent	
Source	
Assessment	*Logical*
Basis	
Cause	
Effect	
Emotive Effect	
Purpose	
Means	
Problem	
Solution	

Tabela 2.4: **Lista das relações semânticas de Jordan.**

Relação	Grupo
Possibility	
Capability	
Correctness	
Propriety	
Necessity	
Need	
Completion	
Achievement	
Future	*Modal*
Intention	
Mandate	
Authority	
Determination	
Permission	
Obligation	
Willingness	
Desire	
Before	
After	*Time*
Simultaneous	
Inverted time	

Tabela 2.4: **Lista das relações semânticas de Jordan.**

Relação	Grupo
Elaboration	
Summary	
Repetition	
Paraphrase	*Text manipulation*
Forecast	
Transition	
Collateral inversion	
Concession	
Compatibility	
Contrast	
Comparison	*Special*
Conditionals	
Document structures	
Hypothetical-Real	
Transition couplets	
Accompaniment	
Circumstance	
Inverted circumstance	
Connection	
Enablement	
Example	*Other*
Extent	
Location	
Inverted Location	
Manner	
True	

Tabela 2.4: **Lista das relações semânticas de Jordan.**

Relação	Grupo

2.4 (Segmented) Discourse Representation Theory

Kamp & Reyle (1993) e Lascarides & Asher (1993) propuseram a *Discourse Representation Theory* (DRT). Posteriormente, Asher & Lascarides (2003) propuseram sua derivação, chamada de *Segmented Discourse Representation Theory* (SDRT). A DRT baseia-se nas semânticas dinâmicas, em que se vê o significado de um discurso como uma função entre seus contextos e não se pode definir essa função considerando apenas o conteúdo textual. A semântica dinâmica é restrita a um conjunto de fenômenos pragmáticos, principalmente focada nos efeitos da estrutura lógica sobre as anáforas. Imagine os seguintes segmentos:

1. O homem entrou na loja.

2. Ele pediu um computador.

A DRT resolveria a anáfora presente nos segmentos com a seguinte notação: seja x e y, $entrou(x)$, $pediu(x, y)$. A resolução seria que $x = homem$ e $y = computador$.

A DRT, no entanto, falha em estruturar o discurso com relações retóricas. Por isso a SDRT incorpora as relações retóricas na DRT, evidenciando a importância das relações retóricas para o entendimento do discurso. Essa teoria busca modelar as interações entre o conteúdo semântico (DRT) dos segmentos e o global, tendo-se a estrutura discursiva do texto.

A SDRT é muito semelhante a RST, mas a SDRT se preocupa mais em explicar uma série de fenômenos, como anáforas, verbos elípticos, implicações e pressuposições. Além disso, a SDRT têm relações para o tratamento do diálogo, tais como *Question Elaboration*, *Correction* e *Question Answer Pair*.

2.5 Relevance Theory

Sperber & Wilson (1995) notaram que as pessoas envolvidas em uma comunicação, como na produção de um texto, preocupam-se com a noção de relevância do que está sendo dito. Isso leva a que o autor se preocupe em dizer o que é relevante e será o mais sucinto possível em comunicar isso.

A teoria baseia-se em dois princípios: i) princípio comunicativo da relevância, em que o ato de escrever (ou falar) uma proposição é digno de ser lido (ou ouvido), o que levará ao ii) princípio cognitivo no leitor, que o fará processar o conteúdo. Desta forma o escritor fornece pistas para suas intenções e o leitor faz a inferência das pistas de acordo com o contexto das informações.

Na *Relevance Theory* (RT), as proposições textuais podem estar em relação de relevância entre si, partindo da premissa de que uma proposição tem relevância sobre outra, em um texto. Essas relações são derivadas parcialmente do contexto, assim, quando uma nova proposição é gerada, esta é interpretada como relevante à proposição anterior já produzida e ao contexto geral.

Nem a RT nem a GSDT definem o tipo de relação retórica, mas as deixam para serem inferidas. Essas teorias definem um conjunto genérico de relações, ora para explicitar a dominância entre elas, ora para definir a relevância entre as mesmas.

2.6 Relações semânticas de Kehler

Kehler (2002) define um rol de apenas três relações semânticas: *Resemblance*, *Cause-Efect* e *Contiguity*. Segundo o autor, todas as relações discursivas poderiam ser classificadas como uma dessas três.

A relação *Resemblance* ocorre quando há algo em comum e/ou em contraste nos segmentos discursivos. A relação *Cause-Efect* ocorre quando há alguma implicação, direta e indireta, entre algo nos segmentos relacionados. Já para a ocorrência de *Contiguity*, é necessária uma sequência de eventos nas proposições relacionadas.

Trabalhos como esses servem como base para uma agrupamento hierárquico das rela-

ções retóricas da RST. Por exemplo, todas as relações de causa-efeito da RST (*Volitional-Result*, *Volitional-Cause*, *Non-volitional-Result*, *Non-volitional-Cause*ß) podem ser agrupadas, inicialmente, pela relação semântica *Cause-Efect* de Kehler. A relação RST *Sequence*, da RST é a relação *Contiguity*, de Kehler. Como exemplos de relação do grupo *Resemblance*, podemos ter *Elaboration*, *List*, *Contrast* e *Restatement*.

As relações de Kehler não levam em consideração a nuclearidade dos segmentos em relação, como a RST, nem pressupõem a existência de uma estrutura discursiva que considere todo o texto.

2.7 Reestruturação de Wolf e Gibson

Embora possa não ser considerada uma teoria discursiva, Wolf & Gibson (2005) continuam a linha relacional de estruturação discursiva, como a RST, mas argumentam que a estruturação discursiva de um texto em formato arbóreo não é a forma mais adequada. Os autores propõem que a estrutura discursiva seja em formato de grafo, sem as restrições do formato arbóreo. Para isso, os autores procederam a uma anotação de córpus mostrando que os anotadores encontraram relações não possíveis em uma estrutura arbórea, como um nó que participa de mais de uma relação. Um exemplo de estrutura discursiva em forma de grafo (não arbóreo) é dado na Figura 2.5 (Wolf & Gibson (2005) - p. 266) traduzida para o Português. No exemplo, além das relações serem diferentes das da RST (como a relação *Similarity*), há relações entre segmentos não adjacentes, como entre os segmentos 2 e 4. Neste caso, o segmento 4 traz detalhes sobre o "manjericão"do segmento 2.

2.8 Outros modelos e representações discursivas

Além dos trabalhos descritos anteriormente, há diversos outros que buscam tratar diversos fenômenos discursivos. A *Centering Theory* (Brennan *et al*, 1987) trata das relações existentes entre anáforas, propondo que a coerência discursiva objetiva dirigir o foco de atenção para a escolha de uma expressão referencial e para a coerência dos enunciados,

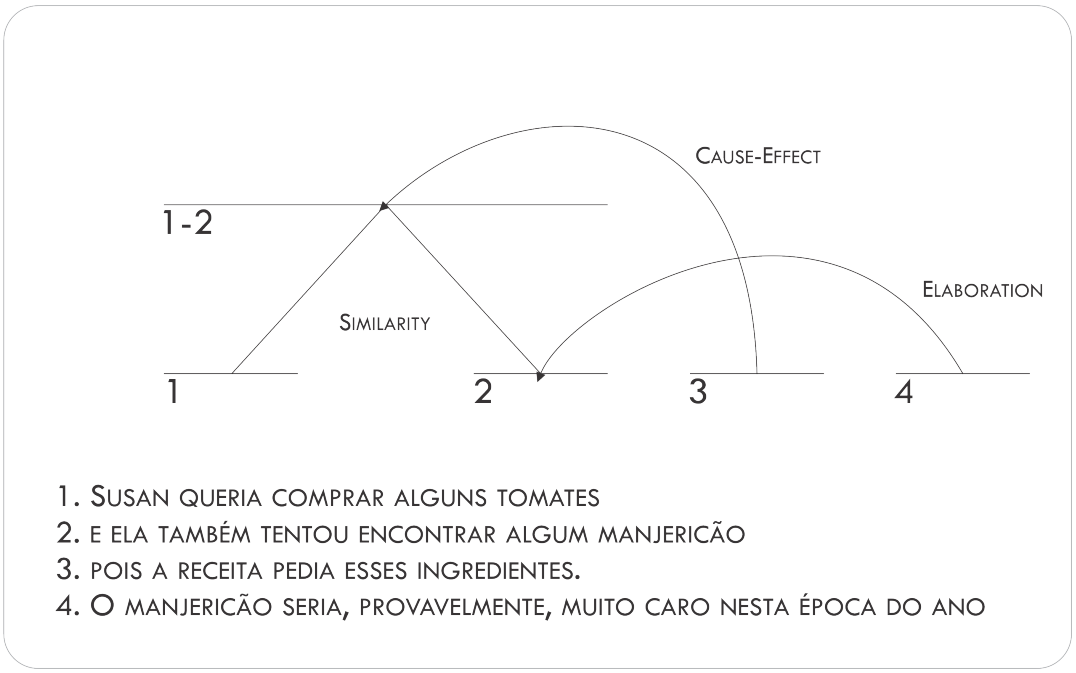


Figura 2.5: **Exemplo de estruturação discursiva não arbórea (Wolf & Gibson (2005) - p. 266)**

dentro dos segmentos discursivos. A teoria busca predizer qual entidade do discurso é mais saliente em determinados segmentos, definindo um conjunto de regras e restrições que governam as escolhas feitas pelos participantes do discurso, como a escolha da estrutura sintática e as expressões referenciais.

Ainda tratando da resolução anafórica, tem-se a Teoria das Veias (Cristea *et al*, 1998). Nessa teoria, relaciona-se a estrutura retórico-discursiva com as cadeias de co-referência em um texto. Assim, dado um elemento anafórico, define-se na Teoria das Veias o domínio de acessibilidade referencial, ou seja, o domínio de ocorrência de elementos que sejam co-referentes ao elemento anafórico. Para a definição do domínio referencial, a árvore discursiva (como a gerada pela RST) é utilizada, em especial a noção de nuclearidade dos segmentos.

Aumentando o escopo da análise discursiva para considerar um grupo de textos, tem-se a *Cross-document Structure Theory* (Radev, 2000) que propõe um conjunto de relações, similares às da RST, para relacionar segmentos de textos diferentes. Essa teoria foi proposta inicialmente para a fusão/sumarização de informações de textos que tratam de

um mesmo assunto. O resultado da análise consiste em um grafo direcionado (pois as relações têm direção) e pode cobrir apenas algumas porções dos textos, gerando um grafo desconexo.

Dentre as teorias apresentadas, a RST é a mais amplamente utilizada no PLN, tanto por sua informatividade, quanto por sua possível automatização, a despeito dos desafios inerentes da análise discursiva. Nesta pesquisa, explorou-se a etapa de identificação das relações discursivas, considerando o rol de relações do Apêndice A.

3 Trabalhos Correlatos

Neste capítulo, são apresentados os trabalhos de *parsing* discursivo. Primeiro, foca-se na identificação das relações retóricas e suas nuclearidades. Posteriormente, são apresentados trabalhos de segmentação discursiva. Por fim, apresentam-se as ferramentas e os recursos da área, utilizados neste trabalho.

3.1 Identificação das relações discursivas

Os trabalhos de análise discursiva automática estão classificados nesta seção por tipo de abordagem:

i) uso de padrões lexicais, obtidos de forma manual, contendo principalmente informações léxicas, como marcadores discursivos, acrescidas de informações morfossintáticas;

ii) uso de aprendizado automático, tanto supervisionado quanto semissupervisionado, gerando classificadores que realizam as diversas etapas da análise discursiva.

Ao final, apresenta-se o agrupamento dos trabalhos, segundo a abordagem utilizada, com conclusões sobre a relação entre a literatura e a presente pesquisa.

3.1.1 Uso de padrões lexicais

Nesta abordagem, apresentam-se os trabalhos em ordem cronológica, a partir do primeiro *parser* discursivo para o Inglês (Marcu, 1997). São apresentados trabalhos para o Inglês, Alemão, Português e Espanhol.

Marcu (1997, 2000) desenvolveu o primeiro *parser* discursivo baseado na RST. Tanto para a identificação dos segmentos discursivos quanto para a identificação das relações retóricas presentes no texto, o autor utiliza padrões lexicais obtidos por análise de córpus. O autor baseou-se em marcadores discursivos obtidos de trabalhos anteriores, como de Grosz & Sidner (1986), para criar um conjunto inicial de mais de 450 marcadores. Um córpus foi processado para extrair exemplos da aplicação de cada marcador, com um contexto de 200 palavras próximas. Esses exemplos foram analisados por especialistas em análise discursiva a fim de verificar oito itens: 1) cenário lexical da aplicação do marcador, incluindo pontuações; 2) tipo de uso: sentencial, discursivo ou ambos; 3) posição do marcador no segmento; 4) o limite direito do segmento em que se encontrou o marcador; 5) posição relativa da unidade textual relacionada com o marcador (antes ou depois); 6) as relações discursivas sinalizadas pelo marcador; 7) tipo das unidades textuais conectadas pelo marcador (oração, sentença ou parágrafo); e 8) a nuclearidade das unidades textuais conectadas pelo marcador.

Para cada padrão, foi gerada uma expressão regular que representa o marcador. Também foi gerado um procedimento para a segmentação textual, baseado nos padrões, e um procedimento para a atribuição das relações retóricas e suas nuclearidades aos segmentos identificados.

O procedimento para identificação da árvore discursiva de um texto é o seguinte conjunto de passos: 1) determinar um conjunto D de marcadores discursivos e conjunto U_T de UBDs no texto T; 2) determinar um conjunto R de relações entre os segmentos em U_T; 3) usar um procedimento para determinar todas as árvores discursivas de T; e 4) associar uma probabilidade de ocorrência a cada árvore, para possibilitar a escolha de uma delas.

O autor reporta cobertura de 0,808 e precisão de 0,895 na identificação correta de marcadores discursivos que indicam uma relação retórica. Com relação à segmentação discursiva, a cobertura foi de 0,813 e a precisão de 0,903.

Visto que esse trabalho foi desenvolvido antes do primeiro córpus RST para o Inglês ter sido compilado, dois especialistas anotaram cinco textos e as árvores discursivas geradas manualmente foram utilizadas na avaliação final do *parser*. É interessante notar que, para

o cálculo da concordância entre os anotadores, o coeficiente de *Spearman* (Spearman, 1987) foi utilizado e levou a um resultado de 0,798, com um nível de confiança $p < 0,0001$. As diferenças entre as anotações foram, principalmente, devido ao foco dado por cada anotador: ora para a estruturação dos parágrafos, ora para a organização lógica do texto. Considerando a estrutura gerada automaticamente pelo *parser* desenvolvido, esse coeficiente foi de 0,449 (com $p < 0,0001$), quando comparada à estrutura gerada por cada um dos anotadores.

Baseado na abordagem de Marcu, Corston-Oliver (1998) desenvolveu um *parser* discursivo que utilizou tanto padrões léxicos quanto estruturas sintáticas lexicalizadas para encontrar as relações retóricas do texto. O córpus utilizado foi do domínio científico, mais especificamente, textos enciclopédicos. Além de utilizar a formalização de padrões discursivos de Marcu (1997, 2000), o autor utilizou informações das estruturas sintáticas lexicalizadas para a identificação das relações retóricas da seguinte forma: i) se os segmentos textuais têm suas correspondentes UBDs subordinadas sintaticamente uns aos outros; ii) se os segmentos têm suas UBDs em voz ativa ou passiva; iii) se os segmentos têm fenômenos como anáforas e elipses; iv) se os núcleos dos constituintes sintáticos são os mesmos ou não; e v) consideraram as classes gramaticais dos segmentos.

Já Schilder (2002) cria, em uma primeira etapa, uma estrutura retórica parcial com o uso de marcadores discursivos. Essa estrutura conterá apenas o que pode ser identificado pelos marcadores. Em outra etapa, completa essa estrutura parcial com o uso técnicas da área de Recuperação de Informação (RI). Além do conhecimento discursivo, o autor utiliza também a topicalidade e posição dos segmentos no texto. Por exemplo, a fim de verificar a importância de um segmento, este é transformado em um vetor e comparado ao vetor que contém o título do documento. Quanto mais próximo do título, o segmento ocupará posição mais importante na estrutura retórica. O sistema desenvolvido realiza a sumarização automática de textos da web com o uso dos atributos utilizados na criação da estrutura retórica.

O autor não realizou uma avaliação intrínseca. Na tarefa de sumarização automática, seu sistema foi avaliado e indicou melhorias em relação a outros sistemas de sumarização

automática.

Para o Português, Pardo (2005) desenvolveu uma metodologia de análise discursiva baseada fortemente na RST (que deu origem ao *parser* DiZer), que utiliza principalmente padrões para a definição da estrutura RST de um texto. Nesses padrões, há marcas textuais (marcadores discursivos e outras frases indicativas) que indicam a relação entre duas porções textuais e a atribuição de núcleo/satélite, dando a importância de cada segmento na relação. Como os padrões (mais de 740) foram extraídos de um córpus de textos científicos, sua aplicação a outro domínio textual leva a um desempenho menor. Além disso, a forma como foi implementado faz com que a ferramenta seja muito dependente da plataforma de desenvolvimento. O DiZer teve medida-F de 0,625 na detecção das relações em textos do domínio acadêmico. Quando avaliado em textos jornalísticos, obteve medida-F de 0,403.

Essas razões motivaram o desenvolvimento do DiZer 2,0 (Maziero *et al*, 2011), que implementa a metodologia do DiZer, utilizando padrões para criação da estrutura RST, mas tem uma interface com os usuários da ferramenta que permite a criação de novos padrões para a análise. Como é possível criar novos padrões de análise discursiva, a metodologia pode ser utilizada para outras línguas (além do Português) e domínios textuais.

Na Figura 3.1, há uma ilustração simplificada da utilização de um padrão na identificação de uma relação discursiva entre duas porções textuais. Nesse exemplo, a expressão "e o resultado foi"na segunda oração da sentença indica a relação *Non Volitional Result* entre as unidades, indicando que a sentença S2 é um resultado não volitivo da sentença S1. A direção da seta indica que o segmento mais importante na relação é o segmento S1, considerado então como núcleo.

A grande limitação no bom desempenho do DiZer 2.0, e dos outros trabalhos acima que utilizam-se apenas de padrões lexicais, é a necessidade de especialistas da área do discurso realizarem a análise de córpus a fim de criarem os padrões a serem utilizados na análise automática. Embora os marcadores discursivos presentes nos textos sejam o recurso mais utilizado na análise discursiva, diversos trabalhos para a língua inglesa utilizam outros conhecimentos e recursos, inclusive de paradigma não supervisionado, não necessitando

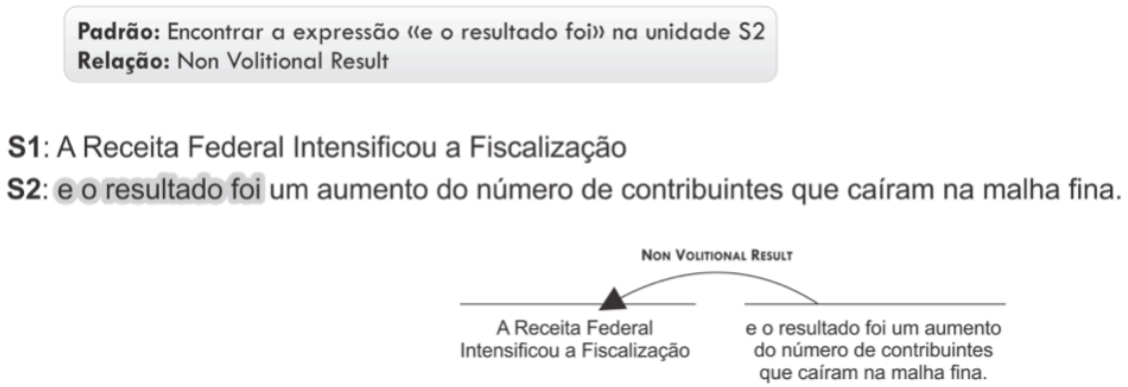


Figura 3.1: **Exemplo simplificado de aplicação de padrão lexical no DiZer**

de um especialista da área para melhorar o repositório de conhecimentos para realização da análise.

Cunha *et al* (2012) realizam a identificação das nuclearidades e relações retóricas no nível intrassentencial para o Espanhol. A exemplo do DiZer, baseado no córpus RST para a referida língua, utilizam um conjunto de padrões lexicais e sintáticos. O córpus RST *Spanish Treebank* (Cunha *et al*, 2011) é o único anotado com RST para o Espanhol e contém textos de nove domínios específicos, por exemplo, Astrofísica, Engenharia de Terremotos e Economia. Esse córpus conta com uma porção (31%) que foi anotada por dois especialistas em RST e essa porção foi utilizada para os testes da metodologia.

A metodologia de extração dos padrões léxico-sintáticos foi baseada na análise do córpus RST. Para cada relação ocorrente no córpus, foram observados todos os possíveis marcadores léxicos ou sintáticos da relação RST. Esses marcadores foram categorizados em três tipos: marcadores discursivos tradicionais (como *segundo*, *desde que*, *assim como*); marcadores incluindo unidades lexicais (especificamente substantivos e verbos); e marcadores incluindo estruturas verbais (por exemplo *para + verbo no infinitivo*).

Sobre a relação *Elaboration*, nenhum padrão foi extraído, pois, na não identificação de alguma outra relação, o sistema atribui a relação *Elaboration*, que é a mais abundante no córpus. Vale notar que esse é um procedimento padrão na maior parte dos *parsers* discursivos. Após a análise de córpus, 778 padrões foram extraídos das 2056 relações presentes. As relações que contêm maior número de padrões extraídos foram *Purpose*, *Antithesis* e *Sequence*, mesmo que os números de ocorrências dessas relações no córpus

não tenham sido os maiores.

O sistema de análise discursiva desenvolvido pelos autores realiza os seguintes passos: i) segmentação sentencial; ii) segmentação discursiva; e iii) dados os segmentos discursivos (UBDs), realiza a aplicação dos padrões discursivos em quatro etapas, a saber, a) marcadores discursivos tradicionais; b) marcadores discursivos incluindo unidades lexicais; c) marcadores com estruturas verbais; e d) aplicação de *Elaboration* quando nenhuma relação foi encontrada. Após a identificação das relações, outras regras são aplicadas para a atribuição da nuclearidade: i) regras núcleo-satélite; ii) regras satélite-núcleo; e iii) regras multinucleares.

Nesse trabalho, o tratamento da ambiguidade inerente aos marcadores discursivos pode ser realizado seguindo três estratégias: i) escolha da relação com maior número do marcador ambíguo; ii) retornar todas as possíveis relações do marcador; e iii) desenvolvimento de estratégia mais elaborada para a escolha de uma das possíveis relações. Os autores optaram pela primeira estratégia para uso no sistema.

A avaliação do sistema foi realizada aplicando os padrões ao córpus de teste. Na etapa de identificação das relações discursivas e suas nuclearidades, obteve-se a medida-F de 0,817. Esse resultado é satisfatório e fica muito acima dos resultados obtidos para o Português. Possivelmente, o bom resultado reportado pelos autores é devido ao cenário de aplicação, composto por textos especializados de domínios bem definidos, o que não ocorre com textos mais gerais, como os jornalísticos.

3.1.2 Uso de Aprendizado de Máquina

Nesta subseção, apresentam-se diversos trabalhos que utilizaram técnicas de aprendizado automático para a identificação das relações retóricas.

3.1.2.1 Modelo probabilístico

Soricut & Marcu (2003), para determinação das relações retóricas, treinam um modelo probabilístico baseados em informação sintática lexicalizada. Esse *parser* (SPADE - *Sentence-Level parsing for DiscoursE*) utiliza informações sintáticas advindas do *parser*

sintático de Charniak (2000) e usa um algoritmo *bottom-up* com programação dinâmica a fim de obter a estrutura retórica de um texto. O córpus utilizado para o aprendizado foi o RST-DT (Carlson *et al*, 2001), que contém 347 documentos para treinamento e 38 para teste, amplamente utilizado nos trabalhos de análise discursiva para o Inglês.

Uma árvore sintática lexicalizada contém, nos nós interiores, a indicação de palavras junto às etiquetas sintáticas. A lexicalização é realizada fazendo-se a projeção, a partir das folhas, das palavras nucleares presentes em cada ramificação da árvore. Veja o exemplo de árvore lexicalizada na Figura 3.2, extraída de Soricut & Marcu (2003), em que os nós internos contêm palavras da sentença. No exemplo, são apresentadas apenas as palavras mais informativas para a análise.

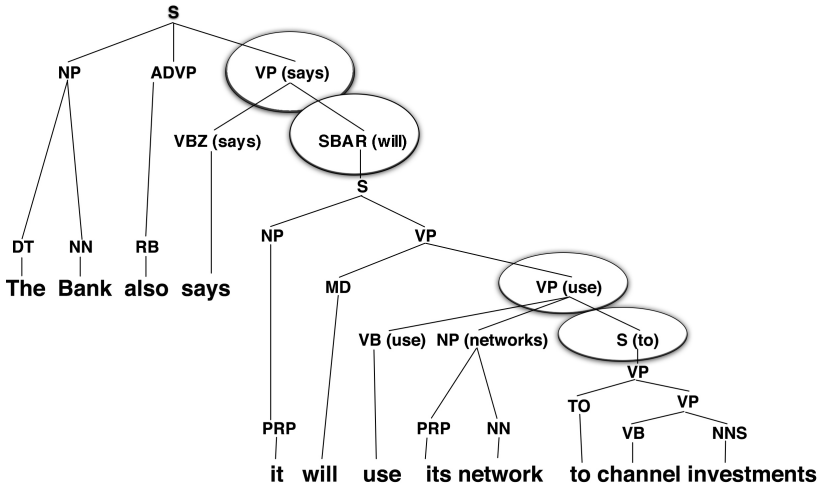


Figura 3.2: **Exemplo de árvore sintática lexicalizada. Os círculos indicam os nós internos mais informativos para realização da análise discursiva. Exemplo extraído de Soricut & Marcu (2003).**

O modelo é constituído por um conjunto de probabilidades que tratam das relações retóricas entre dois segmentos e probabilidades que tratam da estruturação da árvore discursiva intrassentencial dos respectivos segmentos.

Em cada elemento do conjunto de probabilidades Pr (Equação 3.1, que define a relação discursiva (R) mais provável), tem-se as etiquetas sintáticas (ST_i - *syntax tag*) e as respectivas palavras (LH_1 - *lexical head*) dos segmentos $S1$ e $S2$ no argumento θ_1, da

Equação 3.3.

$$Pr(R|\theta_1) = pr \tag{3.1}$$

Em cada elemento do conjunto Ps (Equação 3.2, que define a estrutura (S) mais provável), tem-se, para os mesmos segmentos $S1$ e $S2$, informações codificadas no argumento θ_2, da seguinte forma:

$$Ps(S|\theta_2) = ps \tag{3.2}$$

Os argumentos θ_1 e θ_2, utilizados por Pr e Ps, estão esquematizados na Equação 3.3 e na Equação 3.4.

$$\theta_1 = (S2, LH_2, ST_2) \prec (S1, LH_1, ST_1) \tag{3.3}$$

$$\theta_2 = (S2, ST_2) \prec (S1, ST_1) \tag{3.4}$$

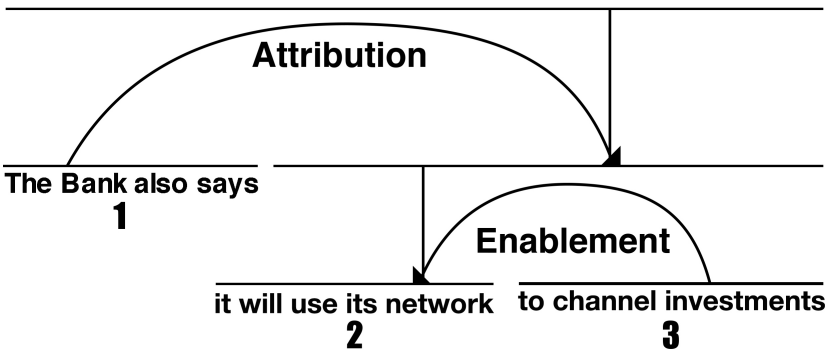


Figura 3.3: **Exemplo de árvore discursiva. Exemplo extraído de Soricut & Marcu (2003)**

Considerando o exemplo da Figura 3.3, Pr (Equação 3.1) poderia ser definida como $Pr(Enablement|\theta_1)$ e $\theta_1 = (S2, to, S) \prec (S1, use, VP)$. Para a definição de Ps, considera-se apenas as ST_i, portanto, sem lexicalização: $Ps(S|\theta_2)$ e $\theta_2 = (S2, S) \prec (S1, VP)$. Os

valores de Pr e Ps são gerados após a contabilização de suas ocorrências em todo o córpus de treinamento.

O símbolo $\prec$ indica a *dominância* entre as informações léxico-sintáticas da Figura 3.2. Essas informações são chamadas pelos autores de conjuntos de dominâncias e consistem nas informações léxico-sintática mais indicativas para a identificação das relações retóricas. Essas informações também são utilizadas na etapa de segmentação discursiva.

Tendo gerado todas as probabilidades com um conjunto de treinamento, o modelo é aplicado ao conjunto de teste da seguinte forma, para uma instância formada pelas sentenças $S1$ e $S2$:

1. Encontram-se as probabilidades em Pr;
2. Encontram-se as probabilidades em Ps;
3. Encontra-se a relação pela Equação 3.5, em que $P(R|\theta)$ é definida na Equação 3.6:

$$R = argmax P(R|\theta) \tag{3.5}$$

$$P(R|\theta) = \prod Pr(R|\theta_1) \times Ps(S|\theta_2) \tag{3.6}$$

A metodologia obteve medida-F de 0,49 na identificação de um conjunto de 18 relações (agrupamento das relações RST segundo suas similaridades), utilizando segmentação e análise sintática automáticas. Quando utilizadas segmentação e análise sintática manual, a metodologia atingiu 0,75, bem próximo da performance humana de 0,77. A perfomance humana foi obtida pela medição da concordância entre dois anotadores humanos em uma parte do córpus utilizado (Carlson *et al*, 2001). A metodologia é baseada na estrutura sintática lexicalizada e, portanto, é aplicável apenas ao nível intrassentencial, não realizando a análise discursiva do texto inteiro.

3.1.2.2 Aprendizado supervisionado

Reitter (2003) utiliza a técnica SVM - *Support Vector Machine* (Vapnik, 1995) para a

identificação das relações retóricas. O autor cria um classificador para cada relação a ser identificada (classificadores binários). Cada classificador diz que se o par de segmentos está ou não relacionado por determinada relação. Essa predição tem uma probabilidade e a de maior probabilidade é então escolhida.

Os autores partiram da hipótese de que a linguagem natural combina várias pistas textuais para a definição da retórica do texto. Dentre os atributos utilizados, estão marcadores discursivos e pronomes, anáforas, pontuações, etiquetas morfossintáticas, similaridade lexical entre os segmentos e tamanho dos mesmos. Atributos (profundos) do nível semântico não foram utilizados dado que, na época, era algo não disponível. Foram utilizados os classificadores binários para a identificação das relações retóricas no texto. Posteriormente, todas as possíveis árvores são construídas e a mais provável é escolhida, considerando-se as probabilidades extraídas do córpus utilizado no treinamento.

Esse foi um dos primeiros trabalhos de *parsing* discursivo a utilizar técnicas de aprendizado de máquina e salientou a importância da anotação de córpus em formato legível ao computador. O sistema criado obteve acurácia de 0,618 para o Inglês. A mesma abordagem foi aplicada a um córpus em Alemão e obteve desempenho de 0,391 na identificação das relações retóricas.

Mahmud & Ramsay (2005) fazem a análise discursiva automática de textos de redações produzidas por alunos do ensino médio, assumindo que tais textos podem não ser de boa qualidade, com diversos erros ortográficos e gramaticais. O objetivo dos autores foi fornecer uma ferramenta que auxiliasse estudantes a corrigirem suas redações com informações além da ortografia e sintaxe. Para isso, tiveram de realizar a análise discursiva dos textos, focando nas relações retóricas entre as proposições textuais.

A segmentação do texto foi realizada apenas no nível sentencial, não identificando as UBDs. Além disso, os autores consideraram que não seria possível extrair informações suficientes para a identificação de um grande número de relações retóricas. Portanto, foi considerado um pequeno conjunto de relações retóricas: *Narrative*, *Elaboration*, *Contrast*, *otherRelation* e *noRelation*. A relação *noRelation* é aplicada quando uma sentença não tem relação com qualquer outra do texto.

Os atributos foram extraídos durante a análise sintática dos textos. Foram utilizadas informações referentes à presença de marcadores discursivos, termos anafóricos nas sentenças, verbo principal e as relações entre esses verbos das sentenças, obtidas da WordNet (Fellbaum, 1998) e distância entre as sentenças.

Os autores utilizaram o algoritmo RandomForest (Ho, 1995) e um córpus de redações anotadas com relações retóricas foi criado para a avaliação do trabalho. Os classificadores gerados obtiveram acurácia máxima de 0,884, em comparação com a anotação humana.

duVerle & Prendinger (2009) utilizaram a técnica de aprendizado SVM com um rico conjunto de atributos para a realização da análise discursiva, segundo a RST. A segmentação discursiva não foi considerada nesse trabalho, pois diversos segmentadores discursivos estavam disponíveis, com bons resultados, tais como o de Soricut & Marcu (2003).

duVerle & Prendinger (2009) focaram na identificação das relações retóricas e utilizaram o conjunto de 18 relações, como em Soricut & Marcu (2003), levando a análise até a construção da árvore discursiva. Os atributos foram agrupados em diversos níveis: i) organização textual, ii) pistas lexicais, iii) pistas sintáticas, iv) conjunto de dominâncias, v) critério de composicionalidade e vi) estrutura discursiva da subárvore.

No nível da organização textual, tem-se atributos que indicam se i) segmentos pertencem à mesma sentença, ii) se pertencem ao mesmo parágrafo, iii) tamanho dos segmentos em *tokens*, iv) tamanho em UBDs, v) distância do segmento ao início do texto e vi) distância do segmento ao fim da sentença, em *tokens*

Para os atributos do nível de pistas lexicais, foi gerado um dicionário de n-gramas ($n \in \{1, 2, 3\}$) construído do córpus de treinamento. A vantagem, segundo os autores, é que esse tipo de abordagem, além de cobrir os marcadores discursivos, leva em consideração pistas como pontuações. No nível de pistas sintáticas, para diminuir a dependência do léxico do texto, foram consideradas as etiquetas morfossintáticas dos três primeiros e três últimos *tokens* de cada segmento.

Semelhante a Soricut & Marcu (2003), os autores utilizaram atributos extraídos dos conjuntos de dominâncias das árvores sintáticas lexicalizadas. Os atributos levaram em consideração tanto as etiquetas morfossintáticas quanto os *tokens* do texto.

Os dois últimos níveis de atributos levam em consideração as subárvores discursivas. Assim, podem ser aplicados quando se busca por uma relação entre *spans* (formados por mais de uma UBDs). Foram consideradas as nuclearidades dos segmentos e padrões nas estruturas das subárvores. Posteriormente à identificação das relações, a construção da árvore discursiva foi feita numa abordagem ascendente (*bottom-up*).

Os autores obtiveram a acurácia de 0,668 na etapa de identificação das relações retóricas e uma das contribuições salientadas pelos autores foi o tempo linear da análise discursiva.

Subba & Di Eugenio (2009) apresentam uma metodologia baseada em representações lógicas de primeira ordem (FOL - *First Order Logic*). Baseiam-se na composição de informações semânticas (quando disponíveis), e utilizam algoritmos de *shift-reduce* para a construção da árvore.

Como técnica de aprendizado para geração dos classificadores das relações retóricas, utilizam o *Inductive Logic Programming* (ILP), que aprende de representações em lógica de primeira ordem. Esses classificadores geram regras em FOL, linguisticamente interpretáveis. Segundo os autores, classificadores baseados em ILP têm mais acurácia que os tradicionais classificadores de algoritmos de aprendizado de máquina, como árvores de decisão e *Naive Bayes*.

Dados os objetivos dos autores, de desenvolver recursos para interfaces educacionais, o córpus utilizado é composto de textos instrucionais do Inglês. Tais textos contêm instruções de como fazer a reparação de problemas em residências.

Para a obtenção da semântica composicional dos textos, foi utilizada a VerbNet (Kipper *et al*, 2000) como léxico de verbos, permitindo a identificação dos papéis semânticos presentes nas UBDs. Essa etapa de anotação foi realizada automaticamente, exceto pela escolha da árvore sintática dentre as derivadas pelo *parser* sintático, que foi realizada manualmente.

Já para a anotação da análise discursiva do córpus, foram consideradas 26 relações, agrupadas em 5 classes: *Causal*, *Elaboration*, *Similarity*, *Temporal* e *Other*. Para a construção da árvore discursiva, foi empregado um algoritmo modificado de *shift-reduce*, si-

milarmente ao utilizado por Marcu (2000).

Na identificação das relações retóricas, utilizando ILP, foi considerada uma base de conhecimento B e um conjunto de exemplos E. E contém exemplos de cada relação. Um exemplo positivo para uma determinada relação é negativo para todas as outras relações. B contém atributos que possibilitarão o aprendizado de regras para a classificação das relações. Essa base contém atributos dos seguintes tipos: i) lista de verbos e substantivos, ii) hiperonímia e hiponímia dos substantivos, iii) pistas linguisticas, como marcadores discursivos, iv) similaridade de cosseno, v) semântica formal (que leva em consideração a sintaxe da língua) e vi) informações estruturais, como se os segmentos pertencem a uma mesma sentença ou não. Ao se relacionar segmentos que contêm mais de uma UBD, os valores dos atributos do nível intrassentencial são obtidos apenas do segmento mais nuclear da subárvore discursiva.

Utilizando B e E, regras FOL são aprendidas. Como os classificadores ILP são binários, um classificador para cada relação foi criado. Para a classificação de uma nova instância, os classificadores são ordenados equacionando o número de exemplos positivos, negativos e quantidade de orações nos segmentos. O classificador que ficar em primeiro colocado é, então, aplicado para identificação de uma relação.

A etapa de classificação de relações obteve um desempenho de 0,631 no nível intrassentencial. Já para o nível intersentencial, a medida-F cai para 0,354, dada a dificuldade da tarefa, inclusive considerando a propagação de erros do nível intrassentencial.

Wellner *et al* (2009) utilizam uma variedade de atributos sintáticos e léxico-semânticos na criação de classificadores de Entropia Máxima. Como córpus, foi utilizado o Graph-Bank, gerado segundo a modificação estrutural da RST proposta por Wolf & Gibson (2005), que permite o relacionamento entre segmentos não adjacentes, o que leva a uma estrutura em formato de grafo, sem as restrições aplicadas à estrutura em árvore.

O córpus foi anotado com diversos níveis de conhecimento e nove classes de atributos foram definidas sobre tais conhecimentos: 1) palavras no início e fim dos segmentos, 2) atributos binários indicando proximidade entre os segmentos, por exemplo, *distância menor que 3* ou *distância maior que 10*, 3) caminhos de tamanho de até 10 palavras

no texto, buscados em uma ontologia semântica, 4) similaridade lexical entre pares de palavras, utilizando a medida *Word Sketch Engine* (WSE) (Kilgarriff *et al*, 2004), 5) núcleo dos eventos identificados nos segmentos, seguindo as especificações do TimeML[1], 6) relacionamento entre os eventos identificados (tipicamente, um evento é um verbo no texto, que indica um acontecimento), 7) pistas lexicais em um segmento relacionado com um evento de outro segmento, 8) dependências gramaticais entre os segmentos e 9) *links* temporais entre os eventos.

Os autores realizaram diversos experimentos de classificação das relações retóricas combinando as classes de atributos. A conclusão é de que os atributos das classes 1 e 2 são os mais úteis na classificação. No entanto, todos os atributos foram utilizados no *parser* final, visto que cada classe de atributos é útil para determinadas relações retóricas.

Retiradas as restrições de relacionamento para a composição estrutural de uma árvore, permitindo a geração de um grafo entre os segmentos, é possível a combinatória $\binom{n}{2}$, que pode levar a um número impraticável de combinações, quanto maior for n. Para resolver essa questão de desempenho, foram consideradas apenas relações entre segmentos em uma mesma sentença. Os autores obtiveram o desempenho de 0,645 na identificação das relações retóricas.

Hernault *et al* (2010) utilizaram a técnica de aprendizado SVM para o desenvolvimento de um *parser* discursivo (HILDA - *High-Level Discourse Analyzer*). Em sua metodologia, os autores utilizaram o aprendizado supervisionado tanto para a segmentação discursiva quanto para identificação das relações RST. O córpus utilizado para o aprendizado foi o RST-DT (Carlson *et al*, 2001). Foram consideradas 18 relações discursivas tal como as utilizadas no *parser* SPADE.

Diferentemente de Soricut & Marcu (2003), que trabalharam apenas no nível intrassentencial, esse trabalho produz uma estrutura discursiva (arbórea) para todo o texto.

Na Tabela 3.1, são apresentados os atributos utilizados pelos autores. O primeiro grupo (Organização Textual) contém atributos tais como i) UDBs na mesma sentença, ii) estar no mesmo parágrafo, iii) tamanho em tokens, iv) tamanho em UBDs, v) distância do

[1]http://www.timeml.org

início da sentença, em tokens, vi) distância do início do texto, em tokens, e vii) distância do fim da sentença, em tokens. Esses atributos buscam capturar informações estruturais que indicam a relação discursiva entre dois segmentos textuais. No segundo grupo, os autores buscaram expandir as informações obtidas a partir da árvore sintática lexicalizada, como i) distância da UBD à raiz da árvore sintática, ii) palavra mais nuclear do nó mais alto da árvore sintática do *span*, iii) etiquetas morfossintáticas dos nós pais e iv) etiquetas morfossintáticas dos nós irmãos. 0,90

Atributo	**Escopo**
Organização Textual	
Mesma sentença	Par de *spans*
Mesmo parágrafo	Par de *spans*
Número de bordas de sentenças	*span*
Número de bordas de parágrafos	*span*
Tamanho em tokens	*span*
Tamanho em UBDs	*span*
Distância ao início da sentença em tokens	*span*
Tamanho do segmento sobre tamanho da sentença em tokens	*span*
Tamanho do segmento sobre tamanho da sentença em UBDs	*span*
Tamanho do par de segmentos sobre tamanho da sentença em UBDs	Par de *spans*
Distância ao início da sentença em UBDs	*span*
Distância ao início do texto em tokens	*span*
Distância ao fim da sentença em tokens	*span*
Sintáticos - conjuntos de dominância	
Distância à raiz da árvore sintática	*span*
Distância ao ancestral comum na árvore sintática	*span*
Média das distância ao ancestral comum	Par de *spans*
Cabeça lexical do nó dominante do segmento	*span*
Etiqueta morfossintática do ancestral comum	Par de *spans*
Cabeça lexical do ancestral comum	Par de *spans*
Etiqueta morfossintática do nó dominante do segmento	Par de *spans*
Cabeça lexical do nó dominante do segmento	Par de *spans*
Etiqueta do nó dominante do segmento irmão	Par de *spans*
Cabeça lexical do nó dominante do segmento irmão	Par de *spans*

Tabela 3.1: **Conjunto de atributos adaptados do *parser* HILDA. Os atributos são agrupados em dois conjuntos: Organização Textual e Sintáticos (relacionados aos conjuntos de dominância).**

Cada atributo pode ter o escopo de um único *span* (uma ou mais UBDs, consideradas como um único segmento textual) ou um par de *spans*. Na etapa da identificação das relações discursivas, os autores obtiveram a medida-F de 0,668 utilizando o RBF (*radial basis function*) como *kernel* do SVM.

Posteriormente, Feng & Hirst (2012) desenvolveram uma metodologia baseada na do *parser* HILDA, melhorando o passo de construção da árvore discursiva pela incorporação

de atributos linguísticos, tais como i) relações anteriores e posteriores aos segmentos com os quais se está trabalhando, ii) regras de produção discursiva, indicando prováveis ordens de ocorrência das relações RST, iii) medidas de similaridades semânticas e iv) marcadores discursivos. Os autores realizaram diversas avaliações entre esse trabalho e o de Hernault *et al* (2010) e os resultados foram melhores em todos os casos, considerando o resultado final do *parser*: a árvore discursiva.

Nesse trabalho, além dos atributos utilizados por Hernault *et al* (2010), foram incorporados atributos adaptados de Lin *et al* (2009). Esse último trabalho foi desenvolvido para outra teoria discursiva, presente no córpus (PDT) *Penn Discourse Treebank* (Prasad *et al*, 2008). Essa teoria discursiva define relações explícitas (quando indicadas explicitamente por algum marcador discursivo) e implícitas (quando não há marcadores discursivos explícitos no texto). Outra importante diferença está na definição dos segmentos discursivos, em que, para cada relação, são definidos dois argumentos (*arg1* e *arg2*) que não são distinguidos pela noção de nuclearidade. Nessa teoria, não se constrói uma estrutura discursiva hierárquica, como na RST. Embora os atributos tenham sido empregados para a PDT, essa teoria também é relacional, como a RST.

Na avaliação, foram obtidas as medidas-F de 0,763 e 0,334 para os níveis intrassentencial e intersentencial, respectivamente. Novamente, vê-se que, no nível intersentencial, o desempenho na tarefa cai drasticamente, dada a dificuldade de identificar o relacionamento entre segmentos textuais maiores que uma UBD.

Recentemente, Joty *et al* (2015) desenvolveram o CODRA (*COmplete probabilistic Discriminative framework for Rhetorical Analysis*) em que buscaram tratar de três questões limitantes dos trabalhos de *parsing* discursivo:

i) identificação da estrutura e relações discursivas separadamente;

ii) utilização de algoritmos gulosos na construção da estrutura discursiva e;

iii) *parsing* intra e intersentencial sem distinção.

Para o tratamento das três questões enumeradas, os autores utilizaram CRFs (*Conditional Random Fields* - Sutton & McCallum (2012)) para inferir as probabilidades de

todos os constituintes da árvore discursiva simultaneamente. Os CRFs são utilizados para representar a estrutura e relações discursivas conjuntamente, capturando as dependências sequenciais na análise discursiva. Para tratar a segunda questão, utilizam probabilidades inferidas, a posterior, das CRFs na construção da árvore discursiva num algoritmo *bottom-up* probabilístico ótimo. Por fim, para tratar a terceira questão, os autores utilizam abordagens diferentes para a análise intrassentencial e intersentencial, a exemplo do que já havia sido feito por Feng & Hirst (2012).

Para a segmentação discursiva, Joty *et al* (2015) utilizam um modelo de Entropia Máxima. Feita a segmentação, realizam o *parsing* intra e intersentencial em sequência para obtenção da árvore discursiva. Esse tratamento simultâneo da estrutura, relações discursivas e suas nuclearidades só é possível com uma quantidade grande de dados de treinamento, o que não é o caso do Português, como será mostrado posteriormente, na descrição dos corpora disponíveis para essa tarefa.

Muitos dos atributos utilizados no treinamento dos modelos foram também utilizados em trabalhos anteriores, como os atributos do nível de organização textual (tamanho do segmento em UBDs e *tokens*, distâncias do segmento ao início ou fim da sentença ou texto, contagens de sentenças e parágrafos nos segmentos). N-gramas foram utilizados com o intuito de capturar pistas além dos marcadores discursivos. Também foram utilizados atributos provindos dos conjuntos de dominâncias como introduzido por Soricut & Marcu (2003), no nível intrassentencial. Cadeias lexicais foram utilizadas no nivel intersentencial. Como atributos contextuais, foram utilizados os vetores de atributos anterior e posterior e a raiz das árvores retóricas da direita e da esquerda, no nível intersentencial.

No *parsing* intrassentencial, CODRA identificou as relações retóricas com medida-F de 0,798, na seção de teste do córpus RST-DT (Carlson *et al*, 2001). No nível intersentencial, CODRA obteve medida-F de 0,558.

Os trabalhos apresentados nesta subseção serviram como base para a exploração de um diverso conjunto de atributos na tarefa de identificar as relações retóricas. A exploração foi realizada seguindo abordagens supervisionadas de aprendizado, como base para a exploração semissupervisionada sem fim com grande quantidade de dados.

3.1.2.3 Aprendizado semissupervisionado e não supervisionado

Marcu & Echihabi (2002), diferentemente dos trabalhos anteriores, utilizaram uma metodologia não supervisionada no aprendizado de algumas relações discursivas, a saber: *Contrast*, *Cause-Explanation-Evidence*, *Elaboration* e *Condition*.

A ideia principal desse trabalho foi aumentar o conjunto de treinamento com dados anotados manualmente por meio de padrões lexicais. Inclusive, objetivou-se melhorar a identificação de relações (do rol mencionado) que não fossem explicitamente indicadas por marcadores discursivos. Como exemplo, os autores observaram que apenas 61 das 238 ocorrências da relação *Contrast* e 79 das 307 ocorrências de *Cause-Explanation-Evidence* eram indicadas por alguma pista lexical.

Foram utilizados dois grandes corpora do Inglês: um composto de mais de 41 milhões de sentenças sem nenhuma anotação (chamado de *Raw* córpus) e o córpus BLIPP, com quase 2 milhões de sentenças anotadas sintaticamente pelo *parser* de Charniak (Charniak, 2000). Esses dois corpora foram processados automaticamente para a extração de pares de sentenças que contivessem uma das pistas lexicais que indicassem uma das relações consideradas. Além de pares de sentenças, dependendo da pista lexical que ocorresse no meio de uma sentença, a mesma era segmentada na pista lexical e extraída para uso na etapa de treinamento dos classificadores. A pistas lexicais utilizadas neste trabalhos são apresentadas na Tabela 3.2, em que *BOS* e *EOS* indicam início da sentença (*begin of sentence*) e fim da sentença (*end of sentence*), respectivamente.

Por exemplo, na ocorrência da pista *But* no início de uma sentença, essa sentença e a anterior são extraídas e rotuladas como mantendo a relação *Contrast*. Isso foi feito com todas as pistas lexicais das quatro relações. Feita a extração e, objetivando a identificação de instâncias implícitas dessas relações (sem a presença da pista lexical), para o treinamento dos classificadores, as pistas lexicais foram removidas.

Com as pistas lexicais removidas, o modelo de aprendizado utilizado foi o *Naive Bayes*, partindo da hipótese de que, a despeito dos marcadores discursivos, os pares de palavras entre os segmentos fornecem informações para a detecção da relação retórica. Por exemplo, dois segmentos discursivos que têm, cada qual, as palavras *bom* e *ruim*, podem indicar

Tabela 3.2: **Pistas lexicais utilizadas por Marcu & Echihabi (2002) para obtenção automática de instâncias de treinamento para a criação de classificadores.**

Contrast (3.881.588 exemplos)
(BOS ... EOS) (BOS *But* ... EOS)
(BOS ...) (*but* ... EOS)
(BOS ...) (*although* ... EOS)
(BOS *Although* ... ,) (... EOS)
Cause-Explanation-Evidence (889.946 exemplos)
(BOS ...) (*because* ... EOS)
(BOS *Because* ... ,) (... EOS)
(BOS ... EOS) (BOS *Thus*, ... EOS)
Condition (1.203.813 exemplos)
(BOS If ... ,) (... EOS)
(BOS If ...) (*then* ... EOS)
(BOS ...) (*if* ... EOS)
Elaboration (1.836.227 exemplos)
(BOS ... EOS) (BOS ... *for example* ... EOS)
(BOS ...) (*which* ...,)

a presença da relação *Contrast*. Assim, duas palavras w_i e w_j de segmentos a serem relacionados podem indicar uma relação r_k. Define-se a probabilidade $P(w_i, w_j|r_k)$ utilizando estimadores de máxima verossimilhança (*maximum likelihood estimators*).

Durante a avaliação, a curva de aprendizado foi monitorada, de acordo com a quantidade de dados utilizados para o treinamento. Verificou-se que, a partir de determinada quantidade de dados, o desempenho do aprendizado não melhorava significativamente. Por exemplo, para as relações *Contrast* e *Cause-Explanation-Evidence*, os classificadores obtiveram desempenho de 0,871 com 2.000.000 instâncias de treinamento e 0,873 quando a quantidade foi aumentada para 4.771.534.

A aplicação dessa metodologia levou a uma melhoria significativa na identificação das relações retóricas tratadas. Por exemplo, utilizando apenas as pistas textuais, é possível identificar 60 das 61 relações marcadas das 238 ocorrências de *Contrast*. Com o aprendizado automático, foi possível identificar mais 123 das relações não marcadas pelas pistas textuais. Isso fez com que a acurácia fosse de 0,26 para 0,77.

Sporleder & Lascarides (2008) utilizaram a metodologia empregada por Marcu & Echihabi (2002) com o intuito de verificarem a plausibilidade da aplicação de instâncias marcadas por pistas lexicais na identificação de instâncias com relações implícitas.

Diferentemente do trabalho que buscaram validar, os autores chegaram à conclusão de que essa não é uma estratégia conveniente ao tratar relações com marcadores discursivos ambíguos, ou seja, os exemplos com pistas textuais explicitas não ambíguas podem não generalizar bem para instâncias com ambiguidade em relação à relação discursiva.

Para que a estratégia utilizada por Marcu & Echihabi (2002) funcione, duas condições devem ser satisfeitas. Primeiramente, além dos marcadores discursivos (pistas textuais), um contexto linguístico entre os segmentos deve ser indicativo da relação discursiva, de forma que, retirando-se os marcadores discursivos, ainda sobrem pistas para a identificação da relação retórica. Em segundo lugar, instâncias com relações explícitas e implícitas devem ser suficientemente similares para o bom comportamento dos classificadores treinados com as instâncias cujas pistas lexicais foram removidas. Um dos argumentos para a dúvida em relação ao trabalho anterior é que os autores apenas reportam a cobertura dos classificadores, não indicando bem como os classificadores se comportam no conjunto de teste.

Em Sporleder & Lascarides (2008), foram utilizadas outras 5 relações: *Contrast*, *Result*, *Explanation*, *Summary* e *Continuation*. Essas relações foram escolhidas dado que há marcadores discursivos não ambíguos que as identificam e também há casos em que não é necessária a ocorrência de marcadores para essas relações.

Para a rotulação automática de novas instâncias, foram utilizados 55 padrões lexicais semelhantes aos utilizados por Marcu & Echihabi (2002). Foi utilizado o córpus BNC (*British National Corpus*), que contém 100 milhões de palavras, e outros dois corpora de notícias jornalísticas, um com 350 milhões de palavras e outro com 1,7 bilhões de palavras.

Foram extraídos 8.495, 14.978, 16.718, 1.490.274 e 6.753.105 exemplos, respectivamente, para as relações *Continuation*, *Result*, *Summary*, *Explanation* e *Contrast*.

O trabalho de Sporleder & Lascarides (2008) foi realizado sobre relações da SDRT, que tem um conjunto diferente de relações. No entanto, os autores realizaram um mapeamento entre as definições das relações RST para as semânticas dinâmicas da SDRT e obtiveram um conjunto de exemplos que trataram manualmente para a identificação das cinco relações (RST) citadas. O valor *kappa* da concordância entre os anotadores foi

0,592.

Para o treinamento dos classificadores, foram utilizadas duas técnicas, uma mais simples como no trabalho de Marcu & Echihabi (2002), o *Naive Bayes*, e uma mais complexa, combinando regras de decisão com *boosting*. Na primeira técnica, tal como no trabalho anterior, foram utilizadas as co-ocorrências de palavras para a geração de probabilidades. Para a segunda, foram utilizados 41 atributos provindos de um *parsing* superficial dos segmentos textuais, agrupados em seis classes: i) atributos de posição, ii) atributos de tamanho, iii) lexicais, iv) etiquetas morfossintáticas, v) atributos temporais e vi) de coesão.

Numa primeira avaliação, as técnicas de aprendizado foram testadas em instâncias cujos marcadores são não ambíguos. Nessa avaliação, os classificadores *Naive Bayes* obtiveram medida-F de 0,407. Embora não comparável, o resultado obtido por Marcu & Echihabi (2002) foi 0,497, utilizando um conjunto de mais de 10 milhões de exemplos. Com a segunda técnica, combinando classificadores de regras de decisão, obtiveram medida-F de 0,596.

Numa segunda avaliação, os autores testaram ambas as técnicas em exemplos cujas pistas textuais são ausentes ou são ambíguas. Nessa avaliação a medida-F dos classificadores *Naive Bayes* cai para 0,192 e das regras de decisão para 0,245.

Em outra avaliação, foi utilizado apenas o córpus anotado manualmente (1.051 exemplos), com metade para treino e metade para teste. Nessa avaliação, os classificadores *Naive Bayes* obtiveram 0,129 e as regras de decisão 0,337. Com esses resultados, verificou-se que, utilizando uma técnica mais elaborada e dados anotados manualmente, levou-se a um resultado melhor do que utilizando dados anotados automaticamente, indo de 0,245 para 0,337.

A conclusão a que chegaram os autores é que, quando modelos treinados com (mesmo que muitos) dados anotados automaticamente, com o uso de marcadores não ambíguos, o aprendizado não generaliza bem para dados cujos marcadores são ambuíguos.

Numa abordagem similar às anteriores, Chiarcos (2012) desenvolveu uma metodologia não supervisionada de geração de uma base de exemplos de segmentos correlacionados

(chamada de *Background Knowledge Base* - BKB), com vistas à identificação de relações RST. A premissa do autor é que todo par de segmentos (ou eventos contidos nos segmentos) tem uma preferência de relação discursiva, indicando um ranque e frequência das relações para cada par de segmentos.

Para a criação da BKB, o autor utiliza três passos. Primeiramente, identifica os eventos contidos nos segmentos e os marcadores discursivos que os ligam. Aqui, os eventos são considerados como os verbos principais dos segmentos. Em seguida, para cada par de segmentos, cria uma tripla consistindo dos dois eventos (verbos principais) e o marcador discursivo (chamado pelo autor de palavra de relação). Essa tripla é adicionada à BKB e, no caso de existir alguma ocorrência da tripla, seu valor no ranque é aumentado. Por fim, realizaram-se cortes nas triplas, pela aplicação de testes de significância e correlação.

Foram utilizados dois corpora anotados sintaticamente, totalizando 2,8 bilhões de palavras. O autor conduziu a avaliação considerando três questões: i) significância, ii) reproducibilidade e iii) interpretabilidade.

Para a primeira questão, a resposta é imediata, visto que a aplicação de testes estatísticos é feita na criação da BKB e a alta quantidade de triplas obtidas confirmam a significância da metodologia. Além disso, os corpora utilizados foram particionados para verificação da reproducibilidade, que foi confirmada, pois os resultados obtidos em uma partição puderam ser reproduzidos na outra partição, indicando que tais triplas capturam características invariantes de contexto para a identificação das relações.

Para a última questão, chegou-se à conclusão de que, para os três marcadores discursivos explorados, dois indicam uma relação de contraste (*but*, traduzido por "mas", e *however*, traduzido por "contudo") e um indica relação temporal ou de causa (*then*, traduzido por "então"). Confirmou-se que as triplas com *but* e *however* estavam correlacionadas entre si e não correlacionadas com as triplas marcadas por *then*, permitindo a correta distinção entre as relações RST

3.1.3 Mapa de abordagens e conclusões

Na Figura 3.4, tem-se uma organização dos trabalhos citados neste capítulo, de acordo com a abordagem utilizada pelos mesmos (uso de padrões lexicais e/ou aprendizado de máquina). Na interseção de *Padrões Lexicais* e *Aprendizado de Máquina*, estão os três trabalhos que tratam da identificação de relações retóricas com o uso de técnicas não supervisionadas e com o uso de grandes quantidades de dados.

Nesta pesquisa, também utilizou-se grande quantidade de dados não rotulados com vistas a melhorar o desempenho da identificação automática das relações RST, mas com o diferencial de que esse uso foi feito de forma incremental, seguindo a ideia (que será apresentada no próximo capítulo) do aprendizado sem fim, verificando-se quais dados contribuíram ou não para a melhoria no aprendizado.

Ao invés do uso em sequência de padrões lexicais e aprendizado de máquina, nessa pesquisa utilizaram-se essas duas abordagens de forma simultânea e cooperativa no *framework* de aprendizado semissupervisionado sem fim, que também será descrito mais à frente nessa tese.

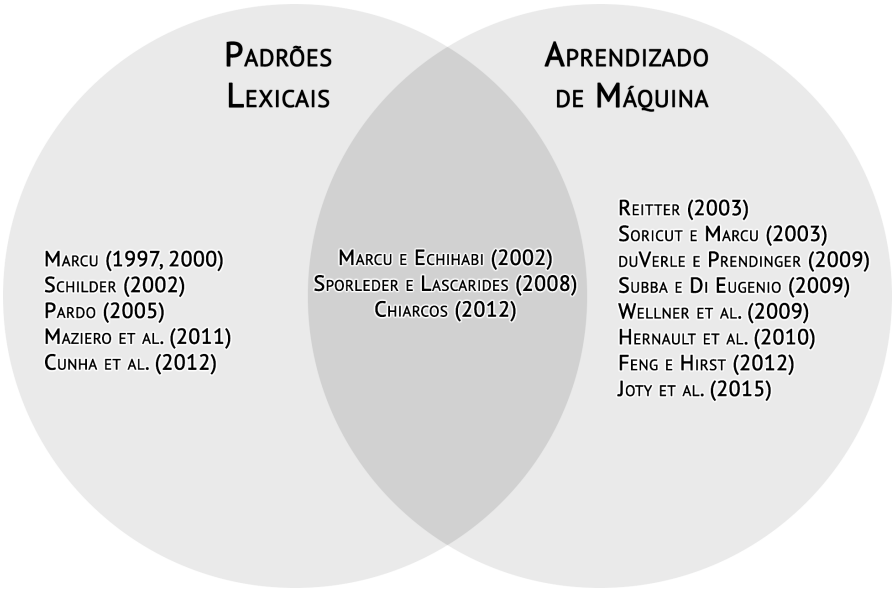


Figura 3.4: **Mapa das abordagens utilizadas no *parsing* discursivo.**

Tanto para a geração de padrões lexicais quanto para a criação de classificadores de

forma supervisionada, há a necessidade de dados rotulados. Tais dados, quando gerados de forma manual, são custosos de se obter. Desta forma, a presente pesquisa mostra potencial na obtenção de bons resultados, buscando suprir a necessidade por mais dados rotulados para um melhor aprendizado, num contexto mais amplo que os apresentados por Marcu & Echihabi (2002), Sporleder & Lascarides (2008) e Chiarcos (2012), que consideraram um contexto com poucas relações RST.

Condizente com a literatura, a identificação das relações retóricas no nível intersentencial é ainda um grande desafio. Os atributos utilizados nos trabalhos capturam informações suficientes ao nível intrassentencial. Para o nível intersentencial, no entanto, tem-se um aumento na subjetividade da tarefa, não ficando claro como capturar as intenções do autor do texto no relacionamento de segmentos compostos por diversas sentenças.

A seguir, são apresentados os principais trabalhos de segmentação discursiva considerados nesta pesquisa.

3.2 Segmentação discursiva

Há diversos trabalhos de segmentação discursiva para diversas línguas. Aqui, relatam-se os principais para o Inglês (Soricut & Marcu (2003), Hernault *et al* (2010) e Joty *et al* (2015)) e para o Português (Pardo (2005) e Maziero *et al* (2007)).

3.2.1 Soricut & Marcu (2003)

No trabalho (*parser* SPADE) de Soricut & Marcu (2003), a segmentação discursiva é realizada em dois passos: segmentação sentencial e intrassentencial. Na segmentação intrassentencial, os autores utilizam um modelo estatístico que calcula a probabilidade de se inserir uma quebra na sentença para cada palavra. A segmentação é feita com base nas probabilidades calculadas, escolhendo quando inserir a quebra. Para realizar essa tarefa, as informações encontradas nas árvores sintáticas lexicalizadas são utilizadas. Essas informações são obtidas de forma semelhante aos conjuntos de dominância, utilizados na identificação das relações retóricas entre segmentos discursivos e discutidos mais à frente.

Assim, dada uma sentença $s = p_1 p_2 ... p_i ... p_n$ com n palavras, o *parser* de Charniak (Charniak, 2000) é utilizado para obter a análise sintática AS da sentença s. Posteriormente, regras canônicas de projeção de núcleos (*head* (Magerman, 1995)) são utilizadas para lexicalizar os nós internos da árvore sintática. O modelo estatístico define uma probabilidade $P(q_i|p_i, AS)$ em que $q_i \in \{quebra, naoquebra\}$ para cada palavra. Essa probabilidade é calculada com o uso da Equação 3.7.

$$P(q_i|p_i, AS) = \frac{C_i(N_p \rightarrow N_w \downarrow N_r)}{C_i(N_p \rightarrow N_w N_r)} \tag{3.7}$$

O numerador indica a contagem de $N_p \rightarrow N_w \downarrow N_r$ em que uma quebra $\downarrow$ é inserida. O denominador indica a quantidade de vezes em que esse padrão ocorre. Os valores N_p, N_w e N_r são nós da árvore sintática, em que N_w é a etiqueta sintática e a palavra flexionada de p, N_r é a etiqueta e a palavra flexionada do irmão de p e N_p é a etiqueta e palavra flexionada do nó superior que tenha como cabeça a palavra p. O segmentador discursivo insere uma quebra quando $P(q_i|p_i, AS) > 0,5$. Os autores reportam uma medida-F de 0,831 quando a análise sintática automática é utilizada. Esse valor sobe para 0,847 quando a árvore sintática é criada manualmente. Uma árvore sintática criada manualmente contém bem menos erros que uma gerada automaticamente por algum *parser*.

3.2.2 Pardo (2005)

O *parser* DiZer realiza um processo simples de segmentação, tanto sentencial quanto intrassentencial. Para a identificação das sentenças do texto, são utilizadas pontuações (como ponto final, ponto de exclamação e ponto de interrogação). Para evitar segmentar após uma abreviação, o segmentador consulta uma lista de abreviações.

No caso da segmentação intrassentencial, busca-se segmentar em orações e, para isso, o segmentador busca marcadores textuais que indicam claramente uma relação retórica e sinais de pontuação (como vírgula e dois pontos, por exemplo). Dada a anotação morfossintática do texto a ser segmentado, verifica-se se em todos os segmentos há pelo menos um verbo. Essa foi uma decisão de projeto que evita segmentos sem verbos, para

garantir uma segmentação automática mais consistente com o manual de segmentação de Carlson & Marcu (2001).

Com o uso da segmentação sentencial, o segmentador obteve uma medida-F de 0,314; já com o uso da segmentação oracional, obteve medida-F de 0,568, quando avaliado em textos do gênero científico, para o qual foi desenvolvido. Quando avaliado para textos jornalísticos, o segmentador obteve medida-F de 0,513 na segmentação oracional.

3.2.3 Hernault *et al* (2010)

Hernault *et al* (2010) utilizam as mesmas informações que Soricut & Marcu (2003) para realizar a segmentação (*parser* HILDA), mas consideram um contexto maior. Além das informações de N_p, N_w e N_r da palavra na posição i, também consideram essas informações da posição $i-2$ e $i-1$. Em vez de um modelo estatístico, os autores utilizam a técnica SVM (Vapnik, 1995) com função de base radial como *kernel* para a criação de um classificador binário que indica ou não quebra após cada palavra dentro de uma sentença.

Utilizando o *parser* sintático de Charniak (Charniak, 2000), o segmentador obteve uma medida-F de 0,94, bem próxima da concordância humana na tarefa, que é de 0,98. Os autores mostraram o ganho de performance ao incluir mais contexto nos atributos. Com o uso das informações léxico-sintáticas (N_p, N_w e N_r) da posicão i, a medida-F obtida foi de 0,80. Acrescentando a posição $i-1$, essa medida sobre para 0,92 e, considerando também $i-2$, o resultado foi 0,94.

3.2.4 Joty *et al* (2015)

Joty *et al* (2015) utilizam um classificador binário para identificar as quebras dentro de uma sentença (*parser* CODRA). O modelo que utilizam é o de Máxima Entropia (Ratnaparkhi, 1996), mais especificamente, eles criam um classificador de regressão logística com parâmetro θ, como na Equação 3.8, em que se define a probabilidade de inserir uma quebra q após a palavra p, representada pelo vetor de atributos x. São utilizadas as distribuições *Bernoulli* a *Sigmóide* (logística).

$$P(q_i|p_i, \theta) = Bernoulli(q_i|Sigmoide(\theta x)) \tag{3.8}$$

Como atributos da segmentação, é utilizado o mesmo conjunto de informações utilizadas na segmentação do *parser* SPADE. Além dessas informações, são utilizadas informações provindas de um analisador sintático superficial, que identifica sintagmas e etiquetas morfossintáticas, tendo em vista que raramente um sintagma seja quebrado para formar duas UBDs. Também foram utilizadas informações da organização textual, como posição relativa na sentença e distâncias em relação ao início e ao fim da sentença. Todos esses atributos foram extraídos considerando o contexto de cada palavra p, para as posições i, $i-1$ e $i+1$.

No córpus RST-DT (Carlson *et al*, 2001), os autores executaram os segmentadores discursivos dos *parsers* SPADE, HILDA e o segmentador proposto (do *parser* CODRA), e os respectivos resultados obtidos (medida-F) foram: 0,852, 0,741 e 0,901. Portanto, o segmentador proposto é o estado da arte para a tarefa de segmentação discursiva para o Inglês, que conta com uma quantidade razoável de informações discursivas anotadas manualmente (Carlson *et al*, 2001).

3.3 Recursos e Ferramentas

Nesta seção, apresentam-se os principais recursos para o Português utilizados neste trabalho: os corpora disponíveis para extração de conhecimento discursivo, a saber, o córpus CSTNews (Cardoso *et al*, 2011a), o CorpusTCC (Pardo & Nunes, 2004), o Rhetalho (Pardo & Seno, 2005), o Summ-It (Collovini *et al*, 2007) e a base de dados lexical TeP 2,0 - *Thesaurus eletrônico para o Português* (Maziero *et al*, 2008). Apresentam-se, também, diversas ferramentas relacionadas à anotação dos corpora, extração de atributos e geração de modelos, tais como RSTTool - *Markup Tool for Rhetorical Structure Theory* (O'Donnell, 2000), Weka - *Waikato Environment for Knowledge Analysis* (Witten & Frank, 2005), o etiquetador morfossintático MXPOST (Ratnaparkhi, 1996) e os *parsers* sintáticos Palavras (Bick, 2000) e LX-Parser (Silva *et al*, 2010).

Os corpora RST foram utilizados para entendimento da tarefa de análise discursiva, para a geração do segmentador discursivo e regras de identificação de algumas relações retóricas. Eles também são utilizados para treinamento e teste dos classificadores que identificam as relações retóricas e suas nuclearidades. Os outros recursos são utilizado na etapa de extração de atributos para o aprendizado automático. As ferramentas foram utilizadas para extração dos atributos (*parser* Palavras, LX-Parser, MXPOST, Lematizador e Cadeias Lexicais), para a anotação dos corpora RST (RSTTool) e para a busca dos textos na web (Crawler).

3.3.1 Recursos

A análise discursiva, conforme já descrita, consiste em identificar as relações entre os segmentos discursivos e construir uma estrutura arbórea com essas relações, mostrando a coerência global de um texto. Um córpus anotado com a teoria RST permite entender melhor as relações, assim como automatizar a tarefa por meio da criação de modelos de aprendizado, tais como classificadores e regras manualmente criadas.

3.3.1.1 CSTNews

Cardoso *et al* (2011a) compilaram um córpus chamado de CSTNews. O córpus é composto, conforme a Tabela 3.3, por 140 documentos, totalizando 2088 sentenças com 5216 relações RST. A anotação RST foi feita por um grupo de pesquisadores do NILC[2] seguindo um manual de anotação criado para a tarefa. Antes da anotação dos textos do córpus, uma etapa de treinamento foi realizada e, periodicamente, a concordância na tarefa foi mensurada (Tabela 3.4).

Tabela 3.3: **Estatísticas do córpus CSTNews**

Total de documentos	140
Total de sentenças	2088
Total de palavras	47240
Total de relações retóricas	5216

[2]http://www.nilc.icmc.usp.br/ - Núcleo Interinstitucional de Linguística Computacional

As relações presentes no córpus e suas frequências são mostradas no gráfico da Figura 3.5. O desbalanceamento das relações é perceptível e é uma questão importante a ser considerada, principalmente na criação dos modelos para identificação automática dessas relações. Por exemplo, a relação *Elaboration* tem mais de 1500 instâncias, enquanto que outras relações têm menos de 10 exemplos, como *Otherwise* e *Solutionhood*. Quanto menos exemplos, menos valores de atributos se podem extrair sobre a relação com baixa frequência a fim de generalizar os modelos para outras ocorrências da relação.

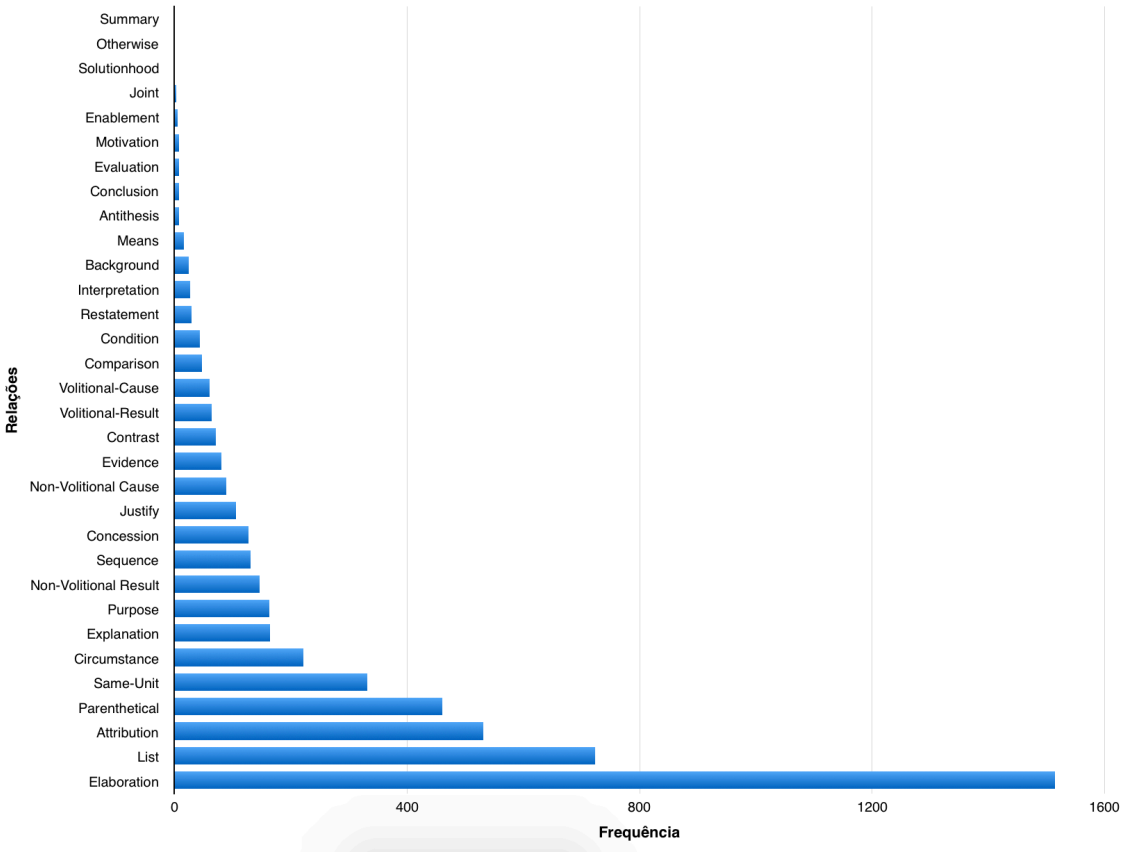


Figura 3.5: **Frequência das relações retóricas no córpus CSTNews**

Observando a concordância obtida nessa anotação (Tabela 3.4), indicando certa sistematização da tarefa, conclui-se ser possível automatizar a análise discursiva com certo nível de desempenho, pois os resultados da anotação (medidos em precisão, cobertura e medida-F entre os anotadores) estão semelhantes aos da literatura, como em Soricut & Marcu (2003), em que os anotadores obtiveram em torno de 0,70 de concordância. Por exemplo, a identificação de segmentos simples tem concordância acima dos 0,90. Esse

valor fica mais baixo para a etapa de identificação de relações RST, 0,66, tendo em vista a subjetividade da tarefa. O córpus CSTNews está publicamente disponível e conta com diversas outras camadas de anotação, como anotação discursiva multi-documento (Radev, 2000), aspectos textuais e desambiguação de sentido verbal e de substantivos. O córpus também conta com diversos sumários extrativos e abstrativos, tanto mono quanto multi-documento.

Tabela 3.4: **Concordância da anotação RST no CSTNews**

Critério de avaliação	**Precisão**	**Cobertura**	**Medida-F**
UBDs	0,91	0,91	0,91
spans	0,78	0,78	0,78
Nuclearidade	0,78	0,78	0,78
Relações	0,66	0,66	0,66

3.3.1.2 Rhetalho

O Rethalho (Pardo & Seno, 2005) é um córpus composto por 50 textos distribuídos conforme a Tabela 3.5. São 30 textos do gênero científico e 20 do gênero jornalístico, sendo esses últimos provindos do jornal online Folha de São Paulo. A anotação foi realizada por dois anotadores e não é relatada concordância entre os mesmos.

Tabela 3.5: **Estatísticas do córpus Rhetalho**

Total de documentos	50
Total de sentenças	168
Total de palavras	3903
Total de relações retóricas	330

A Figura 3.6 apresenta as frequências das relações no córpus Rhetalho. *Elaboration*, novamente, é a relação mais frequente, junto a relações multinucleares como *List*, *Same-unit* e *Sequence*.

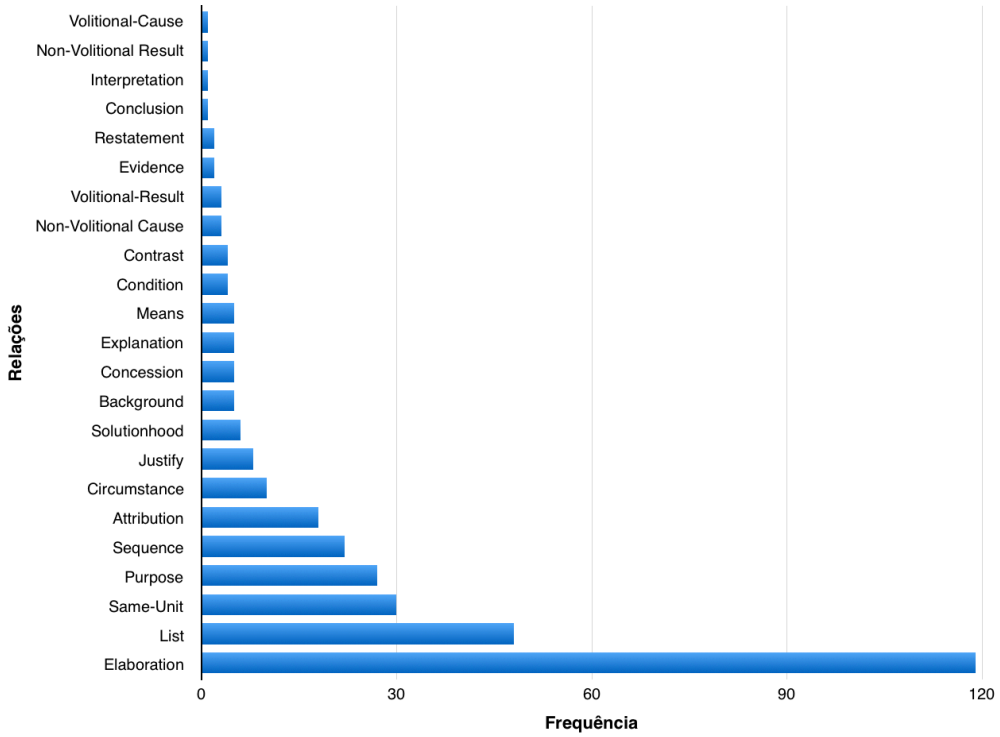


Figura 3.6: **Frequência das relações retóricas no córpus Rhetalho**

3.3.1.3 Summ-it

O córpus Summ-it (Collovini *et al*, 2007) foi compilado e anotado, tendo 50 textos do gênero jornalístico da seção Ciência, do jornal online Folha de São Paulo. Embora tenha o mesmo número de textos do córpus Rhetalho, contém quase cinco vezes mais sentenças.

Tabela 3.6: **Estatísticas do córpus Summ-it**

Total de documentos	50
Total de sentenças	788
Total de palavras	16704
Total de relações retóricas	1677

A Figura 3.7 apresenta a frequência das relações RST no córpus. Tal como nos corpora anteriores, o desbalanceamento é bem notório. As relações *Elaboration*, *Attribution*, *Same-unit* e *Parenthetical* são as mais frequentes, como no córpus CSTNews. É interessante notar que as relações *Volitional-Cause* e *Comparison* no Summ-it estão entre as menos frequentes (2 e 7, respectivamente), enquanto que, no CSTNews, essas relações tiveram

frequência maior (61 e 47, respectivamente).

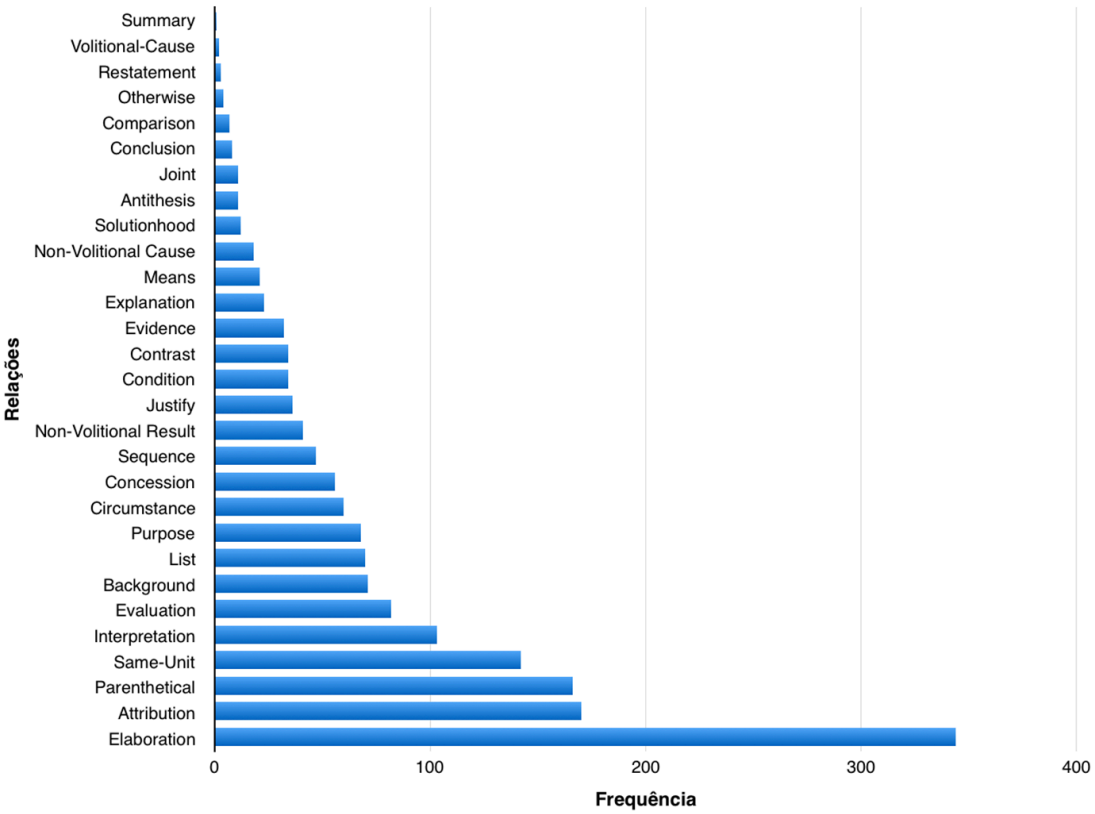


Figura 3.7: **Frequência das relações retóricas no córpus Summ-it**

Também não são relatados resultados de concordância para a anotação do córpus Summ-it.

3.3.1.4 CorpusTCC

O córpus CorpusTCC (Pardo & Nunes, 2004) é composto por 100 textos do gênero científico e domínio da computação. São textos das seções de introdução de qualificações e dissertações de mestrado.

A Tabela 3.7 sumariza as estatísticas do córpus. Embora tenha menos textos que o CSTNews, tem maior número de palavras, tendo em vista que cada documento (seção de introdução de documentos científicos) é maior que um texto jornalístico.

Como o córpus foi anotado por apenas um especialista, não há relato de concordância na anotação. Na Figura 3.8 apresentam-se as frequências de cada relação RST do córpus.

Tabela 3.7: **Estatísticas do córpus CorpusTCC**

Total de documentos	100
Total de sentenças	1350
Total de palavras	53000
Total de relações retóricas	1677

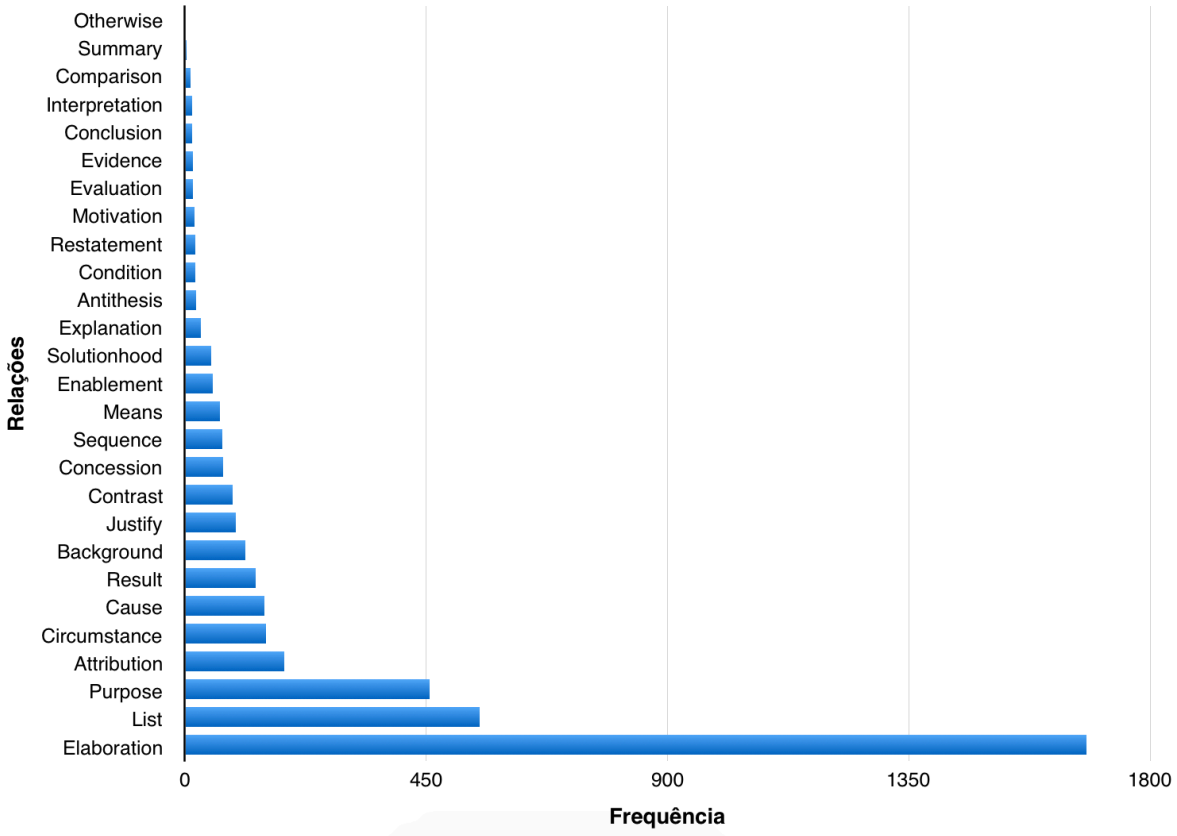


Figura 3.8: **Frequência das relações retóricas no córpus CorpusTCC**

3.3.1.5 Thesaurus para o Português

Outro recurso, do nível léxico e semântico, que foi utilizado nesta pesquisa é a base de dados lexical TeP 2,0 (Maziero *et al*, 2008). Atualmente, a base contém apenas substantivos, verbos, adjetivos e advérbios interligados pelas relações de sinonímia e antonímia. Vale ressaltar que as unidades lexicais (palavras) estão agrupadas pela relação de sinonímia, formando os *synsets*. A relação de antonímia, no entanto, ocorre entre os *synsets* e não entre palavras. A Tabela 3.8 apresenta algumas estatísticas da base.

Considere a palavra "semelhante". Uma busca no TeP retornará os seguintes *synsets*:

Tabela 3.8: **Estatísticas do TeP2 - Thesaurus para o Português do Brasil**

Palavras	44678
Conjunto de sinônimos (*synsets*)	19888
Relações de antonimia entre *synsets*	4276

1) semelhante, afim, análogo, aparente, assemelhado, comparável, conforme, congênere, convizinho, idêntico, mesmo, paralelo, parecente, parecido, símile

2) semelhante, similar

3) semelhante, igual, par, parceiro, parelho

4) semelhante, próximo

Sendo que os *synsets* 1 a 3 são da categoria dos adjetivos, e o 4, dos substantivos. Além dessas informações, o TeP também indica, para o *synset* 1, o seguinte conjunto de antônimos:

1) absemilhante, dessemelhante, diferente, díspar, dissemelhante, dissímil

Esse tipo de informação semântica é útil para a identificação de palavras diferentes com significados próximos, assim como a ocorrência de palavras de sentidos opostos no texto.

3.3.1.6 Listas de Palavras

Tanto durante a segmentação discursiva quanto durante a identificação das relações retóricas, com o uso de regras, foi utilizada uma lista de verbos de atribuição. Essa lista é muito útil na identificação da relação *Attribution*. Assim, como será detalhado no Capítulo 5, o texto é segmentado quando um verbo de atribuição é encontrado e esse mesmo verbo é utilizado como uma pista para a identificação da relação *Attribution*.

Para a identificação do tipo de conjunção presente no início ou fim de cada segmento discursivo, foi utilizada uma lista de conjunções, dado que o *parser* sintático apenas faz a identificação da conjunção coordenativa e subordinativa, não as distinguindo entre

aditiva, adversativa, alternativa, conclusiva, explicativa, causal, comparativa, concessiva, condicional, conformativa, consecutiva, final, proporcional e temporal.

Também foi utilizada uma lista de marcadores discursivos, dado que muitas relações discursivas são explicitamente marcadas por essas pistas. O Apêndice B apresenta as listas de palavras utilizadas.

3.3.2 Ferramentas

Diversas ferramentas estão relacionadas à análise discursiva, úteis às etapas de anotação de córpus (RSTTool - O'Donnell (2000)), extração de atributos (Palavras - Bick (2000), LX-Parser - Silva *et al* (2010) e MXPOST - Ratnaparkhi (1996)), geração de modelos de aprendizado automático (Weka - Witten & Frank (2005)) e avaliação da análise discursiva instanciada em um *parser* (RSTEval - Maziero & Pardo (2009)).

3.3.2.1 RSTTool

Para anotação dos corpora citados, com análise RST, foi utilizada a ferramenta RSTTool (O'Donnell, 2000). Nessa ferramenta, todo o processo de análise discursiva é executado manualmente, por meio de uma interface gráfica. Na Figura 3.9, tem-se a anotação RST do texto apresentado na Introdução, com o uso da ferramenta.

3.3.2.2 *Parser* Palavras

Como evidenciado por diversos trabalhos anteriores (Soricut & Marcu (2003); Sagae (2009), dentre outros), a informação sintática é muito informativa para a realização da análise discursiva, tanto para a identificação das relações, quanto para a segmentação. Para o Português, tem-se o *parser* sintático Palavras (Bick, 2000). O autor reporta um desempenho médio de mais de 0,98 na análise sintática. Embora o desempenho seja alto, na prática diversos erros têm sido relatados. A análise é apresentada em formato arbóreo e, para cada palavra, são apresentadas as seguintes informações: lema, morfossintaxe, sintaxe e classe semântica (para algumas palavras).

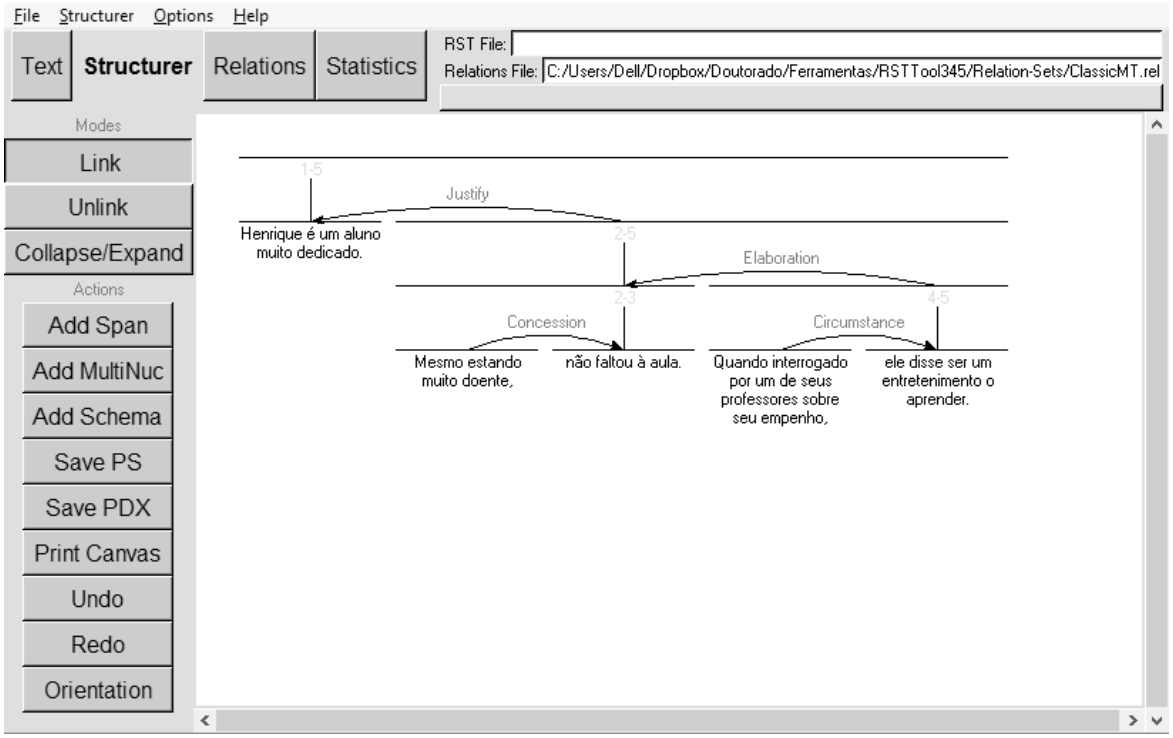


Figura 3.9: **Ferrramenta de anotação RSTTool - (O'Donnell, 2000)**

Como exemplo de análise sintática, considere a árvore obtida pelo Palavras na Figura 3.10.

O mesmo texto do exemplo, em formato textual no formato XML, é exibido a seguir.

```
<s id="s1" ref="1" source="Running text" forest="1" text="Henrique ^^e9 um aluno
          muito dedicado.">
<graph root="s1_500">
<terminals>
<t id="s1_1" word="Henrique" lemma="Henrique" pos="prop" morph="M S" extra="hum *"/>
<t id="s1_2" word="^^e9" lemma="ser" pos="v-fin" morph="PR 3S IND VFIN" extra="fmc vK mv"/>
<t id="s1_3" word="um" lemma="um" pos="art" morph="M S" extra="--"/>
<t id="s1_4" word="aluno" lemma="aluno" pos="n" morph="M S" sem="Hprof" extra="--"/>
<t id="s1_5" word="muito" lemma="muito" pos="adv" morph="--" extra="quant"/>
<t id="s1_6" word="dedicado" lemma="dedicado" pos="adj" morph="M S" extra="jh np-close"/>
<t id="s1_7" word="." lemma="--" pos="pu" morph="--" extra="--"/>
</terminals>
```

```
<nonterminals>
<nt id="s1_500" cat="s">
     <edge label="STA" idref="s1_501"/>
</nt>
<nt id="s1_501" cat="fcl">
     <edge label="S" idref="s1_1"/>
     <edge label="P" idref="s1_2"/>
     <edge label="Cs" idref="s1_502"/>
     <edge label="PU" idref="s1_7"/>
</nt>
<nt id="s1_502" cat="np">
     <edge label="DN" idref="s1_3"/>
     <edge label="H" idref="s1_4"/>
     <edge label="DN" idref="s1_503"/>
</nt>
<nt id="s1_503" cat="adjp">
     <edge label="DA" idref="s1_5"/>
     <edge label="H" idref="s1_6"/>
</nt>
</nonterminals>
</graph>
</s>
```

Nos nós "terminals", podem-se recuperar as informações de lema, *part-of-speech* - *pos*, morfologia e semântica (campo *extra*). Pelos nós "nonterminals", pode-se construir a árvore sintática.

3.3.2.3 LX-Parser

O *parser* LX-Parser (Silva *et al*, 2010) também realiza a análise sintática do texto em Português, mas a apresentação da árvore sintática segue gramática similar aos *parsers* utilizados pelos trabalhos para o Inglês, como o *parser* de Charniak (Charniak, 2000). O LX-Parser utiliza o *parser* de Standford (Klein & Manning, 2003) treinado para o

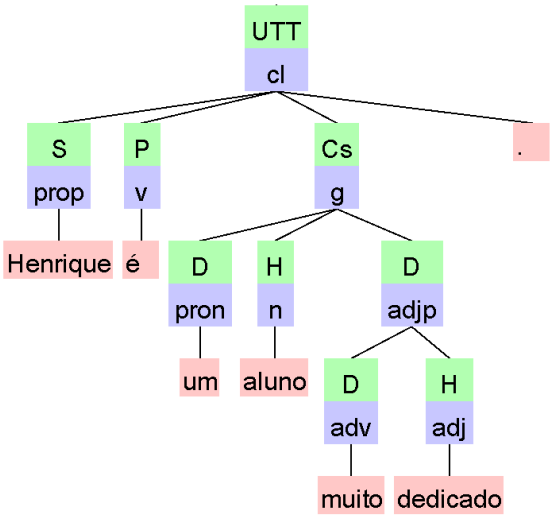


Figura 3.10: **Exemplo de análise sintática pelo Palavras**

Português, com o uso do Treebank CINTIL (Branco & Costa, 2008), no qual obteve medida-F de 0,88.

Como exemplo de análise sintática, considere a árvore obtida pelo LX-Parser na Figura 3.11.

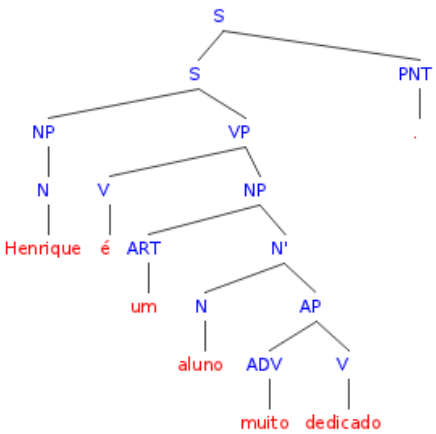


Figura 3.11: **Exemplo de análise sintática pelo LX-Parser**

3.3.2.4 MXPOST

Para extração de conhecimento morfossintático, utiliza-se o etiquetador MXPOST (Ratnaparkhi, 1996), treinado para o Português do Brasil (Aires *et al*, 2000). Esse etiquetador realiza uma análise morfossintática dos textos, apresentando as classes gramaticais das palavras com uma precisão de mais de 0,96.

O texto: "Henrique é um aluno muito dedicado.", após etiquetado pelo MXPOST, tem cada uma das suas palavras marcadas: Henrique__N é__VERB um__ART aluno__N muito__ADV dedicado__ADJ .__.

3.3.2.5 Weka

O Weka (Witten & Frank, 2005) é um ambiente útil à experimentação em aprendizado de máquina por conter diversos algoritmos já implementados e ferramentas de pré e pós-processamento dos dados. Diversos algoritmos de aprendizado podem ser testados, ajustados e avaliados, facilitando a escolha dos melhores atributos e técnicas para a geração de classificadores.

Nesta pesquisa, a interface gráfica da Figura 3.12 foi utilizada apenas para alguns testes iniciais. Durante o desenvolvimento, o Weka foi utilizado por chamadas em linha de comando.

3.3.2.6 Lematizador

Um lematizador foi desenvolvido durante essa pesquisa de doutorado com o uso do MX-POST e a base lexical do UNITEX-BR (Muniz, 2004). O processo é simples e pode ser descrito nos passos do Algoritmo 1.

O texto, inicialmente, é tokenizado, como uma etapa de pré-processamento à etiquetagem morfossintática, que é realizada pelo MX-POST. Feita a etiquetagem, para cada palavra do texto, utiliza-se a etiqueta fornecida para buscar na base do UNITEX uma palavra equivalente, isto é, com mesma flexão e etiqueta morfossintática. Caso uma equivalência não seja encontrada, como no caso de nomes próprios, a palavra original é retornada. Para realizar o mapeamento entre as palavras etiquetadas e o UNITEX, uma API

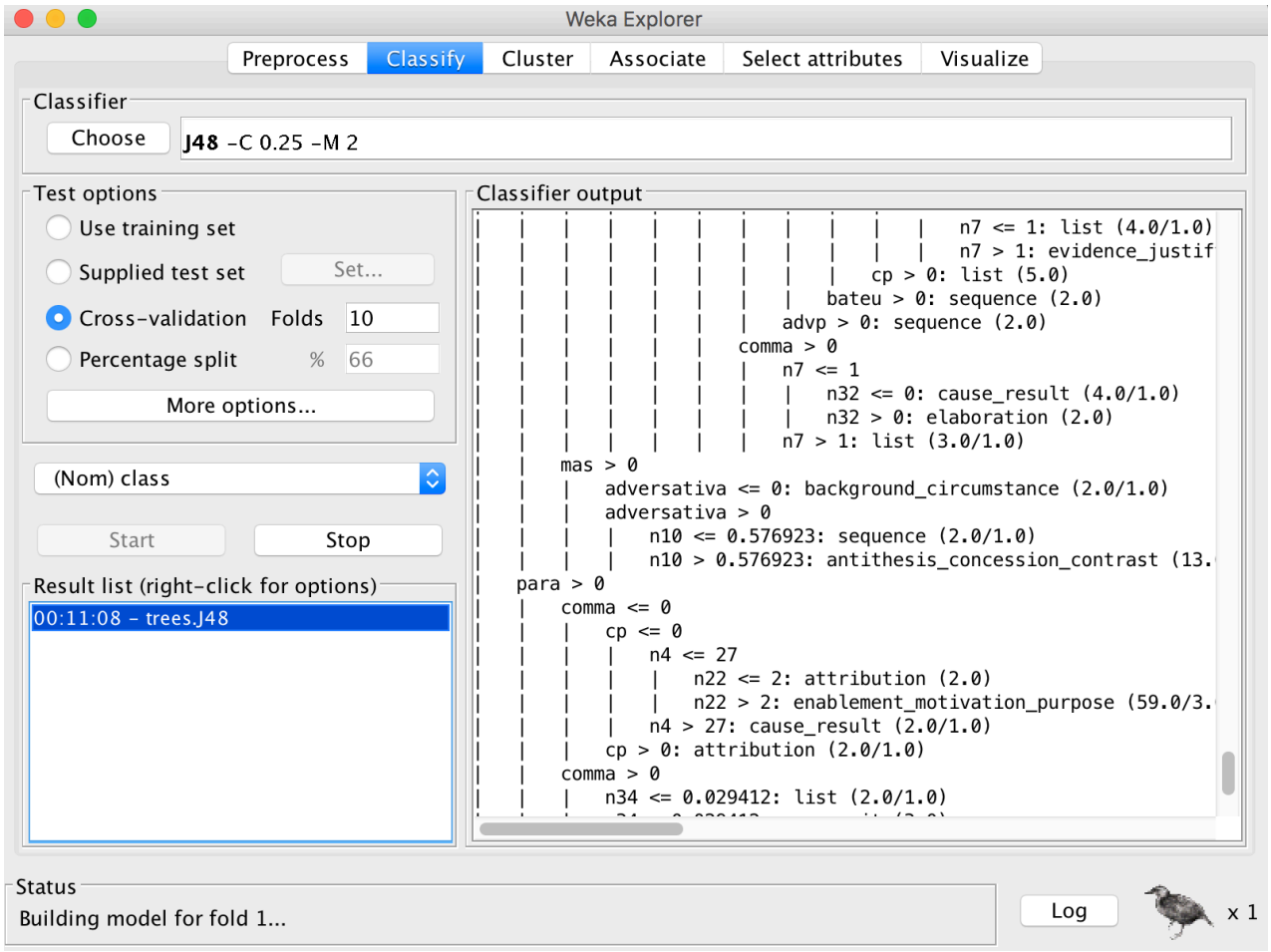


Figura 3.12: **Interface do Weka, para treinamento de classificadores.**

Algoritmo 1: Algoritmo do lematizador

Entrada: *texto* a ser lematizado
Saída: *textoLematizado*

textoTokenizado = tokenizar(*texto*)
textoEtiquetado = etiquetar(*textoTokenizado*)
para *token in textoEtiquetado* **faça**
  se *mapear(etiquetaToken, token) a (etiquetaTokenUNITEX, tokenUNITEX)*
    então
      adicionar *lemaUNITEX* a *textoLematizado*
  fim
  senão
      adicionar *token* a *textoLematizado*
  fim
fim

(*Application Programming Interface*) disponibilizada na página do projeto UNITEX-BR

é utilizada[3].

3.3.2.7 Cadeias Lexicais

Morris & Hirst (1991) introduziram o conceito de cadeias lexicais e as exploraram na identificação da estrutura discursiva de um texto. Uma cadeia lexical é, em essência, uma sequência de palavras do texto que mantêm, entre si, alguma relação, geralmente, semântica. Essa ideia foi explorada nesta pesquisa e, como não se tem conhecimento de uma ferramenta que identifique cadeias lexicais para o Português, um identificador de cadeias lexicais foi desenvolvido e é esquematizado no Algoritmo 2.

No Algoritmo 2, para a função *desambiguadorSentidos*, foi utilizado o desambiguador de sentidos desenvolvido por Nóbrega & Pardo (2014), que faz a indexação dos sentidos desambiguados na WordNet (Fellbaum, 1998). Para isso, é realizada a tradução de cada palavra e utilizada a tradução mais frequente. Tendo a indexação com a WordNet, é utilizado o NLTK - *Natural Language ToolKit* (Garrette & Klein, 2009) para obter os sinônimos e hiperônimos (com os quais se obtêm os hipônimos e irmãos na WordNet). São utilizados os substantivos do texto e, para escolher a melhor cadeia lexical para um substantivo, são utilizadas cinco medidas e, para cada medida, é atribuído um valor de similaridade, a saber:

1) identidade (1),

2) sinonímia (0,8),

3) hiperonímia (0,5),

4) hiponímia (0,5) e

5) se os substantivos são irmãos (0,3)

Assim, para cada substantivo do texto, é escolhida a cadeia lexical com a qual o substantivo tem maior similaridade com as palavras já presentes na cadeia. No caso de

[3]http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/

Algoritmo 2: Algoritmo para identificar cadeias lexicais

Entrada: *texto* a ser processado
Saída: *cadeiasLexicais* do *texto*

```
textoDesambiguado = desambiguadorSentidos(texto)
listaSubstantivos = obterSubstantivos(textoDesambiguado)
para substantivo_i em listaSubstantivos faça
    identidades_i = buscaIdenticos(substantivo_i)
    sinonimos_i = buscaSinonimosWordNet(substantivo_i)
    hiperonimos_i = buscaHiperonimosWordNet(substantivo_i)
fim
cadeiasLexicais = substantivo_1
j = 2
para substantivo_j em listaSubstantivos faça
    maiorSimilaridade = 0
    indiceMaiorSimilaridade = null
    para cadeiaLexical_i in cadeiasLexicais faça
        similaridade_j = identidade(cadeiaLexical_i e substantivo_j) +
         sinonimia(cadeiaLexical_i e substantivo_j) + hiperonimia(cadeiaLexical_i e
         substantivo_j) + hiponimia(cadeiaLexical_i e substantivo_j) +
         irmaos(cadeiaLexical_i e substantivo_j)
        se maiorSimilaridade < similaridade_j então
            maiorSimilaridade = similaridade
            indiceMaiorSimilaridade = i
        fim
    fim
    se indiceMaiorSimilaridade não for null então
        cadeiasLexicais_indiceMaiorSimilaridade recebe substantivo_j
    fim
    senão
        criar nova posição em cadeiasLexicais e adicionar substantivo_j
    fim
fim
```

um substantivo não ter similaridade com quaisquer das cadeias lexicais então existentes para o texto, uma nova é criada com o respectivo substantivo.

Como exemplo, considere o texto segmentado, a seguir:

1) Ao menos 17 pessoas morreram após a queda de um avião de passageiros na República Democrática do Congo.

2) Segundo uma porta-voz da ONU,

3) o avião, de fabricação russa, estava tentando aterrissar no aeroporto de Bukavu em

meio a uma tempestade.

4) A aeronave se chocou com uma montanha

5) e caiu, em chamas, sobre uma floresta a 15 quilômetros de distância da pista do aeroporto.

As cadeias lexicais extraídas estão numeradas, a seguir. Os números subscritos indicam a qual segmento pertence o substantivo.

1) pessoas$_1$, passageiros$_1$

2) queda$_1$

3) avião$_1$, avião$_3$, aeronave$_4$

4) porta-voz$_2$

5) fabricação$_3$

6) aeroporto$_3$, aeroporto$_5$

7) tempestade$_3$

8) montanha$_4$

9) chamas$_5$

10) floresta$_5$

11) quilômetros$_5$

12) distância$_5$

13) pista$_5$

A cadeia lexical 6 indica que os segmentos 3 e 5 tratam sobre um mesmo conceito (*aeroporto*). A cadeia lexical 3, informa que os segmentos 1, 3 e 4 falam sobre um *avião*, que também foi referenciado como *aeronave*.

Essa metodologia de identificação de cadeias lexicais não foi avaliado dada a ausência de um córpus anotado com esse tipo de informação. Os pesos para cada similaridade listada anteriormente foi definido seguindo a ideia de que a similaridade semântica é maior da identidade até a hiponímia.

3.3.2.8 RSTEval

Para avaliação de um *parser* discursivo, a ferramenta RSTEval (Maziero & Pardo, 2009) permite a comparação de duas estruturas discursivas em termos de precisão e cobertura. São quantificados três aspectos da análise discursiva:

i identificação dos segmentos discursivos,

ii identificação das relações retóricas e

iii nuclearidade dos segmentos.

Precisão (Equação 3.9) indicará, de todas as relações (ou segmentos ou nuclearidades), qual a porcentagem corretamente identificada, sendo consideradas corretas as relações que também foram feitas na anotação manual.

$$precisao = \frac{relacoes_corretamente_identificadas}{todas_relacoes_identificadas_automaticamente} \tag{3.9}$$

A medida cobertura (Equação 3.10) indicará, de todas as relações (manuais) na anotação do córpus, qual a porcentagem que foi corretamente identificada automaticamente.

$$cobertura = \frac{relacoes_corretamente_identificadas}{todas_relacoes_identificadas_manualmente} \tag{3.10}$$

3.3.2.9 Crawler

Tendo em vista a necessidade de grande quantidade de dados para a realização do aprendizado semissupervisionado sem fim, jornais online web foram utilizados para obtenção de textos. A forma de obter os textos é pelo uso de um web crawler. Um web crawler

é um programa que obtém, de forma automática, páginas da web e realiza o seu *download*. Uma questão importante no desenvolvimento de um web crawler é decidir quais páginas devem ser obtidas, considerando o gênero e domínio textual com o qual se vai trabalhar, nesse caso o web crawler é chamado preferencial. Quando um crawler obtém indiscriminadamente as páginas da web, é chamado de crawler universal (Liu, 2011).

Um crawler foi desenvolvido com o uso de algumas ferramentas já desenvolvidas e descritas no Algoritmo 3. O esforço para adequação de algum web crawler já desenvolvido justificou a criação de um crawler simplificado que atendeu bem às necessidades desse projeto de pesquisa, a saber, notícias do domínio jornalístico, em Português e com filtragem do conteúdo contendo apenas o texto principal da notícia jornalística.

Algoritmo 3: Algoritmo do web crawler

```
Entrada: Lists toVisitURL and visitedURL
Saída: Repositório crescente de textos

repita
    para link em toVisitURL faça
        se link não está em visitedURL então
            obter page indicada por link com JSpider
            se page esta em Português com uso de guessLanguage então
                obter todos os links presentes em page com JSpider e armazenar em
                  nextLinks
                extrair texto principal de page com NCleaner
                gravar texto principal no repositório de textos
                armazenar link em visitedURL
            fim
        fim
    fim
    substituir conteúdo de toVisitURL por nextLinks
até 1=1;
```

Para obter a página indicada por *link*, assim como os links presentes na página, foi utilizado o JSpider[4]. Para verificar se a página está em Português, uma biblioteca para a linguagem Python, Guess-Language[5], foi utilizada. Para extrair apenas o texto principal da página obtida, foi utilizado o NCleaner (Evert, 2008), treinado para o Português com

[4]http://j-spider.sourceforge.net
[5]https://pypi.python.org/pypi/guess-language

páginas dos seguintes jornais online: Estadão[6], Folha de São Paulo[7], Terra[8], G1[9], BBC[10], Exame[11], Notícias ao Minuto[12], Época Negócios[13], iG[14], R7[15], SBT[16], Zero Hora[17], Diário de SP[18], Diário de Pernambuco[19], Jornal do Brasil[20], O Povo[21], O Globo[22] e Jornal a Cidade[23]. O modelo treinado do NCleaner está disponível na página do Projeto Sucinto[24].

Essa mesma lista de jornais online serviu como lista inicial ($toVisitURL$) ao web crawler.

[6]http://www.estadao.com.br
[7]http://www.folha.uol.com.br
[8]http://www.terra.com.br
[9]http://www.g1.globo.com
[10]http://www.bbc.co.uk/portuguese
[11]http://exame.abril.com.br
[12]http://www.noticiasaominuto.com
[13]http://epocanegocios.globo.com
[14]http://www.ig.com.br
[15]http://www.r7.com
[16]http://www.sbt.com.br/jornalismo/
[17]http://zh.clicrbs.com.br/rs/
[18]http://diariosp.com.br
[19]http://www.diariodepernambuco.com.br
[20]http://www.jb.com.br/capa/
[21]http://www.opovo.com.br
[22]http://oglobo.globo.com
[23]http://www.jornalacidade.com.br
[24]http://conteudo.icmc.usp.br/pessoas/taspardo/sucinto/resources.html

4

Noções Básicas de Aprendizado de Máquina

O AM (aprendizado automático) desempenha papel fundamental na automação da análise discursiva e outras tarefas complexas. A escolha dos atributos a serem utilizados é um dos principais passos para o êxito no AM. Dados valores para um conjunto de atributos, os algoritmos de aprendizado de máquina visam identificar rótulos, ou classes, como as relações retóricas. Para isso, necessitam de conhecimento previamente classificado, ou rotulado, para a geração de modelos que codifiquem o aprendizado.

Geralmente, o conhecimento é obtido por meio de especialistas do domínio da tarefa. Por exemplo, o córpus CSTNews (Cardoso *et al*, 2011a) foi anotado por especialistas em RST e serve como base para a codificação do aprendizado da análise discursiva automática. Esse processo de geração de conhecimento por especialistas é muito custoso e, para se obter bons resultados, é necessário que se tenham muitos dados anotados. Vale salientar que a capacidade de aprendizado é considerada como essencial para que um problema seja tratado de forma inteligente. É nessa concepção que o aprendizado automático se relaciona com a área de IA (inteligência artificial (Faceli *et al*, 2011)).

Mitchel (1997) define o aprendizado automático como a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Assim, o aprendizado automático analisa exemplos da tarefa que se deseja executar, extraindo conhecimento prático para melhorar seu desempenho na realização posterior da tarefa, mas com dados ainda não vistos.

A fim de exemplificar como uma tarefa pode ser tratada com técnicas de AM, considere

os passos ilustrados na Figura 4.1. Inicialmente, deve-se entender bem o fenômeno que se deseja tratar, definindo o que se deseja realizar por meio de um algoritmo de AM. Na análise discursiva automática, por exemplo, deseja-se identificar a relação retórica que ocorre entre dois segmentos textuais de acordo com o intento do autor do texto.

Figura 4.1: **Etapas simplificadas para o uso do AM no tratamento de um problema.**

Posteriormente, devem-se reunir dados que sirvam como exemplos da tarefa. Para a identificação das relações retóricas, os dados de treinamento serão pares de segmentos textuais rotulados manualmente com as relações retóricas. Com o entendimento do fenômeno a ser tratado e tendo os dados de treinamento disponíveis, deve-se pensar em quais características (atributos) serão utilizadas pelas técnicas de AM para resolver o problema. No caso da identificação das relações entre segmentos textuais, as características poderiam ser, por exemplo, a presença de marcadores discursivos, os tipos de orações, e similaridades semânticas entre os segmentos, dentre outros (Taboada & Das, 2013).

Essas características, ou atributos, constituem ponto chave na aplicação do aprendizado automático. Para que se obtenha sucesso nessa abordagem, é necessário que se entenda o problema e se escolham os melhores atributos que possam distinguir entre as classes do problema. As classes do problema são os rótulos (relações discursivas) que se objetivam serem identificados pelo modelo.

Definidos os atributos, aplicam-se as técnicas de AM sobre os valores desses atributos extraídos dos dados de treinamento. Essas técnicas geram modelos que são baseados nos valores de atributos dos exemplos de treinamento, extraindo e codificando o conhecimento de como resolver a tarefa em questão.

Como o aprendizado automático é a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência, uma avaliação dos modelos gerados deve

ser feita. Para uma mesma tarefa, diversas técnicas e conjuntos de atributos podem ser utilizados e a avaliação permitirá escolher e aprimorar o melhor modelo do aprendizado realizado.

Os passos, definidos acima, são seguidos na maior parte das abordagens de aprendizado automático, todavia, há outras abordagens ou paradigmas que diferem, principalmente, na obtenção e uso dos dados de treinamento (rotulados ou não) e na definição das classes do problema tratado. Essas abordagens, ou paradigmas, são apresentados a seguir.

4.1 Paradigmas de aprendizado automático

Considere o exemplo abaixo, em que uma sentença está segmentada em dois segmentos (UBDs), indicados pelos colchetes, e a resposta esperada (classe) consiste em uma relação RST (*Concession*) e a ordem da nuclearidade (*SN* indica que a primeira UBD é *Satélite* e a segunda é *Núcleo*):

Instância: *[Mesmo estando muito doente] [não faltou à aula]*

Resposta esperada: *Concession - SN*

Um exemplo rotulado consiste em uma instância do problema e o valor de resposta esperado, fornecido por um especialista do domínio. No caso da análise discursiva, o exemplo anterior é rotulado. Um exemplo não rotulado consiste apenas dos dados sem a resposta esperada (classe), por exemplo, apenas os segmentos textuais, sem a relação e nuclearidade dos segmentos.

Conforme simplificadamente ilustrado na Figura 4.2 (inferida de Faceli *et al* (2011)), os paradigmas de aprendizado podem ser:

(i) supervisionados;

(ii) não supervisionados e;

(iii) semissupervisionados.

Vale salientar que o paradigma semissupervisionado é uma forma de tratar o aprendizado supervisionado ou não supervisionado (Abney, 2007). No paradigma supervisionado,

utilizam-se apenas dados rotulados e se objetiva a identificação das classes do problema. No não supervisionado, utilizam-se dados não rotulados e o modelo gerado busca identificar agrupamentos das instâncias de treinamento. Já no paradigma semissupervisionado, tanto a supervisão quanto a não supervisão são modificadas com dados não rotulados ou rotulados, respecivamente, buscando atender melhor o problema em tratamento.

Figura 4.2: **Organização dos paradigmas de aprendizado automático.**

Esses três paradigmas são detalhados a seguir em termos de como são os dados de treinamento, como esses dados são utilizados e como resolver o problema em questão.

4.1.1 Supervisionado

Nesta categoria, as técnicas utilizam-se de dados rotulados e são dependentes dos mesmos. Aprendem até onde os dados rotulados permitem e geram um modelo tão representativo do problema quanto forem os dados. Os dados rotulados supervisionam o aprendizado, informando, para cada exemplo processado, qual o rótulo ou classe que se espera identificar.

Comumente, neste paradigma, têm-se técnicas que objetivam a predição dos dados, isto é, geram modelos que visam a classificar novos dados em um dos rótulos dos exemplos utilizados no aprendizado.

Para muitos problemas, como análise discursiva, tem-se um número definido de classes: as relações retóricas. Nesses casos, os modelos gerados são chamados de classificadores, pois rotulam cada nova instância com uma das classes já definidas do problema. Quando

se trata uma instância e se atribui um valor contínuo (e não discreto) à classe, o modelo é chamado de regressor.

Limitada pelos dados anotados disponíveis, essa abordagem é insuficiente para um bom aprendizado de muitas tarefas do PLN, principalmente para tarefas complexas e subjetivas, como a análise do discurso. Ter um córpus representativo é um grande desafio para muitos fenômenos linguísticos, o que motiva o uso dos outros paradigmas de aprendizado, como o semissupervisionado, que será apresentado logo.

4.1.2 Não supervisionado

Neste paradigma, diferentemente do supervisionado, utilizam-se exemplos não rotulados na geração de modelos que descrevem os dados de entrada, seja sumarizando, associando ou agrupando os mesmos. Esse paradigma é utilizando, em geral, quando se deseja encontrar padrões ou tendências nos dados e não classificar um exemplo em um conjunto pré-definido de rótulos (classes).

A sumarização tem como objetivo encontrar uma descrição simplificada dos dados de entrada. A associação busca padrões frequentes de associação entre os atributos dos dados de entrada. O agrupamento é a tarefa mais comum a esse paradigma e consiste em identificar grupos nos dados baseados, principalmente, na similaridade dos mesmos.

Partir de um paradigma não supervisionado pode levar a resultados cuja interpretação não seja muito intuitiva, principalmente quando o alvo seja um grupo de classes que se conheça a priori. No caso da análise discursiva, o aprendizado não supervisionado pode não definir grupos (ou *clusters*) que sejam equivalentes às relações discursivas, dependendo dos atributos utilizados. Isso motiva ainda mais o uso do aprendizado semissupervisionado nesta tese.

4.1.3 Semissupervisionado

Esse paradigma de aprendizado utiliza tanto dados rotulados quanto não rotulados. É utilizado, geralmente, quando o conjunto de exemplos rotulados é insuficiente para um aprendizado satisfatório, e a geração de mais dados rotulados é custosa, se feita por

humanos. Nesse paradigma, há técnicas bem conhecidas, como o *self-training* (Zhu, 2008) e o *co-training* (Blum & Mitchell, 1998).

A ideia básica do *self-training* é apresentada no Algorítmo 4 (Abney, 2007), em que se utilizam os exemplos rotulados (L_0) para a geração de um modelo preditivo, ou classificador (M) com aprendizado supervisionado. O modelo M é utilizado em dados não rotulados (U), a fim de predizer as classes às quais pertencem. A fim de aumentar os dados de treinamento, são selecionados os exemplos que foram classificados com maior confiabilidade (função *selecionar*) e treina-se novamente o modelo (M) com a junção dos exemplos iniciais e os que foram selecionados pela aplicação do modelo M. Essa iteração continua enquanto se verifica que os modelos aprendidos estão melhorando de desempenho e finaliza quando algum critério de parada é atingido.

Algoritmo 4: Algoritmo simplificado do *self-training*

Entrada: L_0 and U

Seja L_0 um conjunto de dados rotulados e U um conjunto de dados não rotulados

Treinar modelo M com L_0
repita
  L recebe L_0 + *selecionar* (U,M)
  Treinar modelo M com L
até *Critério de parada ser atingido*;
Retornar modelo M

Essa técnica tem sido empregada em diversas tarefas de PLN, desde antes da formalização dos principais algoritmos de semissupervisão (Blum & Mitchell, 1998), como desambiguação lexical (Yarowsky, 1995), classificação de diálogos (Rosenberg *et al*, 2005) e análise semântica (Carlson *et al*, 2009), dentre outras.

A semissupervisão também pode ser utilizada de forma a tentar melhorar a não supervisão. Por exemplo, o agrupamento de dados pode ser tratado inicialmente com o uso de exemplos rotulados a fim de informar melhor o algoritmo sobre a formação dos grupos com os dados não rotulados.

Conforme o Algoritmo 5, no *co-training*, os exemplos de treinamento são tratados com a divisão dos atributos em dois conjuntos (técnica chamada de multi-visão). Cada

um desses conjuntos deve ser independente e suficiente para treinar, cada qual, um bom classificador. Nessa técnica, cada classificador será utilizado na tentativa de melhorar o desempenho do outro classificador pelo uso dos exemplos que forem classificados com a maior confiabilidade. Assim, treina-se, com cada conjunto de atributos, um classificador (M_1 e M_2). As novas instâncias rotuladas pelos classificadores são selecionadas segundo a confiabilidade fornecida por cada classificador (função *selecionar*) e as instâncias selecionadas são adicionadas ao conjunto de treinamento L. O conjunto aumentado L é utilizado para re-treinar os classificadores M_1 e M_2. Esse processo é repetido até atingir algum critério de parada.

Algoritmo 5: Algoritmo simplificado do *co-training*

Entrada: L and U

Seja L um conjunto de dados rotulados e U um conjunto de dados não rotulados

Treinar modelo M_1 com $visao_1(L)$
Treinar modelo M_2 com $visao_2(L)$
repita
  L recebe L + *selecionar* (U,M_1) + *selecionar* (U,M_2)
  Treinar modelo M_1 com $visao_1(L)$
  Treinar modelo M_2 com $visao_2(L)$
até *Critério de parada não é atingido*;

O *co-training* pode ser modificado de diversas formas. Por exemplo, em vez de utilizar multi-visão dos atributos, utiliza-se o mesmo conjunto de atributo (mono-visão) em dois classificadores de técnicas diferentes (Gutiérrez, 2010). Podem-se utilizar, em vez de dois, três classificadores (*tri-training*), de forma que os exemplos que forem classificados pela maioria sejam incorporados aos dados de treinamento (Zhou & Li, 2005). Breiman (1996) propõe o uso de um conjunto de classificadores com a finalidade de escolher as melhores classificações de novos exemplos pelo acordo entre a maioria dos classificadores utilizados.

Para o uso da multi-visão (*co-training*, *tri-training*, dentre outros) o conjunto de atributos deve ser particionado em conjuntos o mais independentes possível, ou seja, os atributos de cada conjunto não devem estar correlacionados com atributos de outros conjunto. No caso de um pequeno conjunto de atributos ou de atributos altamente correlacionados, a

técnica mono-visão *self-training* mostra-se mais adequada.

Foram apresentados diversos conceitos sobre aprendizado automático categorizados nos paradigmas supervisionado, não supervisionado e semissupervisionado. Há ainda alguns conceitos relacionados a essa tese de doutorado que são apresentados a seguir, e têm relação com o uso de grande quantidades de dados no aprendizado semissupervisionado sem fim.

4.2 Grandes quantidade de dados

Conforme o título desta tese, *Análise Retórica com Base em Grande Quantidade de Dados*, a análise discursiva automática é feita com o uso suplementar do máximo possível de dados não anotados manualmente, com vistas a melhorar a performance da tarefa automática. Nesta seção, apresentam-se alguns conceitos relacionados ao uso de aprendizado automático quando esse é realizado com grande volume de dados e, nesse caso, os dados são do tipo textual.

Diversos termos estão relacionados a grandes quantidades de dados, tais como *Big Data*, *Data Intensive* (Lin & Dyer, 2010) e *Web como Córpus* (Kilgarriff *et al*, 2004; Banko & Brill, 2001), e diversos trabalhos em parsing discursivo valem-se de suas vantagens.

Marcu & Echihabi (2002) realizaram um experimento não supervisionado com quatro relações discursivas (*Contrast*, *Explanation-Evidence*, *Condition* e *Elaboration*). A rotulação automática foi feita com o uso de marcadores discursivos. Os dados assim rotulados tiveram os marcadores discursivos retirados para servirem ao treinamento de classificadores *Naive Bayes*. Por exemplo, se um marcador discursivo *but* (mas), foi utilizado para rotular dois segmentos com uma relação *Contrast*, esse marcador é retirado antes de servir como dado de treinamento. Os autores averiguaram que, quanto mais dados não rotulados eram utilizados no aprendizado não supervisionado, maior era a performance dos classificadores para identificar as relações citadas. Esse resultado corroborou o que foi confirmado por Banko & Brill (2001), mostrando que os resultados aumentaram na medida em que se utilizaram mais dados, mesmo que obtidos de forma automática pela

simples aplicação de padrões textuais.

Sporleder & Lascarides (2008) averiguaram o trabalho de Marcu & Echihabi (2002), realizando experimentos conforme já relatado. Os autores chegaram à conclusão que tal abordagem só será bem sucedida se os exemplos não rotulados forem muito similares aos dados de treinamento obtidos automaticamente. Relatam que, se essa premissa não for satisfeita, classificadores treinados com exemplos com marcadores explícitos não generalizarão muito bem para dados sem marcadores explícitos. Além disso, marcadores discursivos podem ser ambíguos, podendo indicar mais de uma relação retórica. Segundo os autores, esse comportamento independe da técnica utilizada na geração dos classificadores, mas depende dos dados utilizados.

Embora não seja um trabalho de parsing discursivo, Banko & Brill (2001) relataram experimentos na tarefa de desambiguar palavras em um conjunto de palavra ambíguas. Esses experimentos utilizaram os dados anotados manualmente e, por algumas abordagens, como votação entre classificadores, *active learning* e aprendizado semissupervisionado, exploraram o uso de grande quantidade de dados não anotados. Os resultados melhoraram com o uso massivo de exemplos não rotulados.

Esses trabalhos mostram que, para obter sucesso no uso massivo de dados, é necessário um tratamento cuidadoso dos exemplos e como serão utilizados numa abordagem semissupervisionada. Questões chave ao se intentar o uso de grande quantidade de dados são quais os dados não rotulados e como busca-los para o uso no aprendizado semissupervisionado. Quanto mais dados são utilizados no treinamento, maior a chance de que um novo exemplo seja relacionado com exemplos já presentes nos dados de treinamento (Banko & Brill, 2001).

4.2.1 Fluxos contínuos - *never-ending learning*

Os trabalhos citados anteriormente utilizaram um conjunto finito de dados não anotados, como é característico em trabalhos com semissupervisão. Essa característica pode ser modificada de forma a considerar um conjunto de dados que é obtido continuamente da web, de forma a iterar o aprendizado semissupervisionado um número não pré-definido

de vezes.

Essa forma de conduzir um aprendizado foi utilizada por Carlson *et al* (2010) e Mitchell *et al* (2015) em uma arquitetura nomeada *Never-ending Learning*. Os autores desenvolveram um ambiente em que o aprendizado é conduzido 24 horas por dia, infinitamente, para obter novas informações e aprender a tarefa de i) identificar sintagmas nominais que se referem a determinada categoria semântica e ii) identificar relações semânticas entre pares de sintagmas nominais. Os autores citam que, em 67 dias de execução, 242.453 novas instâncias foram aprendidas, com 123 categorias semânticas.

Pesquisas recentes têm realizado o treinamento utilizando lotes de dados, ao invés de tratar todos os dados de treinamento de uma vez só (Faceli *et al*, 2011). Esse processamento em lotes ocorre devido à muita informação que é coletada automaticamente, levando a desafios na forma de processar essa grande quantidade de dados, que, inclusive, não cabem simultaneamente na memória principal de um computador.

Como uma solução a essa questão, tem-se pesquisado sobre o aprendizado incremental, em que, a cada novo conjunto de dados de treinamento processado, o modelo de aprendizado é modificado para incorporar novos conhecimentos da tarefa em questão.

No aprendizado incremental, surgem duas questões importantes: i) como tratar, com recursos computacionais limitados, os dados de treinamento e ii) como abordar a mudança do conceito (próxima subseção) que se está a aprender.

Como será apresentado posteriormente (Capítulo 6), a adaptação do parser SPADE (Soricut & Marcu, 2003) permite um aprendizado incremental, pois, a cada nova iteração do aprendizado, basta considerar as novas instâncias, sem a necessidade de revisitar os dados de treinamento de iterações anteriores, apenas o repositório de probabilidades (que consiste em uma redução de dimensionalidade dos dados de treinamento). Isso faz com que o aprendizado seja mais eficiente. No entanto, na adaptação do parser HILDA (Hernault *et al*, 2010), em que se utiliza um algoritmo de árvores de decisão (Quinlan, 1993) para a identificação das relações retóricas, o uso de um algoritmo incremental, como árvores de *Hoeffding* (Hulten *et al*, 2011), pode levar a desempenho menor que as árvores de decisão tradicionais. Portanto, optou-se pelo uso de um aprendizado não incremental, na busca

pela melhor acurácia da identificação das relações retóricas, mesmo que se gaste mais tempo e memória nessa abordagem.

4.2.2 Mudança de conceito - *Concept drift*

No aprendizado sem fim, requer-se o cuidado de verificar o ponto no qual o desempenho do aprendizado estagna ou inicia decréscimo. Essa verificação pode ser feita testando os modelos obtidos em um córpus de referência, a cada iteração, avaliando se o desempenho aumenta ou diminui. Um bom parâmetro a ser levado em consideração é o desempenho humano na tarefa. Esse valor pode ser calculado como a concordância obtida na anotação de um córpus entre vários anotadores. Nessa concordância é importante utilizar a mesma medida de avaliação utilizada para os classificadores. A Tabela 3.4 apresenta a concordância entre os anotadores do córpus CSTNews. Eles concordaram em 0,66 na identificação das relações retóricas, utilizando a medida-F. Valores próximos a esse indicam que os classificadores estão chegando a uma perfomance semelhante à humana na tarefa.

Com o decorrer do tempo, o conceito que se deseja aprender pode sofrer algumas mudanças e, com a coleta de novos dados, os dados anteriores podem não se adequar mais ao conceito modificado. Esse fenômeno é chamado de mudança de conceito (*concept drift*, Klinkenberg (2004)).

Os algoritmos que monitoram a mudança de conceito basicamente monitoram o erro do modelo gerado, pois, se não houver mudança de conceito, o erro deve manter-se constante ou diminuir, com a entrada de novos dados. Se o erro aumentar, é um indício de que houve alguma modificação de conceito. Essa técnica de monitoramento da evolução do erro é chamada de *Statistical Process Control - SPC* (Gama *et al*, 2004).

Curran *et al* (2007) trata de um fenômeno similar, chamado de *semantic drift*, no qual, durante um aprendizado iterativo de classes semânticas (entidades nomeadas), instâncias que prejudicam o aprendizado são incorporadas, aumentando o erro da classificação. Como solução, os autores propõem um caso específico de *bootstrapping*, em que as classes semânticas competem por um termo (entidade nomeada) aprendido.

No próximo capítulo, será apresentada a primeira etapa da análise discursiva, a saber,

a segmentação discursiva. Serão descritos trabalhos adaptados ao Português e as regras que compõem o segmentador discursivo utilizado nesta pesquisa.

5 Segmentação Discursiva

O desafio na segmentação discursiva é identificar segmentos que contenham ideias básicas e que consistam, portanto, em proposições simples, conhecidas como UBDs (*Elementary Discourse Units* - Carlson & Marcu (2001)). Conhecimentos sintáticos e pistas textuais podem ser utilizados com sucesso para a segmentação, como comprovado pelos trabalhos de segmentação descritos a seguir.

Vale salientar que utilizar apenas a segmentação do texto em sentenças, apesar de mais simples, tornaria a análise retórica menos informativa do que potencialmente pode ser, pois as relações intra-sentenciais seriam perdidas. Portanto, a segmentação intrassentencial, em orações e outras porções textuais menores, como na ocorrência das relações *Parenthetical* (em que uma proposição é quebrada por um texto dentro de parênteses) leva a melhores resultados no parsing discursivo. A Figura 2.4 da página 20 apresenta um exemplo em que uma oração é quebrada em três segmentos, devido à ocorrência de um texto entre parênteses. A relação *Same-unit* é utilizada, então, para unir a oração que foi quebrada.

5.1 Segmentador baseado em regras lexico-sintáticas

Dada a importância do conhecimento léxico-sintático no discurso, inclusive na etapa de segmentação, Maziero *et al* (2007) desenvolveu um segmentador discursivo baseado em regras manualmente criadas. O trabalho baseou-se no manual de segmentação de Carlson & Marcu (2001), desenvolvido para o Inglês. Dessa forma, as regras de segmentação foram

adaptadas ao Português, seguindo os exemplos dos corpora disponível até então, a saber, Rhetalho, Summ-it e CorpusTCC.

Durante o desenvolvimento dessa pesquisa, realizaram-se experimentos com AM para a criação de um segmentador discursivo, como feito pelos principais trabalhos da área (Soricut & Marcu, 2003; Hernault *et al*, 2010; Joty *et al*, 2015), mas, devido aos corpora disponível, em que o desbalanceamento é muito grande, os resultados não foram satisfatórios. Embora o desbalanceamento entre as classes 'segmentar' e 'não segmentar' seja um fenômeno natural, pois é bem menos pontos de segmentação que de não segmentação, verificou-se que a abordagem por AM apresentou resultados muito baixos para a classe 'segmentar'.

Optou-se, portanto, por um melhoramento de um segmentador baseado em regras (Maziero *et al*, 2007), que tem mostrado boa performance na tarefa. As melhorias consistiram na adição de novas regras de segmentação, e novas regras de categorização dos segmentos em determinados tipos para auxiliar no processo de segmentação. Como etapa de pós-processamento, algumas regras de restrições são aplicadas, juntando segmentos que foram identificados incorretamente pelas regras de segmentação.

5.1.1 As regras

As regras foram criadas analisando-se exemplos de segmentação provindos dos corpora RST. Na análise dos corpora foram consideradas as palavras do texto e suas informações morfossintáticas. Como pode ser visto nas regras apresentadas posteriormente, as pontuações também desempenham um papel importante na tarefa da segmentação discursiva.

Com o intuito de apresentar a aplicação das regras de segmentação discursiva, o Algoritmo 6 é apresentado, indicando os principais passos do processo. Primeiramente, o texto é analisado sintaticamente ($parsingSintatico$), e a segmentação em sentenças da própria análise sintática é utilizada para identificar cada sentença do texto ($segmentarEmSentencas$). Depois, para cada sentença do texto, sua informação sintática é utilizada para a aplicação das regras ($aplicaRegras$), que identificam pontos (quebras) de segmentação. Como uma etapa de pós-processamento, algumas restrições são aplicadas sobre os segmentos gerados,

com a finalidade de corrigir algumas segmentações que tenham ocorrido de forma incorreta (*aplicaRestricoes*). Por fim, com a lista de pontos de segmentação (*quebrasSegmentos*), o texto é segmentado e retornado.

Algoritmo 6: Algoritmo do segmentador discursivo

Entrada: *texto* a ser segmentado
Saída: *textoSegmentado*

analiseSintaticaTexto = parsingSintatico(*texto*)
sentencasTexto = segmentarEmSentencas(*analiseSintaticaTexto*)
para *sentenca em sentencastexto* **faça**
  analiseSintaticaSentenca = obterAnaliseSintaticaSentenca(*analiseSintaticaTexto*)
  quebrasSegmentos = aplicaRegras(*analiseSintaticaSentenca*)
  quebrasSegmentos = aplicaRestricoes(*quebrasSegmentos*, *analiseSintaticaSentenca*)
  textoSegmentado = geraSegmentos(*quebraSegmentos*,*analiseSintaticaSentenca*)
fim

As regras utilizam as informações provindas da análise sintática e podem ser: $palavra_i$ (a palavra na iésima posição da sentença), $lema_i$ (lema da iésima palavra), pos_i (etiqueta morfossintática), $extra_i$ (informação extra fornecida pelo analisador sintático, como indicação de oração relativa).

A seguir, são listadas as regras nas Tabelas 5.1 a 5.13. Para cada regra, um exemplo de aplicação é apresentado e, em negrito, o padrão utilizado.

Tabela 5.1: **Regra de segmentação 1.**

1. Se pos_i for conjunção subordinativa e pos_{i-1} for preposição, então insira quebra antes de $i - 1$;

   1) Acima de esse valor e até R$ 4.340 , o trabalhador teria uma espécie de bônus em a declaração de o Imposto de Renda que fará

   2) **com que** ele receba uma restituição maior ou pague menos impostos.

Feita a aplicação das regras de segmentação, para cada *segmento*, faz-se a categorização dos segmentos segundo as regras auxiliares das Tabelas 5.14, 5.15 e 5.16 (definindo categorias para alguns segmentos de forma a auxiliar no processo de segmentação, tais

Tabela 5.2: **Regra de segmentação 2.**

2. Se pos_i for conjunção subordinativa e pos_{i-1} for pontuação, então insira quebra antes de i;

   1) Apesar da genômica ambiental parecer ser uma boa idéia,

   2) **como** se pode imaginar, em princípio, não é fácil de executar.

Tabela 5.3: **Regra de segmentação 3.**

3. Se pos_i for conjunção coordenativa, então

   Se houver verbo antes e depois de i, então insira quebra antes de i;

   1) Agora os cientistas já agarram um punhado deles de uma vez

   2) **e** trabalham em todos ao mesmo tempo.

Tabela 5.4: **Regra de segmentação 4.**

4. Se pos_i for verbo no gerúndio e pos_{i-1} for pontuação, então insira quebra antes de i;

   1) Uma forma de tratar este problema é mediante o alinhamento de sequências, que consiste em alinhar um padrão biológico sobre a sequência objetivo**,**

   2) **procurando** semelhanças entre o padrão e a sequência.

Tabela 5.5: **Regra de segmentação 5.**

5. Se pos_i for verbo no particípio e pos_{i-1} for pontuação, então insira quebra antes de i;

   1) A Folha Online apurou que o relator**,**

   2) **preocupado** com o desgaste de um arquivamento imediato da segunda representação contra Renan, decidiu recomendar em seu voto a paralisação das investigações por influência de senadores do próprio PMDB .

como segmentos que não contêm verbo e segmentos que contêm um parênteses aberto e não fechado, por exemplo).

Tabela 5.6: **Regra de segmentação 6.**

6. Se pos_i for advérbio, então

   Se houver verbo antes e depois de i, então insira quebra antes de i;

   1) A PEC que prorroga a cobrança de a CPMF está **tramitando** no Senado,

   2) **onde** precisa de 49 votos para **ser** aprovada.

Tabela 5.7: **Regra de segmentação 7.**

7. Se $extra_i$ indicar cláusula relativa e $palavra_{i-1}$ for uma vírgula, então

   Insira quebra antes de $i+1$ E encontre a próxima $palavra_j$ que for vírgula, com $j\ a\ partir\ de\ i+1$, e insira quebra antes de $j+1$;

   Exemplo 1:

   1) A partir de amostras de um lodo rosado que os cientistas têm a delicadeza de chamar de biofilme, foi possível catalogar dois genomas completos e ainda pedaços de outros três**,**

   2) **cujos** donos eram bactérias e micróbios extremófilos.

   Exemplo 2:

   1) **Se** a amostra a ser analisada contiver milhares de organismos diferentes**,**

   2) há uma dificuldade natural de recompor o quebra-cabeças após a leitura das sequências de DNA.

Tabela 5.8: **Regra de segmentação 8.**

8. Se $palavra_i$ for um parênteses de abertura, então

   Insira quebra antes de i E encontre a próxima $palavra_j$ que for parênteses de fechamento, com $j\ a\ partir\ de\ i+1$, e insira quebra antes de $j+1$;

   1) A opinião é do agrônomo Miguel Guerra, da UFSC

   2) (Universidade Federal de Santa Catarina).

Tabela 5.9: **Regra de segmentação 9.**

9. Se $lema_i$ for um verbo de atribuição e pos_{i-1} for substantivo ou nome próprio, então insira quebra antes de $i + 1$;

   1) A polícia de **São Paulo afirmou**

   2) ontem ter detido dois suspeitos de terem participado do roubo do relógio Rolex do apresentador Luciano Huck.

Tabela 5.10: **Regra de segmentação 10.**

10. Se $lema_i$ for um verbo de atribuição e pos_{i-1} for pontuação, então insira quebra antes de i;

    1) Estou indo para São Paulo amanhã [hoje] para fazer o reconhecimento**,**

    2) **disse** Huck.

Tabela 5.11: **Regra de segmentação 11.**

11. Se $palavra_i$ for igual a "conforme"ou "segundo"e pos_{i-1} for pontuação, então insira quebra antes de i;

    1) As lentes feitas sob medida são destinadas a corrigir problemas de foco, que é responsável pela miopia,

    2) **segundo** oftalmologistas.

Tabela 5.12: **Regra de segmentação 12.**

12. Se $palavra_i$ for igual a "conforme"ou "segundo"e $i <= 2$, então

    Encontre pos_j que for pontuação, com $j \ a \ partir \ de \ i+1$, então insira quebra antes de $j + 1$;

    1) **Segundo** ela**,**

    2) o diretor assinou um termo de compromisso onde dá garantias de que os presos não serão torturados depois do motim.

Tabela 5.13: **Regra de segmentação 13.**

13. Se $palavra_i$ for uma vírgula, então
    Se houver verbo antes e depois de i, então insira quebra antes de $i + 1$;
    1) A Câmara **tem** uma agenda de trabalho**,**
    2) ninguém **está** a reboque de ninguém.

Tabela 5.14: **Regra auxiliar na segmentação 14.**

14. Se *segmento* não contiver verbo, mas for circuncidado por parênteses
    então o *segmento* é *Parenthesis*;
    senão o *segmento* é *noVerb*;

Tabela 5.15: **Regra auxiliar na segmentação 15.**

15. Se *segmento* contiver o parênteses de abertura e fechamento
    então o *segmento* é *Parenthesis*;
    senão se *segmento* contiver apenas parêntese de abertura, então o *segmento* é *ParenthesisIncomplete*

Tabela 5.16: **Regra auxiliar na segmentação 16.**

16. Se *segmento* contiver uma elaboração (pelas regras 2,4 e 5)
    então o *segmento* é *Elaborated*;

Feita a categorização dos segmentos, são aplicadas as restrições das Tabelas 5.17 e 5.18 para cada *segmento* encontrado, com vistas a evitar segmentos sem verbos, exceto aqueles que sejam *Parenthesis*.

Como será visto no Capítulo 6, o *framework* de aprendizado semissupervisionado sem fim utiliza o módulo de identificação de relações discursivas do DiZer. Esse módulo utiliza padrões que trabalham sobre três informações de cada palavra do texto: a palavra flexionada, a palavra lematizada e sua etiqueta morfossintática. Desta forma, o segmentador discursivo produz duas saídas, uma contendo apenas as palavras flexionadas em seus respectivos segmentos e outra em que cada palavra é acompanhada de seu lema e etiqueta morfossintática, possibilitando a aplicação do módulo do DiZer: $palavra_1(lema_1)_pos_1$ $palavra_2(lema_2)_pos_2$ ... $palavra_n(lema_n)_pos_n$

Tabela 5.17: **Regra de restrição 17.**

17. Se $segmento_i$ for $noVerb$ Se $segmento_{i-1}$ não for *Elaborated* e não for *Parenthesis* então juntar $segmento_{i-1}$ com $segmento_i$;

Tabela 5.18: **Regra de restrição 18.**

18. Se *segmento* for *ParenthesisIncomplete* então juntar $segmento_{i-1}$ com $segmento_i$;

5.1.2 Resultados

Para avaliação do segmentador descrito anteriormente, foi utilizada a seção de testes dos corpora RST (Tabela 6.1, página 99). Essa seção é constituída de 449 sentenças e foram consideradas apenas as segmentações intrassentenciais, dado que a segmentação sentencial é bem realizada, inclusive pelos *parsers* sintáticos. A Tabela 5.19 apresenta os resultados obtidos pelo segmentador, aqui denominado SegRegras. Como *baselines*, foram

consideradas as avaliações da segmentação oracional realizada pelo *parser* DiZer, tanto para textos científicos (córpus CorpusTCC - Pardo & Nunes (2004)) quanto para textos jornalísticos (seção de textos jornalísticos do córpus Rhetalho - Pardo & Seno (2005)). É importante salientar que o SegRegras foi avaliado em uma seção de teste composta por textos dos quatro corpora. Já o DiZer foi avaliado nas seções indicadas anteriormente e os valores reportados aqui foram retirados de Pardo (2005).

O resultado do SegRegras, embora não comparável com resultados de trabalhos para o Inglês (por utilizar diferentes conjuntos de dados), está bem próximo do resultado obtido por Joty *et al* (2015), que foi de 0,923 e acima das obtidas por Soricut & Marcu (2003) e Hernault *et al* (2010), respectivamente 0,838 e 0,779. Já a cobertura do SegRegras, 0,769, ficou acima da obtida por Hernault *et al* (2010), 0,706, mas abaixo das obtidas por Soricut & Marcu (2003) e Joty *et al* (2015), respectivamente 0,868 e 0,897.

A metodologia de segmentação do *parser* HILDA (Hernault *et al*, 2010), foi adaptada ao Português, mas seu resultado ficou inferior ao SegRegras, que ficou também acima dos *baselines*. O SegRegras constitui-se no estado da arte para a segmentação discursiva para o Português. Vale salientar que a adaptação do segmentador do HILDA pode utilizar mais ou menos contexto em seus atributos. No caso do uso de menos contexto (quando se usa apenas uma palavra da árvore sintática lexicalizada para definir se há ou não quebra), equivale ao método de segmentação utilizado no *parser* SPADE (Soricut & Marcu, 2003). Assim, o resultado da segmentação da adaptação do SPADE não foi considerada na Tabela 5.19.

Tabela 5.19: **Avaliação do segmentador baseado em regras.**

Segmentador	**Precisão**	**Cobertura**	**Medida-F**
DiZer em textos científicos	0,562	0,573	0,568
DiZer em textos jornalísticos	0,541	0,488	0,513
Adaptação do *parser* HILDA	0,729	0,541	0,621
SegRegras	**0,53**	**0,882**	**0,913**

O SegRegras foi escolhido como segmentador discursivo nesta pesquisa dados os resultados obtidos, mesmo que a maior parte dos trabalhos da área utilizem classificadores gerados por aprendizado de máquina. Um experimento utilizando aprendizado automá-

tico, com os mesmos atributos e técnicas do *parser* HILDA (Hernault *et al*, 2010), que adicionou maior contexto para os atributos do parse SPADE (Soricut & Marcu, 2003), foi realizado e os resultados estão apresentados na Tabela 5.19, como adaptação do *parser* HILDA.

O SegRegras gerou a matriz de confusão da Tabela 5.20, em que a classe segmenta obteve cobertura de 0,769 e precisão de 0,924, o que leva a um bom aproveitamento na identificação das relações discursivas.

Tabela 5.20: **Matriz de confusão da segmentação realizada pelo SegRegras**

Não segmenta	**Segmenta**	← classe identificada
11586	61	**Não segmenta**
224	**744**	**Segmenta**

No próximo capítulo, será detalhada a metodologia de identificação das relações discursivas e da nuclearidade dos segmentos.

6

Identificação de Relações Discursivas e suas Nuclearidades

Neste capítulo, inicialmente, será descrito o uso de aprendizado supervisionado na detecção de relações discursivas entre os segmentos discursivos. Nessa etapa, foram adaptados dois modelos do Inglês (Soricut & Marcu, 2003; Hernault *et al*, 2010). Após a adaptação do último modelo, que expande o conjunto de atributos de Soricut & Marcu (2003), novos atributos foram adicionados em busca da melhor abordagem supervisionada.

Em seguida, o aprendizado semissupervisionado sem fim foi explorado desenvolvendo-se uma arquitetura que dispensa o monitoramento manual do aprendizado (através do controle de mudança de conceito). Essa arquitetura, inclusive, tem um caráter híbrido, pois utiliza classificadores de diversos paradigmas e regras criadas manualmente para a identificação das relações.

Por fim, a detecção da nuclearidade é apresentada, tanto com uma abordagem simbólica (pelo uso da ordem canônica dos núcleos e satélites nas relações), quanto pelo uso de classificadores supervisionados.

Nesta pesquisa, buscou-se modelar a análise discursiva com uma solução de aprendizado de máquina. Para isso, explorou-se o uso de grande quantidade de dados no desenvolvimento de um *framework* de aprendizado semissupervisionado sem fim para a identificação de relações retóricas. Nesse *framework*, trata-se da mudança de conceito no uso dos dados que são continuamente obtidos na web utilizando a ideia geral do SPC, do monitoramento do crescimento do erro, como indicador de mudança de conceito.

6.1 Aprendizado supervisionado

O uso do aprendizado supervisionado depende exclusivamente de dados anotados. Como já tratado anteriormente nesta tese, para o Português conta-se com quantidade insuficiente de dados que levem a resultados satisfatórios na análise discursiva automática com o uso do aprendizado supervisionado. Como o aprendizado semissupervisionado conta com uma mesclagem com a supervisão, nesta pesquisa, inicialmente, exploraram-se abordagens supervisionadas com o intuito de proceder à semissupervisão com a melhor técnica supervisonada.

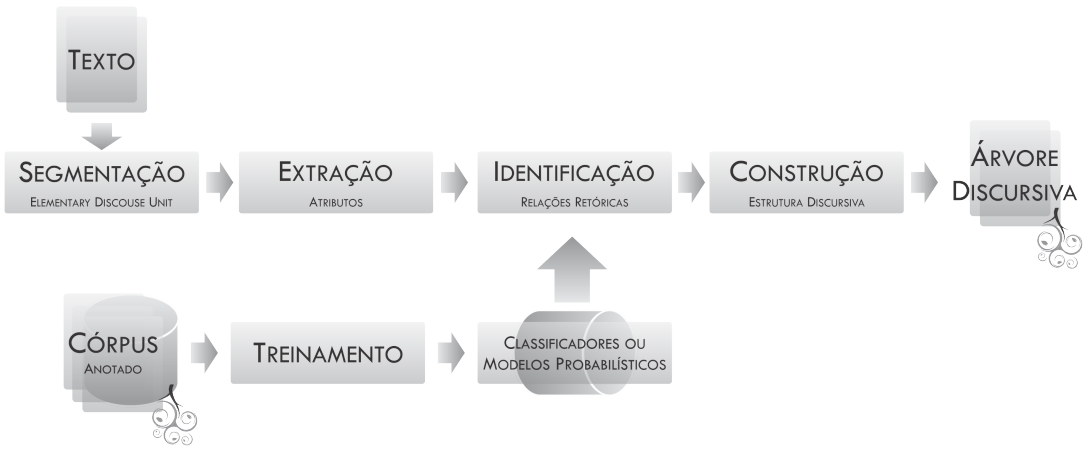


Figura 6.1: **Esquema do aprendizado supervisionado explorado.**

Como ilustrado na Figura 6.1, os corpora RST foram utilizados para o treinamento de um classificador para identificar as relações discursivas. Sua aplicação no *parsing* discursivo se dá da seguinte maneira: primeiramente, segmenta-se o texto em UBDs; depois, para cada par de segmentos adjacentes, extraem-se os atributos para aplicação do classificador treinado no córpus anotado; feita a identificação das relações retóricas (e suas nuclearidades), a árvore discursiva do texto é construída.

Em se tratando dos dados anotados, a Tabela 6.1 resume os números dos quatro corpora anotados com RST utilizados nesta pesquisa e descritos no Capítulo 3.

A Tabela 6.2 apresenta o agrupamento das relações RST que ocorrem no nível intrassentencial e suas frequências. Para o nível inter-sentencial, veja a Tabela 6.3. Esse

Tabela 6.1: **Número de documentos e palavras do conjunto de corpora RST para o Português (composto dos corpora CSTNews, Rhetalho, Summ-it e CorpusTCC).**

Córpus	**Documentos**	**Palavras**
CSTNews - (Cardoso *et al*, 2011a)	140	47.240
Rhetalho - (Pardo & Seno, 2005)	50	2.903
Summ-it - (Collovini *et al*, 2007)	50	16.704
CorpusTCC - (Pardo & Nunes, 2004)	100	53.000
Total	340	119.847

agrupamento é baseado no agrupamento inicial de Mann & Thompson (1987) e incrementado por Soricut & Marcu (2003). Dada a subjetividade de algumas relações bem próximas, como por exemplo, *Evidence*, *Justify* e *Explanation*, elas foram colocadas em um mesmo grupo.

É interessante notar que algumas relações têm maior frequência no nível intrassentencial, como *Attribution*, com 799 ocorrências, contra apenas 8 no nível inter-sentencial. A relação *Same-unit* apenas ocorre no nivel intrassentencial, como esperado, pois é responsável por ligar UBDs quebradas por uma relação *Parenthetical*, por exemplo. Já o grupo formado pelas relações *Interpretation*, *Evaluation* e *Conclusion* apresenta frequência de 40 no nível intrassentencial e 233 no inter-sentencial.

O desbalanceamento das relações nos corpora RST é uma característica natural. A aplicação de técnicas de balanceamento por *oversampling* faria que algumas relações fossem replicadas muitas vezes. Por exemplo, a relação *Summary* ocorre 2 vezes e a relação *Elaboration* ocorre 1491 vezes, tornando a identificação de *Summary* praticamente inviável. Técnicas de balanceamento por *undersampling* não foram cogitadas, pois causaria uma redução considerável nos dados de treinamento. Além do exposto, a replicação em grande número de um pequeno conjunto de instâncias poderia levar ao *overfitting* do modelo aprendido sobre as instâncias replicadas.

As instâncias dos corpora RST foram separadas em conjuntos de treinamento e de teste, seguindo a proporção de 7/10 para treino e 3/10 para teste. Um experimento utilizando a técnica de validação cruzada de 10-pastas foi realizado e é reportado na seção 6.2.2.1. No entanto, como será melhor explicado, essa técnica de avaliação leva ao *overfitting*, pois instâncias classificadas automaticamente em iterações anteriores do aprendizado

Tabela 6.2: **Agrupamento das 29 relações retóricas do nível intrassentencial disponíveis nos corpora RST para o Português em 16 grupos e suas frequências.**

Relação	**Frequência**
Attribution	799
Antithesis Concession Contrast	256
Background Circumstance	362
Volitional-Result Non-Volitional-Result Volitional-Cause Non-Volitional-Cause	449
Comparison	37
Condition Otherwise	104
Elaboration	1491
Enablement Motivation Purpose	695
Evidence Justify Explanation	194
Interpretation Evaluation Conclusion	40
List	703
Means	73
Restatement	28
Same-unit	731
Sequence	199
Summary	2
Joint	0
Solutionhood	0

sem fim serão utilizadas em avaliações posteriores dos classificadores que as identificaram (modificados apenas pela inserção de algumas novas instâncias de treinamento). Assim, para que a avaliação reflita um resultado mais justo, todos os experimentos foram realizados sobre o mesmo conjunto de teste (3/10 do corpora RST).

Tabela 6.3: **Agrupamento das 29 relações retóricas do nível inter-sentencial disponíveis nos corpora RST para o Português em 17 grupos e suas frequências.**

Relação	**Frequência**
Attribution	8
Antithesis Concession Contrast	227
Background Circumstance	264
Volitional-Result Non-Volitional-Result Volitional-Cause Non-Volitional-Cause	235
Comparison	24
Condition Otherwise	6
Elaboration	1967
Enablement Motivation Purpose	77
Evidence Justify Explanation	388
Interpretation Evaluation Conclusion	233
Solutionhood	70
List	802
Joint	13
Means	30
Restatement	32
Sequence	289
Summary	3
Same-unit	0

6.1.1 Adaptação do *parser* SPADE

Uma adaptação da metodologia utilizada por Soricut & Marcu (2003) foi realizada para o Português (aqui chamada de SPADE-PT), utilizando o córpus CSTNews (Cardoso *et al*, 2011a) para a extração das probabilidades e geração do modelo de identificação de relações retóricas.

A diferença entre Pr e Ps (equações 3.1 e 3.2 da página 36) é que ambas as probabilidades têm a mesma estrutura, porém, Ps é sem lexicalização, sendo um conjunto menos esparso que Pr, pois se tem menos etiquetas sintáticas que possíveis palavras da língua Portuguese, por exemplo. Nessa adaptação, isso aconteceu, diferentemente de Soricut e Marcu, pois se utilizou o modelo apenas para a identificação da relação retórica e sua nuclearidade entre dois segmentos intrassentenciais.

O modelo de Soricut e Marcu utiliza árvores sintáticas lexicalizadas de cada sentença e, nesse trabalho de adaptação, todos os textos dos corpora foram analisados pelo *parser* sintático LX-Parser (Silva *et al*, 2010). Como já dito, uma árvore sintática lexicalizada contém, nos nós interiores, a indicação de palavras junto às etiquetas sintáticas. A lexicalização é realizada fazendo-se a projeção, a partir das folhas, das palavras nucleares presentes em cada ramificação da árvore. Por exemplo, em um sintagma nominal "o menino", a palavra nuclear é "menino". Essa palavra será projetada para o nó superior na árvore sintática, causando sua lexicalização.

Similarmente ao trabalho original, a projeção de núcleos foi feita pela definição de listas de prioridades para cada possível rótulo de nós internos da árvore sintática. O processo de lexicalização é ascendente (*bottom-up*). Assim, inicia-se avaliando os nós folha. Posteriormente, para cada nó interno, seus filhos, já lexicalizados, são avaliados para fazer a projeção de um núcleo. A avaliação é feita como uso de listas de etiquetas sintáticas cuja ordem indica a prioridade de cada etiqueta na projeção. Para a geração da lista de etiquetas, foi tomado como base o trabalho de Hernault *et al* (2010), e as etiquetas foram mapeadas para as do *parser* utilizado nesta pesquisa.

Por exemplo, considere a lista de prioridades para um nó interno de um sintagma nominal (*NP - noun phrase*), que é exibida a seguir:

priorityNP=(N, NP, V, VP, CONJ, CONJP, ADV, ADVP, REL, P, PP, PPA, PNT, S, SNS, A, AP, C, CP, QNT, CARD, O, CL, ART, D, DEM, POSS, PRS)

Assim, um sintagma nominal (NP) formado por duas palavras (O) (menino), com respectivas etiquetas (ART) e (N), terá a palavra (menino) projetada pois (N) tem maior prioridade que (ART) na lista de prioridades de (NP).

Esse processo é feito para todos os nós internos da árvore sintática, permitindo a extração das informações necessárias às equações citadas e consequente definição das probabilidades do *parser*.

O LX-Parser foi escolhido na adaptação, pois gera árvores sintáticas similares às do *parser* sintático de Charniak (Charniak, 2000), facilitando a aplicação das regras canônicas de projeção de núcleos (Magerman, 1995), a fim de gerar a árvore sintática lexicalizada.

Em suma, a adaptação do *parser* SPADE foi feita pelos seguintes passos:

1. Obter a árvore sintática para cada sentença do córpus RST;

2. Lexicalizar as árvores sintáticas;

3. Mapear os segmentos discursivos às suas respectivas árvores sintáticas lexicalizas;

4. Extrair as probabilidades para cada par de segmentos intrassentenciais.

A aplicação do SPADE-PT se dá pela consulta às probabilidades geradas, utilizando as informações extraídas das árvores sintáticas lexicalizadas de um par de segmentos que se deseja analisar, obtendo a relação retórica mais provável.

Na avaliação, os resultados obtidos nesta adaptação foram muito baixos, possivelmente devido à quantidade de dados. Então, procedeu-se à adaptação da metodologia utilizada no *parser* HILDA (Hernault *et al*, 2010), tendo em vista que nele se expande o conjunto de atributos utilizados pelo SPADE.

6.1.2 Adaptação do *parser* HILDA

A metodologia empregada por Hernault *et al* (2010) para identificar as relações discursivas foi adaptada ao Português (chamada de HILDA-PT), dado que este *parser* era o estado

da arte durante os primeiros anos desta pesquisa.

O primeiro grupo de atributos usa informações sobre contagem de tokens e UBDs, tais como distâncias (em número de tokens e UBDs) do segmento ao início da sentença e do texto. O segundo grupo (relacionado à ideia introduzida por Soricut & Marcu (2003), a saber, os conjuntos de dominância) usa diversas informações da árvore sintática lexicalizada, nos pontos de ligação entre as UBDs. O escopo de cada atributo pode ser cada *span* (constituído por uma ou mais UBDs, em uma mesma sentença) ou par de *spans*.

Hernault *et al* (2010) utilizaram classificadores SVM para identificar as relações discursivas do texto, mas, durante os experimentos de adaptação, verificou-se que o algoritmo de árvores de decisão (Quinlan, 1993) mostrou-se melhor, além de fornecer um valor de confiança da classificação, que será útil à abordagem semissupervisionada.

Como o valor de diversos atributos são textuais, uma conversão para valores numéricos foi realizada (em que cada palavra do texto torna-se um atributo binário, em que 1 representa a presença da palavra e 0, sua ausência) e algumas generalizações foram feitas. Por exemplo, valores textuais como números, símbolos e pontuação foram substituídos por rótulos genéricos antes da conversão para números (NUM para números, SYM para símbolos e PUNC para pontuações) como o objetivo de reduzir o tamanha do vetor de palavras, na conversão.

Para a adaptação, os atributos utilizados no HILDA (Tabela 3.1, da página 43) foram adaptados para o Português. Para a extração do primeiro grupo de atributos (organização textual) não foi necessária ferramenta ou recurso especial, dado que a identificação de tokens é obtida pela tokenização do texto. A identificação de UBDs, sentenças e parágrafos já está feita na anotação do córpus.

Para o segundo grupo (sintáticos), as árvores sintáticas foram obtidas pelo uso do *parser* LX-Parser. A lexicalização das árvores sintáticas foi gerada pela adaptação das regras canônicas de projeção de núcleos lexicais de Magerman (1995), com o uso de listas de prioridades, como já explicado anteriormente. Para a adaptação dessas regras, foi realizado um mapeamento entre as etiquetas do *parser* utilizado pelo HILDA para as etiquetas do LX-parser.

Feita a extração de atributos, o ambiente Weka (Witten & Frank, 2005) foi utilizado para realizar os experimentos e obter os resultados dos classificadores. A versão utilizada foi a 3.6.11 e, para cada algoritmo aplicado (J48, NaiveBayes,), seus parâmetros padrões foram mantidos. Os resultados serão sumarizados ao final deste capítulo.

6.1.3 Novos atributos

A Tabela 3.1 contém atributos do nível textual (sem necessidade de conhecimento linguístico) e do nível sintático (árvore sintática lexicalizada), mas não explora outros níveis linguísticos ou mesmo outros atributos sintáticos. Desta forma, com o objetivo de explorar ao máximo o aprendizado supervisionado com novos atributos, foram propostos vários atributos dos diversos níveis linguísticos. Esses atributos estão apresentados nas Tabelas 6.4, 6.5, 6.6 e 6.7, indicando o escopo de acordo com o nível linguístico a que pertencem. Para sumarizar todos os atributos utilizados nesta pesquisa para o nível intrassentencial, nas tabelas foram incluídos os atributos do *parser* HILDA.

No nível da organização textual (Tabela 6.4), foram propostos atributos (os atributos 1 a 13 são provindos do *parser* HILDA) para capturar os tokens dos limites dos segmentos (que podem incluir pontuações). Assim, os atributos 14 a 18 utilizam os trigramas iniciais e finais de cada segmento. Além do uso explícito de uma lista de marcadores discursivos (Apêndice B), os tokens podem ser indicativos no caso de relações discursivas não marcadas explicitamente por marcadores discursivos. Nesse nível, também se utilizou a verificação se o segmento é terminado por uma interrogação ou exclamação (atributos 20 e 21), buscando correlacionar tais fenômenos com a ocorrência de relações discursivas.

Um nível que não foi explorado no HILDA é a morfossintaxe (Tabela 6.5), que mostrou-se muito útil à identificação das relações discursivas, conforme os resultados que serão apresentados posteriormente. Nesse nível, identificaram-se a primeira e última conjunções coordenativa ou subordinativa e seus respectivos subtipos (atributos 22 e 23). Essas conjunções podem ter sobreposição com marcadores discursivos (vide Apêndice B).

Além das conjunções, foram feitas contagens de verbos, substantivos, advérbios e adjetivos (consideradas palavras de conteúdo) e conjunções. Esses números foram divididos

Tabela 6.4: **Atributos da organização textual.**

Índice	**Atributo**	**Escopo**
	Organização Textual	
1	Mesma sentença	Par de *spans*
2	Mesmo parágrafo	Par de *spans*
3	Número de limites de sentenças	*span*
4	Número de limites de parágrafos	*span*
5	Tamanho em tokens	*span*
6	Tamanho em UBDs	*span*
7	Distância ao início da sentença em tokens	*span*
8	Tamanho do segmento sobre tamanho da sentença em tokens	*span*
9	Tamanho do segmento sobre tamanho da sentença em UBDs	*span*
10	Tamanho do par de segmentos sobre tamanho da sentença em UBDs	Par de *spans*
11	Distância ao início da sentença em UBDs	*span*
12	Distância ao início do texto em tokens	*span*
13	Distância ao fim da sentença em tokens	*span*
14	Primeiro token	*span*
15	Segundo token	*span*
16	Terceiro token	*span*
17	Último token	*span*
18	Penúltimo token	*span*
19	Ante-penúltimo token	*span*
20	Segmento interrogativo	*span*
21	Segmento exclamativo	*span*

pela quantidade de tokens do segmento, para obtenção de um número proporcional e não absoluto (atributos 24 a 29). Somou-se todas as palavras de conteúdo e dividiu-se pela quantidade de tokens do segmento. As proporções de palavras de conteúdo podem indicar o relacionamento entre os conteúdos dos segmentos, como uma elaboração (substantivos e adjetivos) e cirscunstância (advérbios), por exemplo .

Para a extração dos atributos nesse nível, o *parser* Palavras (Bick, 2000) foi utilizado, dado que outras informações, como entidades nomeadas, serão utilizadas posteriormente.

Tabela 6.5: **Atributos da morfossintaxe.**

Índice	**Atributo**	**Escopo**
	Morfossintaxe	
22	Tipo da primeira conjunção	*span*
23	Tipo da última conjunção	*span*
24	Número de verbos sobre tamanho do segmento em tokens	*span*
25	Número de substantivos sobre tamanho do segmento em tokens	*span*
26	Número de advérbios sobre tamanho do segmento em tokens	*span*
27	Número de adjetivos sobre tamanho do segmento em tokens	*span*
28	Número de conjunções sobre tamanho do segmento em tokens	*span*
29	Número de palavras de conteúdo sobre tamanho do segmento em tokens	*span*

No nível sintático (Tabela 6.6), foram utilizados os atributos relacionados aos conjuntos

de dominância, utilizados no *parser* HILDA. Para a obtenção da árvore sintática, foi utilizado o *parser* LX-Parser(Silva *et al*, 2010), dado que a estrutura arbórea de sua análise sintática é mais similar as dos *parsers* utilizados pelos autores de SPADE e HILDA.

Tabela 6.6: **Atributos da sintaxe.**

Índice	**Atributo**	**Escopo**
	Sintáxe	
30	Distância à raiz da árvore sintática	*span*
31	Distância ao ancestral comum na árvore sintática	*span*
32	Média das distâncias ao ancestral comum	Par de *spans*
33	Cabeça lexical do nó dominante do segmento	*span*
34	Etiqueta morfossintática do ancestral comum	Par de *spans*
35	Cabeça lexical do ancestral comum	Par de *spans*
36	Etiqueta morfossintática do nó dominante do segmento	Par de *spans*
37	Cabeça lexical do nó dominante do segmento	Par de *spans*
38	Etiqueta do nó dominante do segmento irmão	Par de *spans*
39	Cabeça lexical do nó dominante do segmento irmão	Par de *spans*

Na semântica (Tabela 6.7), utilizou-se um thesaurus para o Português (TeP2 - Maziero *et al* (2008)) para identificar os sinônimos entre verbos e substantivos de cada par de segmentos (atributos 40 e 41), assim como a antonímia entre verbos, substantivos, advérbios e adjetivos entre cada par de segmentos (atributos 42 a 45). O motivo de identificar antonímia entre advérbios e adjetivos é buscar identificar relações de contradição, como *Antithesis* e *Contrast*, por exemplo.

Nesse nível, utilizaram-se informações de entidades nomeadas provindas do *parser* Palavras. O atributo 46 faz a contagem da quantidade de entidades nomeadas que está presente em cada segmento e o atributo 47 verifica a quantidade de entidades nomeadas em comum, que pode indicar se uma mesma entidade está sendo tratada no par de segmentos.

As cadeias lexicais foram utilizadas de forma semelhante às entidades nomeadas. O atributo 48 faz a contagem da quantidade em cada segmento e o atributo 49 indica a sobreposição de cadeias lexicais no par de segmentos. Para identificação das cadeias lexicais foi utilizada a ferramenta desenvolvida neste trabalho e descrita no Capítulo 3 (página 70).

No nível discursivo, uma lista de marcadores discursivos foi utilizada com o intuito de identificar as relações marcadas explicitamente. Por exemplo, um segmento que se inicia com um marcador "mas", por exemplo, pode indicar a presença das relações *Antithesis*,

Tabela 6.7: **Atributos da semântica e discurso.**

Índice	**Atributo**	**Escopo**
	Semântica	
40	Número de verbos sinônimos	Par de *spans*
41	Número de substantivos sinônimos	Par de *spans*
42	Número de verbos antônimos	Par de *spans*
43	Número de substantivos antônimos	Par de *spans*
44	Número de advérbios antônimos	Par de *spans*
45	Número de adjetivos antônimos	Par de *spans*
46	Número de entidades nomeadas	*span*
47	Número de entidades nomeadas em comum	Par de *spans*
48	Número de cadeias lexicais	*span*
49	Número de cadeias lexicais em comum	Par de *spans*
	Discurso	
50	Primeiro marcador discursivo	*span*

Contrast ou *Concession*.

Foram realizados diversos experimentos com os atributos indicados nas tabelas acima. Tais experimentos possibilitaram a escolha do melhor conjunto de atributos para proceder ao uso do aprendizado semissupervisionado sem fim.

6.1.4 Resultados

Nesta seção, são apresentados os resultados obtidos na adaptação das metodologias anteriormente descritas e na incorporação dos atributos propostos. Todos os resultados foram obtidos do conjunto de teste dos corpora RST (que é um terço aleatório do corpora utilizado). Dois *baselines* foram utilizados: Relação *Elaboration* e DiZer (Pardo, 2005). O *baseline* Relação *Elaboration* consiste na identificação constante da relação nomeada, dado que ela é a mais frequente nos corpora RST. O *baseline* DiZer consiste na aplicação do módulo de identificação de relações do *parser* DiZer, que utiliza padrões lexicais nesse processo.

SPADE-PT obteve uma medida-F baixa de 0,35, dado que apenas 18% do conjunto de teste foi classificado com alguma relação, dada a esparsidade do modelo gerado. A precisão do modelo ficou em 0,53, mas a cobertura em apenas 0,26. Nessa avaliação levou-se em consideração apenas as relações que puderam ser inferidas pelo SPADE-PT. Assim, na ocorrência de um conjunto de dominâncias (no teste) que não ocorreu no conjunto de

treinamento, os segmentos envolvidos não são rotulados.

O *baseline* Relação *Elaboration* obteve medida-F de 0,26 (que é a porcentagem de relações *Elaboration* no conjunto de teste). Já o *baseline* DiZer obteve medida-F de 0,22, com precisão foi 0,61, mas cobertura muito baixa, 0,14. Esses resultados foram baixos, pois os padrões lexicais utilizados pelo DiZer foram extraídos de um corpus acadêmico, e nos experimentos foram utilizados mais textos jornalísticos que acadêmicos, além de que muitas relações discursivas ocorrem sem marcação explícita e, portanto, não podem ser detectadas por padrões lexicais.

Uma das razões para a baixa medida-F do SPADE-PT é a sobreposição de informações no modelo gerado. Por exemplo, considere o seguinte conjunto de dominância: $(2, CONJP, e) \prec (2, CONJP, ou)$. Este conjunto é utilizado para identificar tanto a relação *Restatement* quanto o grupo formado pelas relações *Interpretation*, *Evaluation* e *Conclusion*. O uso de um conjunto de atributos maior e mais diversificado (Tabela 3.1, página 43) tratou essa questão e melhorou a medida-F na tarefa de identificação das relações retóricas. O HILDA-PT obteve performance melhor que SPADE-PT e *baselines*, atingindo medida-F de 0,57. Os resultados das metodologidas adaptadas está apresentado na Tabela 6.8.

Serão reportadas duas medidas, a saber, medida-F e acurácia. A medida-F é uma média harmônica de precisão (P) e cobertura (C) ($medidaF = \frac{2 \times P \times C}{P+C}$).

Tabela 6.8: **Resultados das adaptações (SPADE-PT e HILDA-PT) em comparação com os *baselines* (Relação *Elaboration* e DiZer) .**

Método	**Precisão**	**Cobertura**	**Medida-F**
SPADE-PT	0,53	0,26	0,35
HILDA-PT	**0,58**	**0,59**	**0,57**
Relação *Elaboration*	0,26	0,26	0,26
DiZer	0,61	0,14	0,22

O HILDA-PT obteve performance muito superior ao SPADE-PT, a Tabela 6.9 apresenta os resultados da adaptação na identificação de cada relação. Algumas relações obtiveram boas medidas-F, acima de 0,6, tais como *Elaboration*, *Attribution*, *Enablement-Motivation-Purpose* e *Same-unit*. Essas relações apresentam frequência relativamente alta (Tabela 6.2), respectivamente: 1491, 799, 695 e 731. As relações *Interpretation-*

Evaluation-Conclusion e *Summary* obtiveram resultados nulos, pois o primeiro grupo de relações contém apenas 40 instâncias nos corpora RST. Interessante notar que *Comparison*, embora tenha obtido baixa medida-F (0,333), tem apenas 37 instâncias nos corpora, e *Means*, embora tenha 73 instâncias nos corpora, obteve medida-F baixa de 0,111, devido à cobertura de 0,067.

Tabela 6.9: **Medida-F para cada relação tratada pelo HILDA-PT, considerando o agrupamento de relações (Tabela 6.2). A medida-F foi ponderada de acordo com a frequência de cada relação/grupo de relações no conjunto de teste.**

Relação	**Precisão**	**Cobertura**	**Medida-F**
Attribution	0,667	0,657	0,662
Antithesis Concession Contrast	0,567	0,548	0,557
Background Circumstance	0,421	0,480	0,449
Volitional-Result Non-Volitional-Result Volitional-Cause Non-Volitional-Cause	0,460	0,343	0,393
Comparison	0,500	0,250	0,333
Condition Otherwise	0,750	0,375	0,500
Elaboration	0,648	0,770	0,704
Enablement Motivation Purpose	0,634	0,717	0,673
Evidence Justify Explanation	0,611	0,367	0,458
Interpretation Evaluation Conclusion	0,000	0,000	0,000
List	0,481	0,578	0,525
Means	0,333	0,067	0,111
Restatement	0,429	0,750	0,545
Same-unit	0,661	0,667	0,664
Sequence	0,467	0,163	0,241
Summary	0,000	0,000	0,000
All relations	0,579	0,593	**0,576**

Considere a matriz de confusão, em que os rótulos das linhas e colunas são como a seguir: a) = *Antithesis-Concession-Contrast* b) = *Background-Circumstance* c) = *Interpretation-Evaluation-Conclusion* d) = *Evidence-Justify-Explanation* e) = *Enablement-Motivation-Purpose* f) = *Condition-Otherwise* g) = *Cause-Result* h) = *Attribution* i) = *Comparison*

j) = *Elaboration* k) = *Restatement* l) = *Same-unit* m) = *Sequence* n) = *List* o) = *Summary* p) = *Means* q) = *Joint* r) = *Solutionhood*. É interessante notar que a coluna de j) tem muitos erros, pois diversas relações são identificadas erroneamente como *Elaboration*. O mesmo ocorre com a coluna l) da relação *Same-unit* e *List*.

Tabela 6.10: **Matriz de confusão do HILDA-PT**

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	← Predito
17	1	0	0	3	0	1	1	0	3	0	2	0	3	0	0	0	0	a
4	**24**	0	1	2	0	3	2	0	9	0	3	1	1	0	0	0	0	b
0	0	**0**	0	0	0	2	0	0	3	0	0	0	1	0	0	0	0	c
0	3	0	**11**	1	0	4	0	0	9	0	0	0	2	0	0	0	0	d
1	2	0	1	**71**	0	2	0	0	15	0	5	0	1	0	1	0	0	e
1	1	0	0	0	**3**	1	0	0	1	0	1	0	0	0	0	0	0	f
0	5	0	3	5	1	**23**	0	0	12	0	5	0	13	0	0	0	0	g
0	0	0	0	5	0	2	**44**	0	9	0	3	0	4	0	0	0	0	h
0	0	0	0	0	0	0	0	**1**	3	0	0	0	0	0	0	0	0	i
2	6	0	0	9	0	3	9	1	**171**	3	11	1	5	0	1	0	0	j
0	1	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	k
0	4	0	1	7	0	2	6	0	10	0	**78**	0	9	0	0	0	0	l
1	5	0	0	2	0	1	1	0	5	0	4	**7**	17	0	0	0	0	m
3	4	0	1	4	0	4	1	0	8	1	6	6	**52**	0	0	0	0	n
0	0	0	0	0	0	0	0	0	1	0	0	0	0	**0**	0	0	0	o
1	1	0	0	3	0	2	2	0	5	0	0	0	0	0	**1**	0	0	p
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	q
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	r

A seguir, serão relatados diversos experimentos realizados com os atributos propostos, adicionados aos atributos da metodologia empregada no HILDA-PT. Nas Tabelas 6.11 e 6.12, foram explorados diversos conjuntos de atributos agrupados de acordo com o nível linguístico dos mesmos (Tabelas 6.4, 6.5, 6.6, 6.7), a saber, OT para novos atributos de organização textual, M para morfossintáticos, S para semânticos e D para discursivos. Os atributos sintáticos já estão todos considerados no *parser* HILDA.

No nível intrassentencial (Tabela 6.11), a melhor combinação de atributos foi utilizar os atributos do HILDA-PT em conjunto com os morfossintáticos (Tabela 6.5), obtendo medida-F de 0,604 e acurácia de 0,623. Já o uso de todos atributos (HILDA-PT + OT + M + S + D) levou a uma medida-F de 0,595, pouco inferior à combinação (HILDA-PT + M). Desta forma, para a exploração do aprendizado semissupervisionado sem fim, apresentado no próximo capítulo, foi considerada essa última combinação de atributos.

A medida-F obteve aumento na seguinte ordem de acréscimo de atributos: OT, S, M + D, M + D + S, OT + M + D + S, M + S e M, sendo a última combinação a melhor,

Tabela 6.11: **Resultados dos experimentos com aprendizado supervisionado para a identificação de relações discursivas no nível intrassentencial, considerando diversos grupos de atributos, em que OT é Organização Textual, M é Morfossintaxe, S é Semântica e D é Discurso.**

Atributos	**Medida-F**	**Acurácia**
HILDA-PT	0,576	0,592
HILDA-PT + OT	0,578	0,589
HILDA-PT + S	0,579	0,597
HILDA-PT + M + D	0,580	0,602
HILDA-PT + M + D + S	0,585	0,604
HILDA-PT + OT + M + D + S	0,595	0,602
HILDA-PT + M + S	0,598	0,617
HILDA-PT + M	**0,604**	**0,623**
Relação *Elaboration*	0,260	0,260

como já dito.

Tabela 6.12: **Resultados dos experimentos com aprendizado supervisionado para a identificação de relações discursivas no nível inter-sentencial, considerando diversos grupos de atributos, em que M é Morfossintaxe, S é Semântica e D é Discurso .**

Atributos	**Medida-F**	**Acurácia**
HILDA-PT	0,236	0,408
HILDA-PT + M	0,260	0,420
HILDA-PT + S	0,246	0,413
HILDA-PT + D	0,260	0.,11
HILDA-PT + M + S	0,278	0,423
HILDA-PT + M + D	0,281	0,420
HILDA-PT + S + D	0,280	0,420
HILDA-PT + M + S + D (SVM)	**0,289**	**0,425**
HILDA-PT + M + S + D (J48)	**0.,27**	**0,338**
HILDA-PT + M + S + D (Adaboost com J48)	**0,362**	**0,400**
Relação *Elaboration*	0,421	0,421

Na Tabela 6.12, é importante notar que, dado o desbalanceamento das relações, quando utilizado o conjunto de atributos do HILDA-PT, são identificadas corretamente apenas as relações *Elaboration*, que é a relação majoritária. No nível inter-sentencial, a medida-F foi baixa dado que a dificuldade aumenta, pois a maior parte das pistas textuais são deixadas entre as UBDs em nível intrassentencial. Além disso, quando se vai relacionar dois *spans* que consistem de diversas UBDs, deve-se escolher uma UBD de cada *span* para definir uma relação. Por exemplo, sendo A,B,C,D sentenças com árvores discursivas para cada uma delas, os relacionamentos entre essas sentenças deve levar em consideração todas as UBDs da sentença ou apenas algumas delas? No caso de relacionar, por exemplo, *spans*

formados por AB e CD, deve-se levar em consideração qual parte do *span* AB e qual do CD para a escolha da relação retórica?

A melhor configuração de atributos para a identificação das relações no nível inter-sentencial foi obtida com a combinação HILDA-PT + M + S + D. Além dessa combinação, foram testadas mais dois algoritmos de aprendizado de máquina, além do SVM: J48 (Quinlan, 1993) e Adaboost (Freund & Schapire, 1996) com J48. O Adaboost (*Adaptive Boosting*) é um meta algoritmo que combina as saídas de um algoritmo base, neste caso, o J48, para formar a saída do classificador. Como pode ser visto, essa técnica melhorou a performance na tarefa inter-sentencial.

6.2 Aprendizado semissupervisionado sem fim

Com base nas possibilidades advindas da grande quantidade de dados disponíveis na web e diante das limitações atuais na metodologia de análise discursiva para a língua portuguesa (abordagem única por padrões lexicais gerados manualmente e quantidade insuficiente de dados para um bom aprendizado automático), nesta pesquisa, foi investigado o uso massivo de dados aliado a técnicas de aprendizado de máquina semissupervisionadas para a tarefa de análise discursiva, especificamente, na etapa de identificação das relações discursivas.

Segundo Banko & Brill (2001), vale a pena investir em córpus em detrimento de se investir em diferentes algoritmos para o aprendizado. Neste caso, foi utilizada a técnica de árvores de decisão (J48), que tem sido amplamente empregada em diversos cenários, inclusive em muitas tarefas de PLN, por se tratar de uma técnica que permite um entendimento das decisões internas do modelo, pois cria uma árvore de decisão com os atributos do aprendizado. Além disso, nos experimentos realizados com outras técnicas (incluindo *Naive Bayes* e SVM), o J48 teve um bom comportamento, inclusive, no fornecimento da confiabilidade de suas classificações e resultados obtidos.

Na linha semissupervisionada, acredita-se que a grande quantidade de dados disponíveis na web pode ser incorporada ao aprendizado. Para isso, partiu-se dos corpora

anotado manualmente e fez-se uso de métodos de aprendizado iterativo, semelhante ao *self-training* e ao *co-training*.

A Figura 6.2 exemplifica a abordagem utilizada: os classificadores são aprendidos, inicialmente, a partir dos corpora RST anotados manualmente (descritos no Capítulo 3) e são utilizados na classificação de exemplos de pares de segmentos discursivos advindos da web através do web crawler desenvolvido.

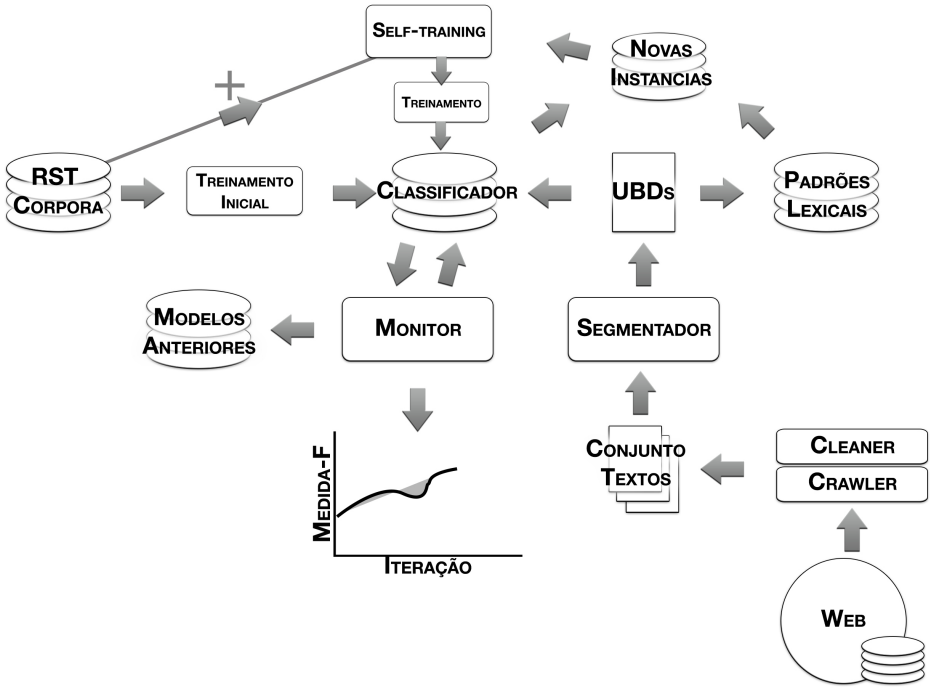


Figura 6.2: **Esquema do aprendizado semissupervisionado explorado.**

Como cada classificação é dada com um valor ou probabilidade de confiança, os exemplos que forem classificados com um valor de confiança maior que determinado limiar são armazenados para re-treino dos classificadores, juntamente com os dados advindos dos corpora inicial.

Desta forma, o ciclo de treino, classificação e re-treino é iterado (em lotes), analisando-se a curva de aprendizado dos classificadores. Quando se verifica decréscimo no aprendizado, o atual classificador (e as instâncias classificadas automaticamente e utilizadas em seu treinamento) é dispensado e o melhor classificador anterior é resgatado para continuar a semissupervisão.

Pensando em grande quantidade de dados, o uso de aprendizado incremental, como

uma versão incremental das árvores de decisão conhecida como árvores de *Hoeffding* (Hulten *et al*, 2011), gera modelos cuja acurácia é inferior às árvores de decisão que são obtidas com todo o conjunto de treinamento de uma vez. Num aprendizado incremental, os dados de treinamento são vistos em lotes e o modelo aprendido é alterado a cada novo lote utilizado.

Nesta pesquisa, foi tomada a decisão de não utilizar o aprendizado incremental, o que possivelmente levou a um maior tempo de execução do *framework*. Essa escolha foi feita visando a busca da melhor performance do aprendizado.

Além disso, quanto maior for o número de classes na tarefa de aprendizado, maior o desafio da mesma. Pensando nisso e tendo em vista que as relações *Parenthetical* e *Attribution* são facilmente identificadas pela superfície textual (*Parenthetical*, pela ocorrência de texto entre parênteses, colchetes, chaves, etc.) ou estrutura sintática lexicalizada (*Attribution*, pela ocorrência de algum verbo de atribuição em um contexto definido, a ser descrito mais adiante), essas relações foram dispensadas do aprendizado automático, pois regras criadas manualmente foram utilizadas para as identificar. Isso causou a diminuição do número de classes, melhorando os resultados dos modelos provindos do aprendizado automático.

6.2.1 Arquitetura do *framework* de Semissupervisão Sem Fim

O Algoritmo 7 apresenta a ideia central desta pesquisa, em que uma abordagem de semissupervisão é utilizada, similarmente à abordagem dada ao aprendizado por Carlson *et al* (2010), dando origem ao *framework* de aprendizado semissupervisionado sem fim (daqui para a frente referenciado como *framework* SSNEL) A metodologia desta pesquisa consiste em utilizar dois modelos de classificação, a saber, um módulo de padrões lexicais (*LPS - Lexical Patterns Set*, do *parser* discursivo DiZer), e um classificador multi-rótulo C gerado com o melhor conjunto de atributos explorados anteriormente.

Aqui cabe uma diferenciação com a abordagem de Carlson *et al* (2010): no aprendizado sem fim empregado pelos citados autores, as informações aprendidas são revistas, com a finalidade de readeaquer ou melhorar o conhecimento adquirido. Nesta pesquisa no

entanto, o aprendizado adquirido não é revisitado, podendo-se assemelhar a um aprendizado baseado em fluxo de dados (Oliveira, 2015), com a exceção de não ser um aprendizado incremental.

Por ser um aprendizado realizado continuamente (com infinitas iterações da semissupervisão), um web crawler busca constantemente textos na web para suprir a demanda por dados não rotulados. Esses textos são utilizados no *framework* em lotes de cinco textos. Isso foi feito para que se pudesse ter um controle maior sobre as instâncias que podem ou não contribuir para a melhoria do aprendizado. Se um número maior de textos fosse utilizado, a cada iteração haveria um número muito grande de novas instâncias.

6.2.1.1 Aprendizado Automático

Como técnica de semissupervisão, foi utilizada uma abordagem entre o *selft-training* e o *co-training*. Como o módulo LPS não fornece valor de confiabilidade para as instâncias que rotula, essas instâncias são sempre incorporadas ao conjunto de treinamento para re-treino do classificador C. Já as instâncias classificadas por C vêm com um valor de confiança entre 0 e 1, e são filtradas, de acordo com a confiabilidade retornada. Outra diferença é que o módulo LPS não sofre re-treino, como acontece com C.

Inicialmente, em cada iteração i, o texto é segmentado e, para cada par de UBDs adjacentes, aplica-se LPS e C_i. As instâncias rotuladas por C_i são filtradas de acordo com a confiabilidade retornada e adicionadas ao conjunto $dadosRotulados_i$. As instâncias rotuladas por LPS também são adicionadas ao conjunto $dadosRotulados_i$, que será utilizado no re-treino.

Para a filtragem das novas instâncias rotuladas automaticamente, foram experimentados diversos limites (*thresholds*) numa escala de 0 a 1: $0,3$, $0,5$, $0,6$, $0,7$ e $0,8$. Além do uso do valor de confiabilidade da classificação, dois outros experimentos foram realizados, um selecionando, a cada iteração, instâncias das cinco relações menos frequentes do conjunto de treinamento, e outro, utilizando apenas os classificadores C_i, sem contar com as classificações do módulo LPS. Ambos experimentos utilizaram $threshold > 0,7$.

Desta forma, quando um novo classificador C_{i+1} é treinado com $dadosRotulados_i$,

Algoritmo 7: Algoritmo do SSNEL utilizando dois modelos de identificação de relações retóricas entre cada $PairUBDs$.

```
Entrada: dadosRotulados_1 e texto
início
    treinar um classificador C_1 usando dadosRotulados_1
    Index ← 1

    repita
        obter um texto de conjuntoTextos
        aplicar Segmenter em texto para obter textoSegmentado
        obter PairUBDs de textoSegmentado

        para PairUBDs faça
            aplicar LPS para obter dadosRotuladosLPS
            aplicar C_Index para obter dadosRotuladosC

            para dadosRotuladosC como novaInstanciaC faça

                se confidence de novaInstanciaC ≥ threshold então
                    dadosRotuladosCconfident ←
                                novaInstanciaC
                fim
            fim
            dadosRotulados_Index+1 ←
                        dadosRotuladosLPS+
                        dadosRotuladosCconfident
            treinar um novo classificador C_Index+1
                        usando dadosRotulados_Index+1
            aplicar Monitor e obter
                        FmC_Index+1
            plotar FmC_Index+1 no gráfico G

            se FmC_index+1 < FmC_Index então
                descartar C_Index+1
                C_Index+1 ← C_Index
            fim
        fim
    até existe algum texto;
fim
```

aplica-se um módulo $Monitor$ para verificar se as novas instâncias, rotuladas automaticamente, contribuíram para o aprendizado na classificação das relações retóricas.

Como técnica de aprendizado para treinamento dos C_i, foi utilizado o algoritmo J48, que cria uma árvore de decisão. O J48 é uma implementação do algoritmo C4.5 (Quinlan,

1993). Esse algoritmo comportou-se melhor que o SVM com os atributos utilizados, além de retornar um valor de confiabilidade para cada classificação.

Em detrimento de explorar melhores técnicas de aprendizado, foi focado o uso de grande quantidade de dados não rotulados, que são descritos na próxima subseção.

6.2.1.2 A grande quantidade de dados

Em busca de textos não rotulados a serem utilizados no aprendizado semissupervisionado sem fim, o web crawler desenvolvido faz a busca a partir de uma lista (*seeds*) de jornais online. Conforme o Algoritmo 3 (página 73), a cada página visitada, seus *links* são visitados posteriormente. Tais *links* podem levar a páginas que não sejam do gênero jornalístico. Caso as instâncias geradas a partir de textos não jornalísticos cooperem para o aprendizado, elas são mantidas. Caso prejudiquem o aprendizado, são descartadas pelo módulo $Monitor$, como será visto à frente.

Uma dificuldade encontrada foi com relação à codificação dos textos. Algoritmos de conversão nem sempre são bem sucedidos. Assim, foi tomada a decisão de considerar apenas os textos em codificação UTF-8 e ISO-8859-1. Como as ferramentas e recursos trabalham com a codificação UTF-8, os textos que foram obtidos em ISO 8859-1 foram convertidos para UTF-8 (essa conversão foi testada e nenhum problema foi encontrado, como aconteceu com outras codificações). Os textos de outras codificações são desconsiderados pelo web crawler.

A seguir, as páginas obtidas pelo web crawler são processadas pela ferramenta NCleaner (Evert, 2008) para extração do texto principal, que geralmente corresponde à notícia da página. O NCleaner coloca algumas marcações de parágrafo, assim, um pós processamento foi realizado em que essas marcações são retiradas. Nesse pós processamento, alguns caracteres especiais são substituídos pelo respectivo caracter do alfabeto. Por exemplo, o caracter especial *é* é substituído por *é*. Ainda no pós processamento, o tamanho do texto retornado é verificado para evitar gerar arquivos de texto que sejam compostos por menos que uma sentença, como ocorre com páginas que não apresentam um texto principal (como uma página de propaganda).

Feita a extração do texto principal da página, esse texto é verificado ser, ou não, escrito em Português. Para isso, uma biblioteca para a linguagem Python, Guess-Language[1] foi utilizada. Essa ferramenta utiliza um modelo de n-gramas. Caso o texto esteja em outra língua, que não seja o Português, ele é dispensado.

Os textos obtidos são nomeados sequencialmente e armazenados em um diretório. Cada uma das instâncias do *framework* SSNEL utiliza os mesmos textos a cada iteração sem a necessidade de replicação dos mesmos.

A seguir, o módulo LPS é descrito, descrevendo os padrões lexicais utilizados, exemplificando sua aplicação.

6.2.1.3 Padrões Lexicais - LPS

O módulo LPS é utilizado dada a sua precisão (0,61), embora sua cobertura seja baixa (0,14). Isso é feito com vistas a passar o conhecimento dos padrões lexicais para o classificador C. Os padrões lexicais contêm diversos campos: i) relação retórica, ii) ordem da nuclearidade, iii) marcador 1, iv) posição do marcador 1, v) marcador 2 e vi) posição do marcador 2.

A relação retórica é uma das listadas no Apêndice A. A nuclearidade pode ser núcleo-satélite, satélite-núcleo ou núcleo-núcleo. Os marcadores 1 e 2 podem conter i) palavras flexionadas, ii) palavras lematizadas, iii) etiqueta morfossintática e iv) referência a uma lista de palavras (lista de verbos de atribuição, lista de pronomes pessoais, etc.). A posição dos marcadores podem ser i) início do segmento, ii) meio do segmento, iii) final do segmento e iv) qualquer posição.

Um padrão lexical para identificar o grupo de relações *Non-Volitional Cause*, *Non-Volitional Result*, *Volitional Cause* e *Volitional Result* contêm as informações esquematizadas na Tabela 6.13

No exemplo da Tabela 6.13, o marcador 2 contém referências a duas listas de palavras: *PronomesDemonstrativos* e *VerbosCausa*. Já não há exigências com relação ao marcador 1, que está vazio. Esse padrão lexical é aplicado a dois segmentos adjacentes no texto,

[1]https://pypi.python.org/pypi/guess-language

Tabela 6.13: **Exemplo de padrão lexical utilizado para identificação do grupo de relações de causa/resultado.**

Campo	Valor
Relação	Cause-Result
Nuclearidade	Satélite-Núcleo
Marcador 1	"Vazio"
Posição do marcador 1	Qualquer
Marcador 2	PronomesDemonstrativos VerbosCausa
Posição do marcador 2	Início

de forma que o primeiro marcador será aplicado ao primeiro segmento e o marcador 2 ao segundo segmento. A ocorrência, no início do segundo segmento, de casos como *Isso causou*, *Aquilo tornou* ou *Nisso resultou*, por exemplo, caracteriza uma relação *Cause-Result*, com o primeiro segmento como satélite e o segundo como nuclear.

Como é conhecido, há marcadores discursivos ambíguos, isto é, que indicam mais de uma relação. Considere outro padrão lexical, especificado na Tabela 6.14, ele pode ser aplicado ao par de segmentos a seguir:

1. O levantamento é indicativo do malabarismo imposto a os consumidores, que a cada dia têm de escolher o que vão comprar,

2. **pois** a renda é insuficiente para comprar tudo.

Isso faz com que seja incorporado ao aprendizado duas instâncias iguais que geram diferentes relações igualmente plausíveis, de acordo com o padrão encontrado.

Tabela 6.14: **Exemplo de padrão lexical utilizado para identificação do grupo de relações de causa/resultado (*Cause-Result*) e *Evidence-Justify-explanation*.**

Campo	Valor
Relação	*Cause-Result* ou *Evidence-Justify-explanation*
Nuclearidade	Satélite-Núcleo
Marcador 1	"Vazio"
Posição do marcador 1	Qualquer
Marcador 2	*pois*
Posição do marcador 2	Início

O módulo *LPS* conta com esses marcadores e, portanto, incorpora ambiguidade no aprendizado. Essa é uma das limitações do uso de padrões lexicais na identificação.

6.2.1.4 Regras manuais

Como já dito, algumas relações discursivas podem ser identificadas com o uso de regras manualmente criadas. Nesta pesquisa, duas relações (*Parenthetical* e *Attribution*) foram tratadas desta forma. As regras criadas fazem uso tanto das palavras flexionadas presentes no texto, quanto de suas formas lematizadas, pontuações e etiquetas morfossintáticas. Também se utiliza uma lista de verbos de atribuição em suas formas lematizadas (Apêndice B). Essa lista de verbos foi extraída do *parser* DiZer.

A seguir, as regras são esquematizadas nas tabelas 6.15 a 6.18.

Tabela 6.15: **Regra Attribution 1.**

1. Se $lema_i$ estiver presente em *lista de verbos de atribuição*
   Então $classe = Attribution$;

Tabela 6.16: **Regra Attribution 2.**

2. Se $lema_i$ for igual a ($conforme$ ou *segundo* ou *consoante*) e pos_i for preposição
   Então $classe = Attribution$;

Tabela 6.17: **Regra Attribution 3.**

3. Se $lema_i$ for igual a ($de\ \ acordo$) e pos_i for advérbio
   Então $classe = Attribution$;

Tabela 6.18: **Regra Parenthetical 4.**

4. Se primeiro caracter de $palavra_0$ for igual à abertura de parênteses
   Então $classe = Parenthetical$;

Considere, por exemplo, o par de segmentos:

1. **Segundo** Benjamin Gaston, um dos cientistas envolvidos na pesquisa,

2. a descoberta pode abrir alternativas para o tratamento de disfunções respiratórias.

A preposição *segundo* é encontrada no primeiro segmento, indicando a presença de uma relação *Attribution*.

Essas regras são aplicadas logo após a segmentação do texto para cada par de segmentos e, caso encontrem uma das duas relações, os segmentos não são analisados pelo classificador C.

6.2.1.5 Controle de mudança de conceito (*concept drift*)

Devido ao uso de grande quantidade de textos não rotulados, faz-se necessário um módulo $Monitor$ que verifique constantemente se o que está sendo rotulado automaticamente está contribuindo para o aprendizado.

O monitoramento realizado baseia-se na técnica *Statistical Process Control - SPC* (Gama *et al*, 2004), que monitora a evolução do erro à medida em que novos dados entram no aprendizado. O aumento do erro é tido como um indício de que o aprendizado está sendo degradado. Assim, caso a medida-F de C_{i+1} (FmC_{i+1}) seja menor que a medida-F de C (FmC_i), C_{i+1} é descartado e C_i é utilizado para rotular novas instâncias na iteração $i + 2$.

O módulo $Monitor$ também é responsável por gerar um gráfico de FmC_i por iteração i, o que deu origem à figuras da seção de resultados, a seguir.

Uma limitação dessa pesquisa está relacionada ao tamanho e representatividade do conjunto de teste. O $Monitor$ julgará baseado nesse conjunto que equivale a 3/10 dos corpora RST. A seguir, os resultados obtidos em experimentos do *framework* SSNEL são exibidos e discutidos.

6.2.2 Resultados

Uma questão crítica na avaliação do *framework* SSNEL está relacionada ao tamanho do conjunto de teste e à estratégia de avaliação. Abaixo são exibidos resultados para duas estratégias de avaliação: cruzada de 10 pastas e o uso de um conjunto de testes fixo de aproximadamente 3/10 dos corpora RST. A avaliação cruzada de 10 pastas foi mantida nessa seção com o intuito de apresentar a diferença dos resultados obtidos em contraste com o uso de uma avaliação com conjunto de teste fixo.

A avaliação cruzada de 10 pastas é uma estratégia de avaliação conveniente numa abordagem supervisionada, pois utiliza a média de 10 avaliações, em que os dados rotulados são segmentados em 10 pastas e a cada iteração, 9 pastas são utilizadas para treinamento e 1 para avaliação, refletindo um resultado mais representativo dos dados.

O uso dessa abordagem de avaliação em um aprendizado semissupervisionado sem fim pode levar a um resultado enviesado, pois, a cada nova iteração, novos dados rotulados automaticamente são incorporados aos dados que serão utilizados tanto no treinamento quanto no teste, dada a segmentação em 10 pastas. Assim, quanto maior o número de iterações, mais dados rotulados automaticamente serão considerados no conjunto de teste, indicando um resultado que pode não ser real, dado que instâncias utilizadas no treinamento podem ser utilizadas também na avaliação.

Essa é a motivação para o uso de um conjunto de testes fixo desde o início da semissupervisão, contendo apenas dados rotulados manualmente, com aproximadamente 30% dos corpora RST. Assim, o classificador C_1 é treinado, inicialmente, com aproximadamente 70% dos corpora RST. No decorrer das iterações, esse tamanho aumenta à medida com que novos dados são rotulados automaticamente. A desvantagem dessa abordagem de avaliação é que o resultado FmC_i (medida-F do classificador C_i, da iteração i) refletirá tanto quanto for a representatividade do conjunto de teste utilizado, que é fixo e não é incrementado a cada iteração i.

6.2.2.1 Validação cruzada de 10 pastas

Um experimento utilizando o *framework* SSNEL foi realizado considerando o conjunto de atributos da adaptação HILDA-PT, com $threshold = 0,7$ e validação estratificada cruzada de 10 pastas. A Figura 6.3 mostra o comportamento de FmC_i a cada iteração i. Na mesma figura é exibida uma outra curva que indica a melhor medida-F obtida até a iteração atual. Em aproximadamente 180 iterações, FmC foi de 0,55 a 0,79, um ganho de mais de 43% na medida-F inicial.

Esse rápido crescimento do resultado pode ser explicado pelo sobre-ajuste (*overfitting*) dos classificadores C_i aos dados de treinamento, que contêm dados rotulados automaticamente em iterações anteriores a i. Por exemplo, o classificador C_0 é utilizado para rotular diversos exemplos novos. Alguns desses são rotulados com $threshold \geq 0,7$ e são incorporados ao conjunto de treinamento para gerar o classificador C_1. Na próxima iteração, quando as pastas forem geradas para o treino e avaliação, elas conterão tantos os exemplos rotulados manualmente, quanto os gerados automaticamente, o que, possivelmente, levará a melhores resultados na avaliação, devido ao *overfitting* aos dados.

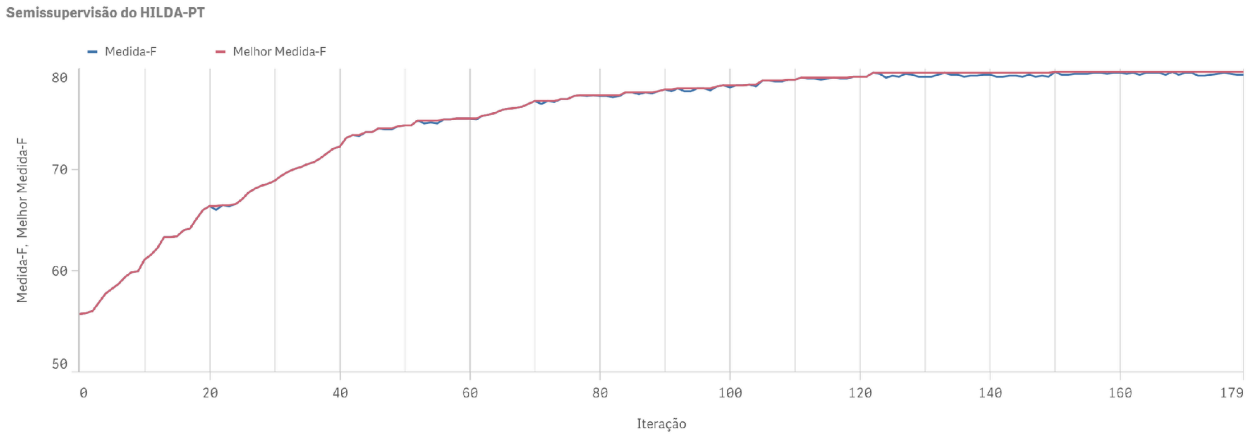


Figura 6.3: **Resultado do uso do *framework* de SSNEL para a adaptação HILDA-PT avaliado com validação estratificada cruzada de 10 pastas.**

Como será visto na subseção à frente, essa curva tem um comportamento diferente dos experimentos com um conjunto fixo de testes, que não está sujeito ao *overfitting*.

6.2.2.2 Conjunto de teste fixo

Para obter uma medida mais real do aprendizado no *framework* SSNEL, um conjunto de testes foi separado para uso exclusivo na avaliação dos C_i. Espera-se que os resultados obtidos nessa abordagem de teste não tenham sofrido de sobre-ajuste. No entanto, são tão representativos quanto o é o conjunto de testes.

Com a finalidade de identificar o melhor *threshold* de classificação para a semissupervisão, foram criadas diversas instâncias no *framework* SSNEL com diversos *thresholds*, a saber: $0,3$, $0,5$, $0,6$, $0,7$ e $0,8$. Além desses valores para o *threshold*, foram criadas duas instâncias que modificam alguns pontos do *framework* proposto: uma instância modificando o método de seleção de novos dados anotados de acordo com as relações menos frequentes nos dados de treinamento e outra que dispensa o uso do módulo LPS, utilizando apenas os classificadores C_i.

A seleção de novos dados de acordo com as relações menos frequentes é feita selecionando de $dadosRotuladosC_{i+1}$ as instâncias que pertencem às cinco classes menos frequentes na iteração i. A cada iteração é realizada uma nova contagem, para verificar as relações menos frequentes. Para não selecionar instâncias com baixa confiabilidade na classificação, foi utilizado também o *threshold* de 0,7. No caso do experimento utilizando apenas os classificadores C_i, o *threshold* utilizado também foi de 0,7. Esse experimento foi realizado com o intuito de melhorar a identificação das classes menos frequentes no corpora.

Para cada um dos sete experimentos citados anteriormente nesta seção, foi gerada uma figura indicando a curva de aprendizado (FmC_i por iteração i), possibilitando escolher pelo melhor comportamento de aprendizado, a fim de compor o módulo final de identificação de relações retóricas.

Após a apresentação das figuras, tem-se a Tabela 6.19. Essa tabela auxilia na interpretação das curvas, dado que apresenta o valor máximo de medida-F obtido em cada experimento e em qual iteração esse valor foi obtido. Em cada iteração, todos os experimentos utilizaram os mesmos textos, possibilitando uma comparação mais justa.

As Figuras 6.4, 6.5 e 6.6 apresentam as curvas de aprendizado geradas a partir de valo-

res relativamente baixos de confiabilidade nas classificações, respectivamente, $threshold > 0,3$, $threshold > 0,5$ e $threshold > 0,6$. Quando $threshold > 0,3$, em 831 iterações executadas, o valor máximo foi $FmC_{752} = 0,581$. Já para $threshold > 0,5$, em 817 iterações executadas, o valor máximo foi $FmC_{745} = 0,593$. Esse valor cai um pouco quando $threshold > 0,6$, atingindo no máximo $FmC_{802} = 0,585$ em 831 iterações executadas. Nas figuras que apresentam a medida-F pela iteração, é apresentada uma curva que representa a melhor medida-F obtida até a atual iteração, portanto, essa curva sempre fica acima da curva que representa a medida-F obtida na atual iteração (FmC_i).

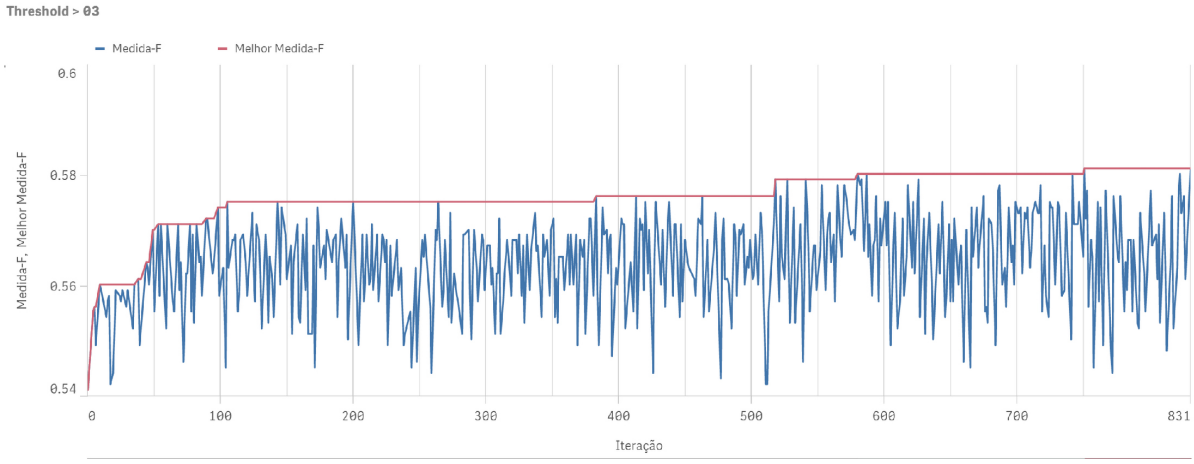


Figura 6.4: **Resultado para SSNEL com** $threshold > 0,3$

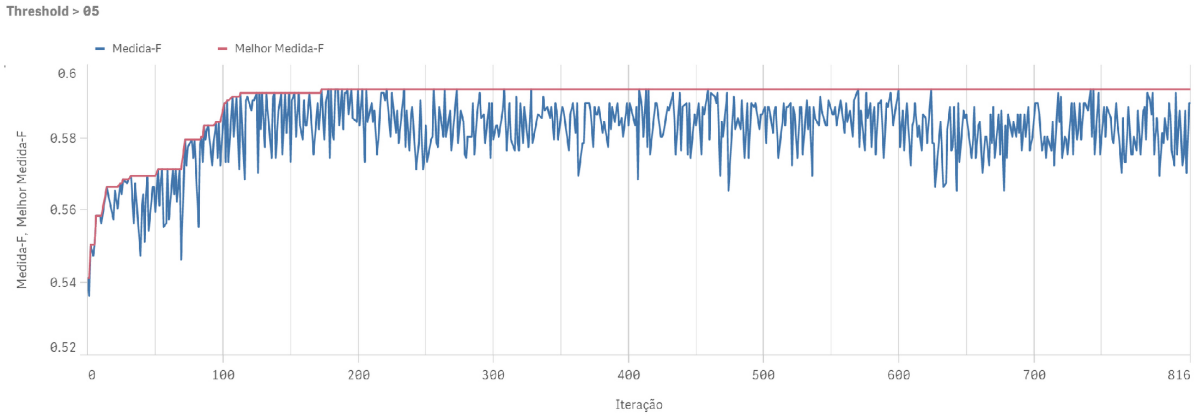


Figura 6.5: **Resultado para SSNEL com** $threshold > 0,5$

Considerando as Figuras 6.7 e 6.8, com respectivos *thresholds* maiores que $0,7$ e $0,8$, os

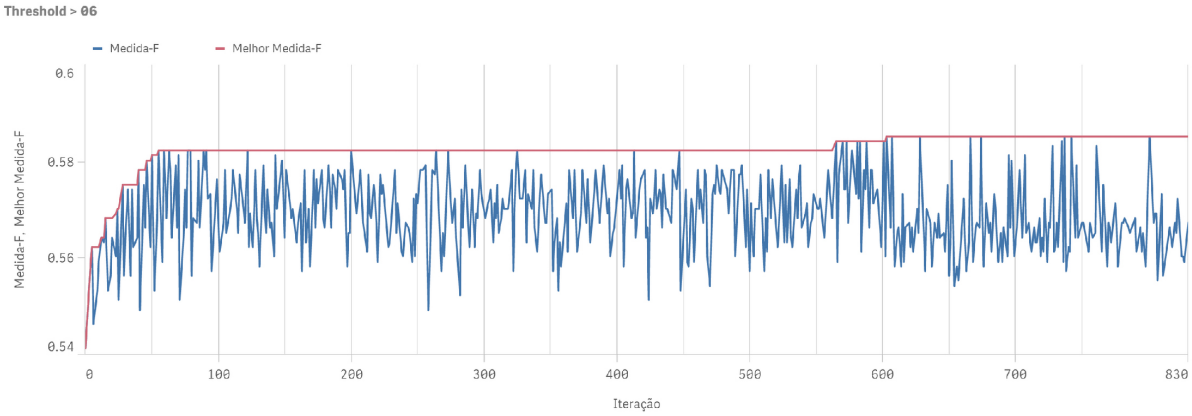


Figura 6.6: **Resultado para SSNEL com** $threshold > 0,6$

valores máximos obtidos foram, respectivamente, $FmC_{1343} = 0,612$ e $FmC_{1591} = 0,621$, embora mais iterações tenham sido realizadas. Para $threshold > 0,7$ foram realizadas 2553 iterações e para $threshold > 0,8$, 1728.

Considerando o valor de i para que $FmC_i > 0,593$ (melhor resultado obtido por $threshold > 0,5$), para $threshold > 0,7$, $i \geq 92$. Já para $threshold > 0,8$, $i \geq 174$. Esses valores indicam claramente uma convergência mais veloz para melhores valores de medida-F quando a confiabilidade das classificações são maiores.

Assim, para os outros dois experimentos cujas curvas de aprendizado são apresentadas nas Figuras 6.9 e 6.10, foi utilizado $threshold > 0,7$.

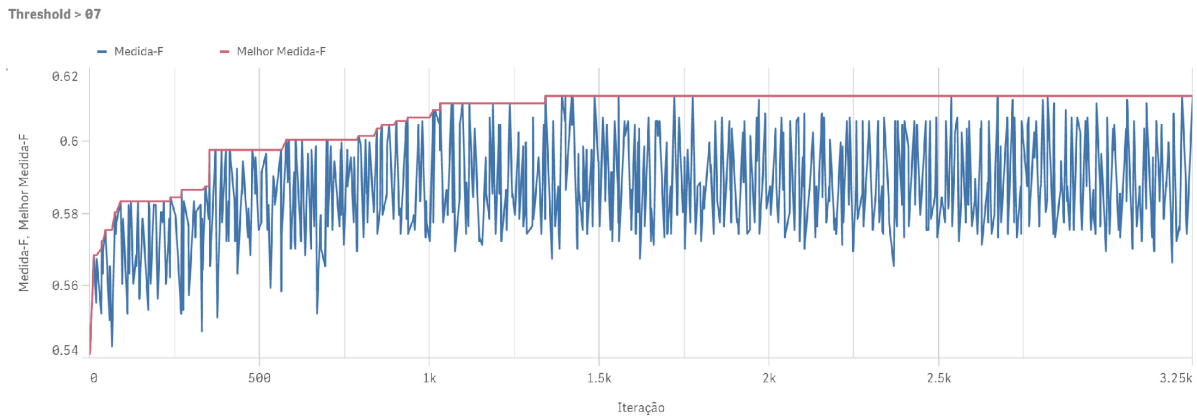


Figura 6.7: **Resultado para SSNEL com** $threshold > 0,7$

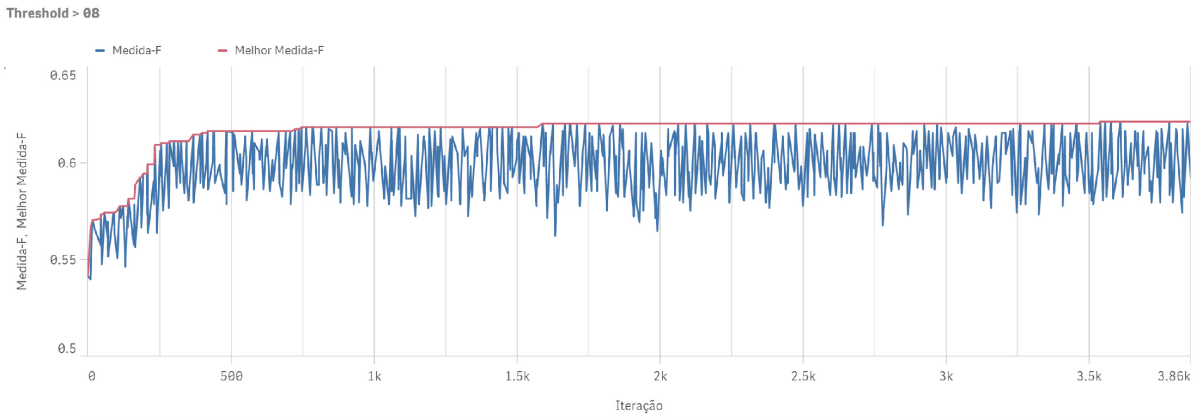


Figura 6.8: **Resultado para SSNEL com** $threshold > 0,8$

Na Figura 6.9, em 387 iterações, o valor máximo $FmC_{387} = 0,582$ foi alcançado quando a forma de selecionar novas instâncias de treinamento foi utilizar instâncias rotuladas com as cinco relações menos frequentes do conjunto de treinamento, indicando que, em comparação com os experimentos com $threshold > 0,7$ e $threshold > 0,8$, a medida-F teve melhora menos veloz, possivelmente porque as relações menos frequentes nos corpora tenham fornecido pouca informação para generalizar seu aprendizado. Quando todas as relações são consideradas, as mais frequentes foram melhor aprendidas e podem ter seu aprendizado mais generalizado.

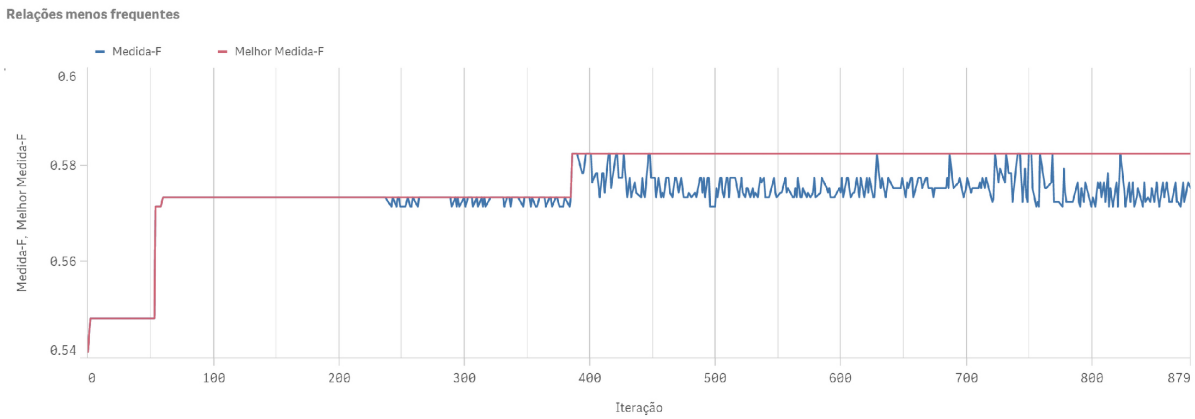


Figura 6.9: **Resultado para SSNEL com seleção de relações menos frequentes e** $threshold > 0,7$

O experimento que teve o menor ganho nos resultados foi quando o módulo LPS não foi considerado. A Figura 6.10 exibe a curva de aprendizado desse experimento. Mesmo após 1284 iterações executadas, a maior medida-F foi $FmC_{838} = 0,575$. Aqui temos que o módulo LPS desempenhou papel fundamental na melhoria dos classificadores C_i, transferindo seu conhecimento (representado pelos padrões lexicais) para os mesmos.

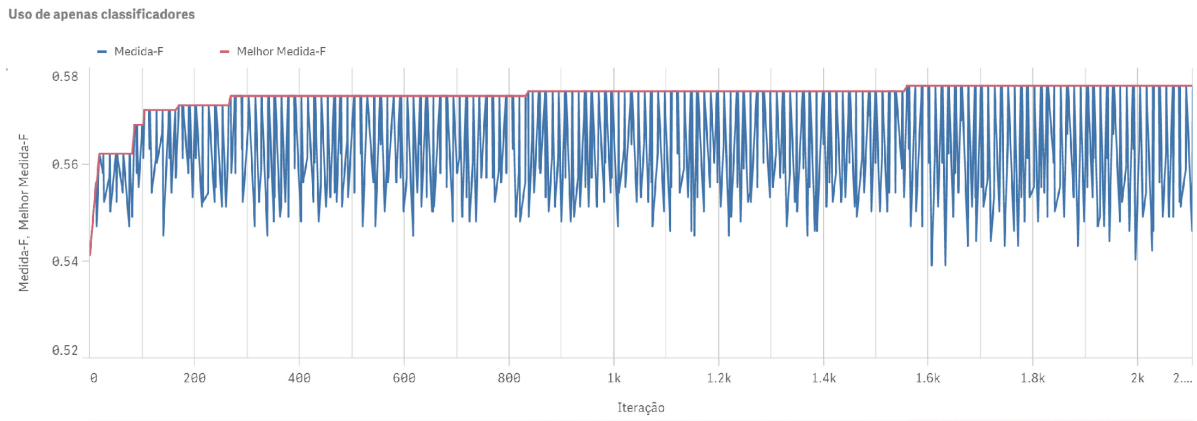


Figura 6.10: **Resultado para SSNEL utilizando apenas classificadores e** *threshold* $> 0,7$

O experimento com *threshold* $> 0,8$ obteve até o momento a maior medida-F, com $FmC_{1591} = 0,621$, com um aumento de 14,36% no valor inicial do aprendizado. Essa melhoria foi obtida com a incorporação de 1894 novas instâncias de treinamento (de 5400 para 7294), selecionadas em 152 iterações que levaram a melhor desempenho. No total, foram realizadas 3856 iterações, com um lote de 5 documentos cada.

A seguir, um rápido experimento para o Inglês é relatado, aplicando o *framework* SSNEL. Tal experimento foi realizado com o objetivo de verificar se o SSNEL melhoraria os resultados mesmo quando o aprendizado supervisionado conta com uma quantidade razoável de dados anotados manualmente.

6.2.2.3 Experimento para o Inglês

Para instanciar o *framework* SSNEL para o Inglês, foi utilizado o córpus RST-DT (RST *Discourse Treebank* - Carlson *et al* (2001)). Esse córpus é o mais amplamente utilizado

Tabela 6.19: **Valor máximo da medida-F obtido em cada experimento SSNEL. Também é apresentada a quantidade de iterações necessárias para se chegar ao valor e a quantidade de iterações executadas.**

Experimento SSNEL	**Item**	**Valor**
$threshold > 0,3$	Medida-F Max	0,581
	Iterações para Max	752
	Iterações executadas	831
$threshold > 0,5$	Medida-F Max	0,593
	Iterações para Max	745
	Iterações executadas	817
$threshold > 0,6$	Medida-F Max	0,585
	Iterações para Max	802
	Iterações executadas	831
$threshold > 0,7$	Medida-F Max	0,612
	Iterações para Max	1343
	Iterações executadas	**2553**
$threshold > 0,8$	Medida-F Max	**0,620**
	Iterações para Max	1591
	Iterações executadas	1728
Relação menos frequente e $threshold > 0,7$	Medida-F Max	0,582
	Iterações para Max	387
	Iterações executadas	599
Somente classificadores e $threshold > 0,7$	Medida-F Max	0,575
	Iterações para Max	838
	Iterações executadas	1284

nos trabalhos de *parsing* discursivo para o Inglês. Na Tabela 6.20, o RST-DT é comparado aos corpora para o Português (mencionados como RST-PT). Vê-se que, embora o RST-PT seja formado por quatro corpora, o RST-DT contém mais documentos (13,24%) e mais palavras (45,96%). O córpus RST-DT foi particionado em 7/10 para treinamento inicial e 3/10 para teste, similarmente ao RST-PT.

Tabela 6.20: **Tamanho dos corpora RST-PT para o Português, seus componentes, e do córpus para o Inglês (EN), o RST-DT.**

Córpus/corpora	**Língua**	**Documentos**	**Palavras**
RST-PT	PT	340	120,847
CSTNews		140	47,240
Rhetalho		50	2,903
Summ-it		50	16,704
CorpusTCC		100	53,000
RST-DT	EN	385	176,383

A diferença na instanciação desse experimento é que não foi utilizado um módulo LPS, apenas os classificadores C_i. Os classificadores foram criados considerando os atributos do *parser* HILDA (Hernault *et al*, 2010) e, para a geração da árvore sintática, foi utilizado o *parser* sintático de Stanford (Klein & Manning, 2003).

O comportamento da curva de aprendizado desse experimento é exibido na Figura 6.11 com 105 iterações no *framework*. O maior valor de medida-F foi $FmC_{65} = 0,652$, um aumento de 2,6% em relação a $FmC_1 = 0,635$. Nessa figura, a curva superior indica a melhor medida-F obtida até a iteração atual. A fim de confirmar a significância estatística de C_{65}, em comparação com C_1, o teste T, com significância $> 0,1$ foi aplicado, indicando significância estatística entre os dois classificadores. Essa melhoria foi obtida sem a necessidade de intervenção humana no aprendizado, indicando que mesmo com uma quantidade razoável de dados rotulados iniciais, é possível melhorar o desempenho na tarefa com o uso da semissupervisão sem fim.

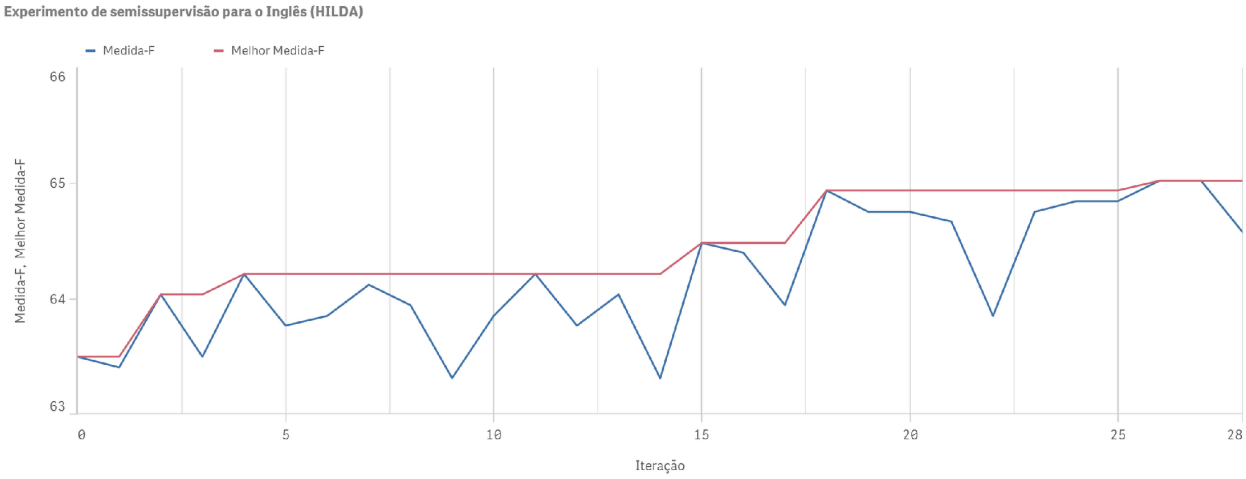


Figura 6.11: **Resultado para SSNEL utilizando apenas classificadores e** *threshold* $> 0,7$ **para o Inglês**

6.2.2.4 Conclusões dos resultados

Conclui-se que o *framework* SSNEL possibilita melhoria nos resultados da identificação das relações discursivas, sem a necessidade de geração manual de mais dados rotulados. O uso de grande quantidade de dados numa abordagem semissupervisionada sem fim, com controle de mudança de conceito, obteve sucesso em melhorar a classificação.

Em comparação com o aprendizado supervisionado (classificador C_0), os resultados melhoraram significativamente. Inclusive, o segmentador discursivo (Capítulo 5) contribuiu com essa melhoria, pois todos os textos obtidos da web foram segmentados automaticamente, antes da detecção das relações discursivas. Um segmentador discursivo que não realiza bem a segmentação prejudica o processo de *parsing* discursivo, pois pode gerar segmentos a mais ou a menos, causando, respectivamente, a identificação de relações não necessárias/incorretas e perder relações entre segmentos não detectados.

Verificou-se, também, que uma abordagem além do *self-training*, utilizando um outro modelo de classificação (módulo LPS), melhora ainda mais os resultados. Nessa pesquisa, o módulo com padrões lexicais do *parser* DiZer "transferiu"conhecimento para os classificadores, melhorando o desempenho desses últimos, na tarefa discursiva.

No entanto, quando não se tem disponível um outro modelo de aprendizado, o mero *self-training* com controle de mudança de conceito pode melhorar significativamente o desempenho de classificadores na tarefa discursiva. Essa afirmação foi corroborada pelo experimento com o *framework* SSNEL para o Inglês.

A seguir, são apresentadas metodologias para a identificação da nuclearidade dos segmentos, após a relação discursiva ter sido definida.

6.3 Nuclearidade

A nuclearidade é uma das características que diferenciam a RST (Mann & Thompson, 1987) de outras teorias discursivas, pois permite a atribuição de importância aos segmentos relacionados por relações discursivas. Na ocorrência de uma relação discursiva entre dois segmentos textuais, geralmente adjacentes, cada um desempenha um papel em relação ao outro. Por exemplo, numa relação *Justify* (Tabela **??**, página **??**), o segmento mais nuclear é a ideia que o autor deseja justificar. Para isso, o autor apresenta um segmento satélite (menos importante nesse contexto) que, quando aceito, aumentará a prontidão do leitor a aceitar o segmento nuclear.

Há um grupo de relações retóricas cujos segmentos têm a mesma importância para o

intento do autor do texto. Tais relações são ditas multinucleares, por apresentarem mais de um núcleo, diferentemente das mononucleares (como a exemplificada anteriormente, *Justify*), que apresentam apenas um núcleo. As relações multinucleares consideradas nesta pesquisa são *Contrast* (quando duas ideias estão sendo contrastadas, mas são ambas importantes), *Joint* (quando não se encontrou uma relação retórica entre os segmentos), *List* (ao listar ideias relacionadas sem grau de importância entre elas) e *Sequence* (ao apresentar ideias igualmente importantes que se sucedem no tempo).

Em um *parser* discursivo, a identificação da nuclearidade pode ser feita em conjunto com a identificação da relação retórica. No caso desta pesquisa, essas etapas foram separadas para diminuir a esparsidade dos dados de treinamento. Por exemplo, se a relação *Antithesis* fosse representada junto com sua nuclearidade, duas outras classes seriam geradas: *Antithesis-NS* e *Antithesis-SN*, cada uma com menos instâncias que apenas uma classe *Antithesis*.

A seguir, são apresentadas as metodologias utilizadas para a identificação das nuclearidades entre dois segmentos. A primeira abordagem é o uso de ordem canônica, ou mais frequente, das nuclearidades nos corpora RST. Posteriormente, experimentos com aprendizado automático são apresentados de forma similar ao aprendizado supervisionado, explorado anteriormente, neste capítulo.

6.3.1 Ordem Canônica

A identificação da nuclearidade de relações multinucleares (núcleo-núcleo - NN) é feita tão logo uma dessas relações seja encontrada, o que não ocorre na identificação da nuclearidade para relações mononucleares, pois, para uma relação mononuclear, a ordem núcleo-satélite (NS) e satélite-núcleo(SN) não é direta. No entanto, verifica-se que, para boa parte das relações, há uma ocorrência predominante, conforme pode ser vista na Tabela 6.21. Nessa tabela, também são apresentadas as ocorrências das relações multinucleares e, para cada uma das relações listadas, é calculada a Medida-F, caso a ordem da nuclearidade mais frequente (chamada ordem canônica, em negrito) seja utilizada como rótulo.

A medida-F para o nível intrassentencial foi obtida considerando todos os pares de

segmentos do respectivo nível dos corpora RST. Para a medida-F do nível intersentencial (Tabela 6.22), os pares de segmentos do respectivo nível foram considerados. Nessas avaliações, foram considerados os segmentos presentes nos corpora, portanto, o segmentador discursivo não teve influência nesses resultados.

No nível intrassentencial (Tabela 6.21), a medida-F média obtida foi de 0,844, considerando as relações multinucleares, que têm sempre resultado 1. Quando essas relações são desconsideradas, a medida-F média cai para 0,803.

Tabela 6.21: **Ordem canônica da nuclearidade nas relações retóricas. O valor em negrito representa a maior ocorrência nos corpora RST para o nível intrassentencial.**

Relação	**NS**	**SN**	**NN**	**Medida-F**
Attribution	340	**560**	0	0,622
Antithesis Concession Contrast	79	**219**	199	0,441
Background Circumstance	245	**410**	0	0,626
Volitional-Result Non-Volitional-Result Volitional-Cause Non-Volitional-Cause	**478**	224	0	0,681
Comparison	**53**	8	0	0,869
Condition Otherwise	54	**59**	0	0,522
Elaboration	**3564**	32	0	0,991
Enablement Motivation Purpose	**643**	154	0	0,807
Evidence Justify Explanation	**533**	60	0	0,899
Interpretation Evaluation Conclusion	**227**	52	0	0,814
List	0	0	**1572**	1,000
Joint	0	0	**13**	1,000
Means	**90**	17	0	0,841
Restatement	**60**	1	0	0,983
Same-unit	0	0	**953**	1,000
Sequence	0	0	**494**	1,000
Solutionhood	3	**67**	0	0,957
Parenthetical	**951**	10	0	0,989
Summary	**5**	0	0	1,000
Média geral	**7325**	1873	3231	0,844
Média sem multinucleares	**7325**	1873	–	0,803

No caso da relação *Summary*, geralmente se apresenta um sumário (segmento satélite) após a descrição dos itens sumarizados (segmento nuclear). Essa ordem canônica pode ser observada tanto no nível intra quanto no intersencial e leva a medida-F de 1. A relação *Elaboration* tem análise semelhante em ambos os níveis, pois geralmente se elabora algo (segmento satélite) depois que esse algo foi apresentado (segmento nuclear). Tal comportamento no nível intrassentencial pode ser observado ainda nas relações *Restatement* e *Solutionhood*. Para o nível intersentencial, observa-se nas relações *Means*, *Restatement* e *Solutionhood*.

Na Tabela 6.22, são apresentados as medidas-F obtidas para o nível inter-sentencial. A média geral ficou em 0,865. Sem considerar as relações multinucleares, a medida diminui para 0,837.

Uma limitação encontrada na atribuição da nuclearidade está relacionada ao grupo de relações formada por *Antithesis*, *Concession* e *Contrast*, em que há relações mononucleares (as duas primeiras) e multinucleares (a última). Essa limitação ocorre devido ao único rótulo atribuído pelo módulo de identificação das relações retóricas. Tal agrupamento é motivado pelas restrições similares das relações citadas. Caso seja necessário a escolha de uma das relações do grupo, a ordem canônica pode ser aplicada às duas primeiras relações e a nuclearidade NN (Núcleo-Núcleo) para a última relação.

Como alternativa ao uso da ordem canônica para identificar as nuclearidades, foi explorado aprendizado automático de forma semelhante aos experimentos com conjuntos de atributos para identificação das relações retóricas no Capítulo 6. Esses experimentos são relatados a seguir.

6.3.2 Classificação com aprendizado supervisionado

Tendo em vista que os mesmos atributos extraídos para a classificação das relações retóricas podem ser utilizados na identificação da nuclearidade em um par de segmentos, procedeu-se à experimentação com aprendizado automático nessa tarefa. Isso se justifica dado que a nuclearidade é definida de acordo com a relação escolhida. Grupos de atributos foram testados para escolha do melhor conjunto que identifique a nuclearidade.

Tabela 6.22: **Ordem canônica da nuclearidade nas relações retóricas. O valor em negrito representa a maior ocorrência nos corpora RST para o nível intersentencial.**

Relação	**NS**	**SN**	**NN**	**Medida-F**
Attribution	**5**	3	0	0,625
Antithesis Concession Contrast	22	93	**112**	0,974
Background Circumstance	53	**211**	0	0,799
Volitional-Result Non-Volitional-Result Volitional-Cause Non-Volitional-Cause	**175**	65	0	0,745
Comparison	**19**	5	0	0,792
Condition Otherwise	**5**	6	0	0,833
Elaboration	**1950**	17	0	0,991
Enablement Motivation Purpose	**49**	28	0	0,636
Evidence Justify Explanation	**351**	37	0	0,905
Interpretation Evaluation Conclusion	**195**	38	0	0,837
List	0	0	**802**	1,000
Joint	0	0	**13**	1,000
Means	**30**	0	0	1,000
Restatement	**31**	1	0	0,969
Sequence	0	0	**289**	1,000
Solutionhood	3	**67**	0	0,957
Parenthetical	1	1	0	0,500
Summary	**3**	0	0	1,000
Média geral	**2892**	572	1216	0,865
Média sem multinucleares	**2892**	572	–	0,837

No seção anterior, foram apresentadas duas medidas-F médias, nas Tabelas 6.21 e 6.22, a primeira considerando as relações multinucleares, e a segunda, não. Os valores obtidos nas tabelas a seguir (6.23 e 6.24) não consideraram as relações multinucleares e devem, portanto, ser comparados com as medidas-F médias das ordens canônicas apresentadas.

Nas Tabelas 6.23 e 6.24, os conjuntos de atributos são: HILDA-PT para os atributos da adaptação do *parser* HILDA (Hernault *et al*, 2010); OT para organização textual (Tabela 6.4, página 106); M para morfossintaxe (Tabela 6.5, página 106); S para semântica e D

para discurso (Tabela 6.7, página 108).

Tabela 6.23: **Resultados dos experimentos com aprendizado supervisionado para a identificação de nuclearidade no nível intrassentencial, considerando diversos grupos de atributos, em que OT é Organização Textual, M é Morfossintaxe, S é Semântica e D é Discursos.**

Atributos	**Medida-F**	**Acurácia**
HILDA-PT + OT + M + S + D	0,830	0,832
HILDA-PT + OT	0,830	0,836
HILDA-PT	0,851	0,859
HILDA-PT + S	0,857	0,862
HILDA-PT + M + S + D	0,862	0,870
HILDA-PT + M + S	**0,868**	**0,873**
HILDA-PT + M	0,865	0,872
Ordem canônica sem multinucleares	**0,803**	**0,803**

Na Tabela 6.23, verifica-se que o grupo de atributos da adaptação HILDA-PT juntamente com os atributos morfossintáticos e semânticos obteve medida-F de 0,868, superior à aplicação da ordem canônica (Tabela 6.21), para o nível intrassentencial (medida-F de 0,803). Isso indica que os atributos foram bons indicadores da ordenação da nuclearidade, especialmente para relações em que a ordem canônica não apresenta bons resultados, como nas relações *Attribution*, *Antithesis-Concession-Contrast*, *Background-Circumstance* e *Condition-Otherwise*.

Já para o nível intersentencial, os resultados com aprendizado de máquina (Tabela 6.24) não foram superiores à aplicação da ordem canônica no mesmo nível (Tabela 6.22) que obteve medida-F média de 0,837, contra medida-F máxima de 0,758 no aprendizado automático com atributos da adaptação HILDA-PT mais atributos discursivos (HILDA-PT + D). Isso se deve a que a maior parte das relações obtiveram bons resultados utilizando a ordem canônica, exceto por *Attribution* e *Enablement-Motivation-Purpose*.

A identificação da nuclearidade mostrou-se uma tarefa com melhores resultados que a identificação das relações retóricas. No nível intersentencial, tanto com o uso da ordem canônica quanto com o uso dos classificadores, obteve-se melhores medidas-F que no nível intra. Tal fenômeno ocorreu de forma inversa na identificação das relações retóricas, em que a identificação das relações no nível intersentencial obteve resultados muito inferiores ao nível intra. Isso se deve ao fato já explicado que, quando uma relação é identificada, é

Tabela 6.24: **Resultados dos experimentos com aprendizado supervisionado para a identificação de nuclearidade no nível inter-sentencial, considerando diversos grupos de atributos, em que M é Morfossintaxe, S é Semântica e D é Discurso.**

Atributos	**Medida-F**	**Acurácia**
HILDA-PT + S + D	0,737	0,801
HILDA-PT	0,737	0,819
HILDA-PT + M	0,737	0,819
HILDA-PT + S	0,742	0,819
HILDA-PT + M + S	0,746	0,819
HILDA-PT + M + D	0,748	0,816
HILDA-PT + M + S + D	0,757	0,819
HILDA-PT + D	**0,758**	**0,816**
Ordem canônica sem multinucleares	**0,837**	**0,837**

mais simples definir a nuclearidade dos segmentos.

Dado os resultados, na composição do *parser* discursivo, na etapa da identificação das nuclearidades, quando se trata de relações do nível intrassentencial, aplica-se o classificador com a combinação de atributos da adaptação HILDA-PT mais os morfossintáticos e semânticos (Tabela 6.23 - HILDA-PT + M + S). Quando se tratar de relações do nível intersentencial, a ordem canônica será aplicada.

7 Considerações Finais

Esse trabalho preencheu uma lacuna de pesquisa na área do discurso para o Português do Brasil, explorando técnicas de aprendizado automático na tarefa discursiva. Para isso, valeu-se do pioneirismo do *parser* DiZer (Pardo, 2005) em realizar a análise com uso de padrões discursivos, que foram muito úteis no aprendizado.

A tese foi confirmada, pois a utilização de grande quantidade de dados não rotulados no aprendizado sem fim melhorou significativamente os resultados na identificação das relações retóricas para o Português, atingindo resultados do estado-da-arte. As hipóteses secundárias foram confirmadas, especialmente a hipótese relacionada ao hibridismo, em que as regras geradas manualmente contribuiram para o aprendizado automático, reduzindo o número de classes para o classificador que identifica as relações entre pares de segmentos.

Esperava-se que conhecimentos linguísticos profundos fossem mais úteis no nível discursivo, no entanto, como visto no capítulo anterior, o uso de atributos do nível morfossintático levou a melhores resultados que o uso de atributos do nível semântico e discursivo. Outros atributos do nível semântico e discursivo devem ser explorados a fim de corroborar essa conclusão.

7.1 Contribuições

A seguir as contribuições desta pesquisa foram agrupadas em teóricas e práticas. São resumidas as limitações encontradas nesse trabalho e, ao final, são apresentados trabalhos

futuros.

7.1.1 Teóricas

A principal contribuição teórica relaciona-se à tese desta pesquisa e consiste na formalização de uma arquitetura de aprendizado semissupervisionado sem fim que utiliza dois modelos de identificação de relações discursivas, em que apenas um é treinado (classificadores C_i), dado que o outro (LPS) foi desenvolvido manualmente.

Proeminentes modelos de análise discursiva do Inglês (Soricut & Marcu (2003) e Hernault *et al* (2010)) foram adaptados para o Português. Esses modelos foram explorados e seus conjuntos de atributos foram expandidos com novos atributos agrupados de acordo com seus níveis linguísticos. Nesse processo, foi possível verificar o impacto dos atributos de cada nível linguístico na análise discursiva.

Nesta pesquisa, também, explorou-se todos os corpora RST de forma conjunta. Em trabalhos anteriores, eles eram tratados de forma independente.

Confirmou-se que o hibridismo ajuda nas mais diversas tarefas de aprendizado automático. Isso é visto pelo tratamento de algumas relações (*Attribution* e *Parenthetical*) com o uso de regras criadas manualmente, diminuindo a confusão entre as classes. A exclusão das relações multinucleares dos classificadores de nuclearidade também ajudaram na tarefa, visto que a ordem da nuclearidade dessas relações é direta, dada apenas a relação.

7.1.2 Práticas

A principal contribuição prática deste trabalho é o avanço do estado da arte do *parsing* discursivo para o Português, em específico, na etapa mais complexa de identificar as relações retóricas entre UBDs. Por se tratar de um aprendizado sem fim, espera-se melhorar ainda mais os resultados com o decorrer das iterações no *framework* SSNEL.

Os dados anotados automaticamente com alta confiabilidade (maior que 0,8, por exemplo) e que contribuíram para a geração de classificadores com melhor desempenho compõem um acréscimo ao corpora RST anotado manualmente.

Para a extração dos atributos utilizados no aprendizado automático, foram geradas algumas ferramentas: um identificador de cadeias lexicais entre substantivos de um texto; um lematizador; e um web crawler que busca textos jornalísticos, faz a extração do texto principal da página e certifica-se que esteja escrito em Português. Também foram compiladas listas de marcadores discursivos, verbos de atribuição e listas de tipos de conjunções de trabalhos anteriores neste trabalho.

7.1.3 Limitações e Dificuldades

Dada a ambiguidade de algumas relações e seguindo trabalhos da área, as relações RST similares foram agrupadas em 19 grupos. Esse agrupamento levou a algumas limitações, como a identificação da nuclearidade com o uso da ordem canônica. Considere o grupo *Antithesis-Concession-Contrast*: o uso da nuclearidade canônica atribuirá a ordem *NN, Núcleo-Núcleo* aos exemplares desse grupo. No entanto, sabe-se que as relações *Antithesis* e *Concession* são mononucleares, e, portanto, conterão incorreta nuclearidade. Tal limitação ocorre de forma inversa no uso dos classificadores para identificar as nuclearidades, dado que as instâncias multinucleares não são consideradas, fazendo com que, para o referido grupo, uma ordem mononuclear seja atribuída (*SN* ou *NS*). As outras relação multinucleares não são afetadas, pois não estão agrupadas com relações mononucleares.

Com relação à segmentação discursiva, como diversas regras dependem da pontuação do texto, o desempenho tende a cair em texto mal escritos, em que não se tem as pontuações corretamente colocadas.

7.2 Trabalhos futuros

O aprendizado semissupervisionado sem fim foi abordado de maneira relativamente simples, mas indicando caminhos promissores para futuros trabalhos.

Tendo em vista que algoritmos tradicionais de aprendizado automático, como árvores de decisão (Quinlan, 1993) e *support vector machines* (Vapnik, 1995) foram utilizados, a exploração de outras técnicas de aprendizado supervisionado, como os algoritmos de

Deep Learning e *Conditional Randon Fields* (CRFs), pode levar a bons resultados iniciais e permitir melhor generalização do aprendizado SSNEL.

No caso da análise discursiva para o Português, com a pouca quantidade de dados, o conjunto de testes fixo pode não ser tão representativo da tarefa, mesmo que o número de instâncias de cada classe seja selecionada proporcionalmente ao total encontrado nos corpora. Portanto, uma abordagem de expansão do conjunto de testes que seja feita durante o aprendizado sem fim pode aumentar a representatividade do conjunto, refletindo melhor a acurácia do aprendizado sem fim. A avaliação extrínseca pode ser útil nesse cenário, pois o desempenho de tarefas que utilizem o *parser* discursivo pode ser um bom indicador do desempenho do aprendizado.

A multi-visão no aprendizado automático (*co-training*, *tri-training* (Zhou & Li, 2005) e *emsembles* (Zhou, 2012)), com a separação do conjunto de atributos em dois ou mais conjuntos independentes e realizar o re-treino de todos os classificadores envolvidos na semissupervisão é uma abordagem que pode ser explorada. Como visões, os atributos agrupados por nível linguístico podem ser utilizados para a criação de classificadores distintos.

Dada a utilidade do módulo LPS no aprendizado semissupervisionado, o aprendizado automático de regras lexicais a serem incorporadas a LPS, como uma forma de re-treino, possibilitaria seu uso na multi-visão.

A exploração e desenvolvimento de técnicas de aprendizado incremental que possibilitem um aprendizado de máquina mais eficiente, não considerando todo o conjunto de dados simultaneamente, mas mantendo a mesma acurácia de um modelo não incremental é uma oportunidade de exploração.

Um outro ponto a ser explorado é o paralelismo durante o treinamento. Os algoritmos de aprendizado podem ser estudados para a verificação de partes que podem ser executadas em paralelo, reduzindo a complexidade de tempo do aprendizado semissupervisionado.

A medida-F (FmC_i) é utilizada a cada iteração i para verificar se houve degradação do aprendizado. A medida é utilizada como critério para identificação de uma possível mudança de conceito, descartando um classificador C_i se FmC_i for menor que FmC_{i-1}.

Como é utilizado um lote de novas instâncias ($dadosRotuladosCconfident$) para o retreino, esse lote pode conter instâncias que beneficiam e outras que prejudicam o aprendizado. Descartar todo esse lote pode não ser a melhor alternativa. Isso motiva a busca por técnicas mais elaboradas com vistas a tratar melhor o fenômeno de mudança de conceito.

Por fim, diversas tarefas do PLN contam com grande volume de dados disponível na web, permitindo a aplicação deste trabalho. A análise de sentimentos conta com diversos fóruns de discussão e portais de vendas com opiniões de consumidores sobre produtos. A sumarização automática, a exemplo da análise discursiva, conta com inúmeros textos que podem ser sumarizados no aprendizado semissupervisionado, assim como diversos corpora de sumários produzidos manualmente. Além de beneficiar outras áreas, a aplicação do *framework* SSNEL em outras tarefas possibilitaria sua avaliação extrínseca.

Bibliografia

Abney, S. (2007). *Semisupervised Learning for Computational Linguistics*. Número 1st in Chapman - Hall/CRC Computer Science - Data Analysis. Chapman E Hall/CRC.

Aires, R.; Aluísio, S.; Kuhn, D.; Andreeta, M.; Jr., O. O. (2000). Combining multiple classifiers to improve part of speech tagging: A case study for brazilian portuguese. In *Proceedings of the Brazilian AI Symposium*, pp. 20–22.

Asher, N.; Lascarides, A. (2003). Intentions and information in discourse. In *Proceedings of 32nd Meeting of the Association for Computational Linguistics*, pp. 34–41.

Balage F., P. (2012). *Use of Discourse Knowledge to Improve Lexicon-based Sentiment Analysis*. Master's thesis, Universidade do Algarve and University of Wolverhampton.

Banko, M.; Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pp. 26–33.

Bick, E. (2000). *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Ph.D. thesis, Aarhus University., Denmark University Press.

Blum, A.; Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100. ACM, New York, NY, USA.

Bosma, W. E. (2005). Extending answers using discourse structure. In *Proceeding of Recent Advances in Natural Language Processing Workshop on Crossing Barriers in Text Summarization Research*, pp. 2–9.

Branco, A.; Costa, F. (2008). A computational grammar for deep linguistic processing of portuguese: Lxgram, version a.4.1. Technical Report DI-FCUL-TR-08-17, University of Lisbon.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Brennan, S. E.; Friedman, M. W.; Pollard, C. J. (1987). A centering approach to pronouns. In *Proceedings of the 25th ACL*, pp. 155–162.

Cardoso, P.; Maziero, E.; Jorge, M.; Seno, E.; Felippo, A. D.; Rino, L.; Nunes, M.; Pardo, T. (2011a). CSTnews - a discourse-annotated corpus for single and multi-document summarization of news texts in Brazilian Portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pp. 85–105. Cuiaba/Brazil.

Cardoso, P. C. F.; Jorge, M.; Pardo, T. A. S. (2015). Exploring the rhetorical structure theory for multi-document summarization. In *Proceedings of the 5th Workshop RST and Discourse Studies*, pp. 1–10. Alicante/Spain.

Cardoso, P. C. F.; Pardo, T.; Nunes, M. (2011b). Métodos para sumarização automática multidocumento usando modelos semântico-discursivos. In *Proceedings of the 3rd RST Brazilian Meeting*, pp. 59–74.

Cardoso, P. C. F.; Pardo, T. A. S. (2016). Multi-document summarization using semantic discourse models. *Processamiento de Lenguaje Natural*, 56:57–64.

Cardoso, P. C. F.; Pardo, T. A. S.; Taboada, M. (2013). On the contribution of discourse to topic segmentation. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue – SIGdial*, pp. 92–96. Metz, France.

Carlson, A.; Betteridge, J.; Junior, E. H.; Mitchell, T. (2009). Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL HLT Workshop on*

Semi-supervised Learning for Natural Language Processing, pp. 1–9. Association for Computational Linguistics, Colorado/USA.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; H., E. R.; Mitchell, T. (2010). Toward an architecture for never-ending language learning. In *Proceedings of Association for the Advancement of Artificial Intelligence*, volume 5, pp. 1306–1313.

Carlson, L.; Marcu, D. (2001). Discourse tagging reference manual. Technical Report, ISI-TR.

Carlson, L.; Marcu, D.; Okurowski, M. (2001). Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of Second SIGdial Workshop on Discourse and Dialogue*, volume 16, pp. 1–10.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 132–139. Association for Computational Linguistics, Stroudsburg, PA, USA.

Chiarcos, C. (2012). Towards the unsupervised acquisition of discourse relations. In *Proceedings of 50th Annual Meeting of the Association for Computational Linguistics*, pp. 213–217.

Clarke, J.; Lapata, M. (2010). Discourse constraints for document compression. *Computer Linguistics*, 36(3):411–441.

Collovini, S.; Carbonel, T.; Coelho, J.; Fuchs, J.; Vieira, R. (2007). Summ-it: um corpus anotado com informações discursivas visando à sumarização automática. *Congresso Nacional da SBC*, pp. 1605–1614.

Corston-Oliver, S. (1998). *Computing Representations of the Structure of Written Discourse*. Ph.D. thesis, University of California, Santa Barbara, CA, USA.

Cristea, D.; Ide, N.; Romary, L. (1998). Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the Coling/ACL*, pp. 281–285. Montreal, QC.

Cunha, I.; SanJuan, E.; Torres-Moreno, J.; Cabré, M. T.; Sierra, G. (2012). *Computational Linguistics and Intelligent Text Processing*, volume 7181 of *Lectures Notes in Computer Science*, chapter A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra-sentence Discourse Segments in Spanish, pp. 462–474. Springer-Verlag, Berlin, Heidelberg.

Cunha, I.; Torres-Moreno, J.; Sierra, G. (2011). On the development of the rst spanish treebank. In *Proceedings of the fifth Law Workshop*, pp. 1–10.

Curran, J. R.; Murphy, T.; Scholz, B. (2007). Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pp. 172–180.

Dias, M. S.; Pardo, T. A. S. (2015). Enriching entity grids and graphs with discourse relations: the impact in measuring local coherence in multi-document summaries. In *Proceedings of the 10th Brazilian Symposium in Information and Human Language Technology - STIL*, pp. 151–160. Natal/Brasil.

duVerle, D.; Prendinger, H. (2009). A novel discourse parser based on support vector machine classification. In *Proceedings of Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pp. 665–673.

Evert, S. (2008). A lightweight and efficient tool for cleaning web pages. In *In Proceedings of the 6th International Conference on Language Resources and Evaluation (2008) Key: citeulike:8194850*. European Language Resources Association.

Faceli, K.; Lorena, A.; Gama, J.; Carvalho, A. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina.* LTC.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, Massachusetts.

Feng, V.; Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of 50th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 60–68.

Freund, Y.; Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pp. 148–156. San Francisco, USA.

Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. (2004). Learning with drift detection. In *Proceedings of 17th Brazilian symposium on Artificial Intelligence SBIA*, pp. 286–295.

Garrette, D.; Klein, E. (2009). An extensible toolkit for computational semantics. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pp. 116–127. Association for Computational Linguistics, Stroudsburg, PA, USA.

Grosz, B.; Sidner, C. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3).

Gutiérrez, V. (2010). *Classificação semi-supervisionada baseada em desacordo por similaridade*. Master's thesis, University de São Paulo, São Carlos, SP.

Hernault, H.; Prendinger, H.; duVerle, D.; Ishizuka, M. (2010). HILDA: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282. Montreal, QC.

Hulten, G.; Spencer, L.; Domingos, P. (2011). Mining time-changing data streams. In *Proceedings of Association for Computational Linguistics - Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 97–106. ACM Press.

Hutchins, J. (2010). Machine translation: a concise history. *Journal of Translation Studies*, 13:29–70.

Inam, S.; Shoaib, M.; Majeed, F.; Sharjeel, M. I. (2012). Ontology based query reformulation using rhetorical relations. *International Journal of Computer Science Issues*, 9(4):261.

Iruskieta, M.; Cunha, I.; Taboada, M. (2015). A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora. *Language Resources and Evaluation*, 49(2):263–309.

Jordan, M. (1992). An integrated three-pronged analysis of a fund-raising letter. *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pp. 171–226.

Jorge, M.; Dias, M. S.; Pardo, T. A. S. (2014). Building a language model for local coherence in multi-document summaries using a discourse-enriched entity-based model. In *Proceedings of the Brazilian Conference on Intelligent Systems - BRACIS*, pp. 44–49. São Carlos, SP.

Joty, S.; Carenini, G.; Ng, R. T. (2015). Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Kamp, H.; Reyle, U. (1993). From discourse to logic: Introduction to model theoretic semantics of natural language, formal logic and discourse representation theory. *Springer Science and Business Media*, (42).

Kehler, A. (2002). *Coherence, Reference and the Theory of Grammar*. CSLI Publications.

Kilgarriff, A.; Rychly, P.; Smrz, P.; Tugwell, D. (2004). The sketch engine. In *Proceedings of Euralex*, pp. 105–116.

Kipper, K.; Dang, H. T.; Palmer, M. (2000). Class-based construction of a verb lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pp. 691–696. AAAI Press.

Klein, D.; Manning, C. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423–430.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300.

Lascarides, A.; Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5):437–493.

Lin, H., Z. Ng; Kan, M. (2011). Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language*, pp. 997–1006.

Lin, J.; Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce (Synthesis Lectures on Human Language Technologies)*. Morgan and Claypool Publishers, 171 pp..

Lin, Z.; Kan, M.; Ng, H. (2009). Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of 2009 Conference on Empirical Methods in Natural Language Processing*, volume 1, pp. 343–351.

Lioma, C.; Larsen, B.; Lu, W. (2012). Rhetorical relations for information retrieval. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval*, pp. 931–940.

Liu, B. (2011). *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data (Data-Centric Systems and Applications)*. Springer.

Magerman, D. (1995). Statistical decision-tree models for parsing. In *Proceedings of Association for Computational Llinguistics 1995*, pp. 276–283. Cambridge, Massachusetts.

Mahmud, R.; Ramsay, A. (2005). *Finding Discourse Relations in Student Essays*, chapter Finding Discourse Relations in Student Essays, pp. 116–119. Springer-Verlag, Berlin, Heidelberg.

Makatchev, M.; Lee, M.; Simmons, R. (2009). Relating initial turns of human-robot dialogues to discourse. In *Proceedings of the 4th ACM/IEEE international conference on human robot interaction*, pp. 321–322.

Mann, W.; Thompson, S. (1987). Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts.* Ph.D. thesis, University of Toronto.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization.* The MIT press, Cambridge, Massachusetts.

Marcu, D.; Carlson, L.; Watanabe, M. (2000). The automatic translation of discourse structures. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 9–17.

Marcu, D.; Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 368–375.

Maziero, E.; Pardo, T.; da Cunha, I.; Torres-Moreno, J.; SanJuan, E. (2011). Dizer 2.0 - an adaptable on-line discourse parser. In *Proceedings of 3rd RST Brazilian Meeting*, pp. 1–17.

Maziero, E.; Pardo, T.; Di Felippo, A.; Dias-da Silva, B. (2008). A base de dados lexical e a interface web do tep 2.0 - thesaurus eletrônico para o português do brasil. In *Anais do VI Workshop em Tecnologia da Informação e da Lingaugem Humana*, pp. 390–392.

Maziero, E.; Pardo, T.; Nunes, M. (2007). Identificação automática de segmentos discursivos: o uso do parser Palavras. Technical Report 305, University of Sao Paulo.

Maziero, E. G.; Pardo, T. A. S. (2009). Automatização de um método de avaliação de estruturas retóricas. In *Proceedings of the RST Brazilian Meeting*, pp. 1–9.

Mitchel, T. (1997). *Machine Learning.* McGraw-Hill.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.;

Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; Welling, J. (2015). Never-Ending Learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificil Intelligence*.

Mitkov, R. (1993). How could rhetorical relations be used in machine translation (and at least two open questions)? In *Proceedings of Association for Computational Llinguistics Workshop on Intentionality and Structure Discourse Relations*.

Morris, J.; Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Muniz, M. C. M. (2004). *A construção de recursos lingüístico-computacionais para o português do Brasil: o projeto de Unitex-PB*. Master's thesis, Universidade de São Paulo.

Nakasone, A.; Prendinger, H.; Ishizuka, M. (2006). Web presentation system using rst events. In *Proceedings of the fifth International joint conference on Autonomous agents and multiagents systems*, pp. 955–957.

Nóbrega, F.; Pardo, T. (2014). General purpose word sense disambiguation methods for nouns in portuguese. In *Proceedings of the PROPOR 2014 PhD and MSc/MA Dissertation Contest / 11st International Conference on Computational Processing of Portuguese*, pp. 94–101.

O'Donnell, M. (2000). Rsttool 2.4 - a markup tool for rhetorical structure theory. In *Proceedings of the International Natural Language Generation Conference*, pp. 253–256.

Oliveira, L. (2015). *Classificação de fluxo de dados não estacionários com algoritmos incrementais baseados no modelo de misturas gaussianas*. Master's thesis, Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos, SP.

Pardo, T. (2005). *Métodos para Análise Discursiva Automática*. Ph.D. thesis, University of Sao Paulo.

Pardo, T.; Nunes, M. (2004). Relações retóricas e seus marcadores superficiais: Análise de um corpus de textos científicos em Português do Brasil. Technical Report 231, University of Sao Paulo.

Pardo, T.; Nunes, M. (2008). On the development and evaluation of a Brazilian Portuguese discourse parser. *Journal of Theoretical and Applied Computing*, 15:43–64.

Pardo, T.; Seno, E. (2005). Rhetalho: um corpus de referência anotado retoricamente. In *Proceedings of V Encontro de Corpora.*

Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; Webber, B. (2008). The Penn Discourse TreeBank. In *In Proceedings of LREC.*

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Radev, D. (2000). A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *roceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, pp. 74–83.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the First Empirical Methods in NLP Conference*, pp. 133–142.

Reitter, D. (2003). Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models.

Rosenberg, C.; Hebert, M.; Schneiderman, H. (2005). Semi-supervised selftraining of object detection models. In *Proceedings of Seventh IEEE Workshop on Applications of Computer Vision*, pp. 29–36. IEEE Computer Society, Washington, DC, USA.

Sagae, K. (2009). Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*, pp. 81–84. Association for Computational Linguistics, Paris, France.

Schilder, F. (2002). *Robust discourse parsing via discourse markers, topicality and position*, volume 8, pp. 235–255. Cambridge University Press.

Silva, J.; Branco, A.; Castro, S.; Reis, R. (2010). Out-of-the-box robust parsing of portuguese. In *Proceedings of 9th International Conference on the Computational Processing of Portuguese*, pp. 75–85. Springer-Verlag, Berlin, Heidelberg.

Soricut, R.; Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pp. 149–156.

Spearman, C. (1987). The proof and measurement of association between two things. *The American journal of psychology*, 100(3):441–471.

Sperber, D.; Wilson, D. (1995). *Relevance: Communication and Cognition 2nd Ed.* Blackwell Publishing.

Sporleder, C.; Lascarides, A. (2008). Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416.

Subba, R.; Di Eugenio, B. (2009). An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pp. 566–574.

Sutton, C.; McCallum, A. (2012). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Taboada, M. (2016). Sentiment analysis: An overview from linguistics. *Annual Review of Linguistics*, (2):325–347.

Taboada, M.; Das, D. (2013). Annotation upon annotation: Adding signalling information to a corpus of discourse relations. *Dialogue and Discourse*, 4(2):249–281.

Vapnik, V. N. (1995). *The nature of statistical learning theory.* Springer-Verlag, New York, NY, USA.

Verberne, S.; Boves, L.; Oostdijk, N.; Coppen, P. (2007). Discouse-based answering of why-questions. *Traitement Automatique des Langues, special issue on Computational Approaches to Discourse and Document Processing.*

Webber, B.; Joshi, A. (2012). Discourse structure and computation: Past, present and future. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, ACL '12, pp. 42–54. Association for Computational Linguistics, Stroudsburg, PA, USA.

Wellner, B.; Pustejovsky, J.; Havasi, C.; Rumshisky, A.; Saur, R. (2009). Classification of discourse coherence relations: An exploratory study using multiple knowledge sources. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pp. 117–125.

Witten, I.; Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann Publishers Inc.

Wolf, F.; Gibson, E. (2005). Representing discourse coherence: A corpus-based analysis. *Computational Linguistics*, 31(2):249–287.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196.

Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms.* Chapman E Hall/CRC.

Zhou, Z.-H.; Li, M. (2005). Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17:1529–1541.

Zhu, X. (2008). Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison.

A

Apêndice A: Conjunto das relações discursivas

A.1 Descrição das relações retóricas

Neste apêndice, listam-se todas as relações discursivas utilizadas nesta pesquisa. As definições foram obtidas de Pardo (2005). Esse conjunto de relações foi utilizado por se encontrar nos corpora RST e, durante esta pesquisa, as relações abaixo foram agrupadas, seguindo Mann & Thompson (1987); Soricut & Marcu (2003), conforme a Tabela A.1.

Tabela A.1: **Agrupamento das relações retóricas utilizadas nesta pesquisa**

Grupos de relações RST
Attribution
Antithesis
Concession
Contrast
Background
Circumstance
Volitional-Result
Non-Volitional-Result
Volitional-Cause
Non-Volitional-Cause
Comparison

Condition
Otherwise
Elaboration
Enablement
Motivation
Purpose
Evidence
Justify
Explanation
Interpretation
Evaluation
Conclusion
Solutionhood
List
Joint
Means
Restatement
Same-unit
Sequence
Summary

Tabela A.2: **Definição da relação *Antithesis***

Relação *Antithesis*	
Restrição sobre núcleo (N):	O escritor julga N válido
Restrição sobre o satélite (S):	Nenhuma
Restrição sobre a combinação N e S:	N e S se contrastam e, por esse motivo, não podem ser válidos simultaneamente. Compreendendo-se S inválido levará o leitor a aceitar melhor N
Efeito no receptor:	O leitor aceita melhor N

Tabela A.3: **Definição da relação *Attribution***

Relação *Attribution*	
Restrição sobre núcleo (N):	N apresenta uma expressão, fala ou pensamento de alguém ou algo
Restrição sobre o satélite (S):	S apresenta alguém ou algo que produz N
Restrição sobre a combinação N e S:	S e N indicam, respectivamente, a fonte de uma mensagem e a mensagem
Efeito no receptor:	O leitor é informado sobre a mensagem e sobre quem ou o que a produziu

Tabela A.4: **Definição da relação *Circumstance***

Relação *Circumstance*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Apresenta uma situação (realizável)
Restrição sobre a combinação N e S:	S provê uma situação na qual o leitor pode interpretar N
Efeito no receptor:	O leitor reconhece que S provê uma situação na qual N deve ser interpretado

Tabela A.5: **Definição da relação *Comparison***

Relação *Comparison*	
Restrição sobre núcleo (N):	Apresenta uma característica de algo ou alguém
Restrição sobre o satélite (S):	Apresenta uma característica de algo ou alguém comparável com o que é apresentado em N
Restrição sobre a combinação N e S:	As características de S e N estão em comparação
Efeito no receptor:	O leitor reconhece que S é comparado a N em relação a certas características

Tabela A.6: **Definição da relação *Concession***

Relação *Concession*	
Restrição sobre núcleo (N):	O escritor julga N válido
Restrição sobre o satélite (S):	O escritor não afirma que S pode não ser válido
Restrição sobre a combinação N e S:	O escritor mostra uma incompatibilidade aparente ou em potencial entre N e S; o reconhecimento da compatibilidade entre N e S melhora a aceitação de N pelo leitor
Efeito no receptor:	O leitor aceita melhor N

Tabela A.7: **Definição da relação *Elaboration***

Relação *Elaboration*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta detalhes adicionais sobre a situação ou algum elemento de N
Efeito no receptor:	O leitor reconhece S como apresentando detalhes adicionais sobre N

Tabela A.8: **Definição da relação *Conclusion***

Relação *Conclusion*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	S baseia-se no que é apresentado em N
Restrição sobre a combinação N e S:	S apresenta um fato concluído a partir da interpretação de N
Efeito no receptor:	O leitor reconhece que S é uma conclusão produzida devido à interpretação de N

Tabela A.9: **Definição da relação *Condition***

Relação *Condition*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	S apresenta uma situação hipotética, futura ou não realizada
Restrição sobre a combinação N e S:	A realização de N depende da realização de S
Efeito no receptor:	O leitor reconhece como a realização de N depende da realização de S

Tabela A.10: **Definição da relação *Enablement***

Relação *Enablement*	
Restrição sobre núcleo (N):	Apresenta uma ação do leitor não realizada
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	A compreensão de S pelo leitor aumenta sua habilidade para realizar a ação em N
Efeito no receptor:	A habilidade do leitor para realizar a ação em N aumenta

Tabela A.11: **Definição da relação *Evidence***

Relação *Evidence*	
Restrição sobre núcleo (N):	O leitor poderia não acreditar em N de forma satisfatória para o escritor
Restrição sobre o satélite (S):	O leitor acredita em S ou o achará válido
Restrição sobre a combinação N e S:	A compreensão de S pelo leitor aumenta sua convicção em N
Efeito no receptor:	A convicção do leitor em N aumenta

Tabela A.12: **Definição da relação *Justify***

Relação *Justify*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	A compreensão de S pelo leitor aumenta sua prontidão para aceitar o direito do escritor de apresentar N
Efeito no receptor:	A prontidão do leitor para aceitar o direito do escritor de apresentar N aumenta

Tabela A.13: **Definição da relação *Evaluation***

Relação *Evaluation*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S se relaciona a N pelo grau de avaliação positiva do escritor por N
Efeito no receptor:	O leitor reconhece que S avalia N e reconhece o valor que ele atribui

Tabela A.14: **Definição da relação *Explanation***

Relação *Explanation*	
Restrição sobre núcleo (N):	apresenta um evento ou situação
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S explica como e/ou porque o evento ou situação apresentado em N ocorre ou veio a ocorrer
Efeito no receptor:	O leitor reconhece que S é a razão para N ou que S explica como N ocorre

Tabela A.15: **Definição da relação *Interpretation***

Relação *Interpretation*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta um conjunto de ideias que não é expresso em N propriamente, mas derivado deste
Efeito no receptor:	O leitor reconhece que S apresenta um conjunto de ideias que não é propriamente expresso no conhecimento fornecido por N

Tabela A.16: **Definição da relação *Means***

Relação *Means*	
Restrição sobre núcleo (N):	uma atividade
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta um método ou instrumento que faz com que a realização de N seja mais provável
Efeito no receptor:	O leitor reconhece que o método ou instrumento em S faz com que a realização de N seja mais provável

Tabela A.17: **Definição da relação *Motivation***

Relação *Motivation*	
Restrição sobre núcleo (N):	Uma ação volitiva não realizada
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	A compreensão de S motiva a realização de N
Efeito no receptor:	O leitor reconhece que S motiva a realização de N

Tabela A.18: **Definição da relação *Non-Volitional Cause***

Relação *Non-Volitional Cause*	
Restrição sobre núcleo (N):	Apresenta uma ação não volitiva
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta uma situação que pode ter causado N; sem S, o leitor poderia não reconhecer o que causou a ação em N; N é mais central para a satisfação do objetivo do escritor do que S
Efeito no receptor:	O leitor reconhece a situação apresentada em S como a causa da ação apresentada em N

Tabela A.19: **Definição da relação *Non-Volitional Result***

Relação *Non-Volitional Result*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Apresenta uma ação não volitiva
Restrição sobre a combinação N e S:	N apresenta uma situação que pode ter causado S; sem N, o leitor poderia não reconhecer o que causou a ação em S; N é mais central para a satisfação do objetivo do escritor do que S
Efeito no receptor:	O leitor reconhece a situação apresentada em N como a causa da ação apresentada em S

Tabela A.20: **Definição da relação *Otherwise***

Relação *Otherwise*	
Restrição sobre núcleo (N):	Apresenta uma situação não realizada
Restrição sobre o satélite (S):	Apresenta uma situação não realizada
Restrição sobre a combinação N e S:	A realização de N impede a realização de S
Efeito no receptor:	O leitor reconhece que a realização de N impede a realização de S

Tabela A.21: **Definição da relação *Parenthetical***

Relação *Parenthetical*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Apresenta informação extra relacionada a N que não está expressa no fluxo principal do texto
Restrição sobre a combinação N e S:	S apresenta informação extra relacionada a N, complementado N; S não pertence ao fluxo principal do texto
Efeito no receptor:	O leitor reconhece que S apresenta informação extra relacionada a N, complementando N

Tabela A.22: **Definição da relação *Purpose***

Relação *Purpose*	
Restrição sobre núcleo (N):	Apresenta uma ação
Restrição sobre o satélite (S):	Apresenta uma situação não realizada
Restrição sobre a combinação N e S:	S apresenta uma situação que pode realizar N
Efeito no receptor:	O leitor reconhece que a atividade em N pode ser iniciada por meio de S

Tabela A.23: **Definição da relação *Restatement***

Relação *Restatement*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S se relaciona a N; ambos apresentam conteúdo comparável; N é mais importante para a satisfação do objetivo do escritor
Efeito no receptor:	O leitor reconhece que S expressa o mesmo conteúdo de N, mas de forma diferente

Tabela A.24: **Definição da relação *Solutionhood***

Relação *Solutionhood*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Apresenta um problema
Restrição sobre a combinação N e S:	N é uma solução para o problema em S
Efeito no receptor:	O leitor reconhece N como uma solução para o problema em S

Tabela A.25: **Definição da relação *Summary***

Relação *Summary*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta o conteúdo de N resumido
Efeito no receptor:	O leitor reconhece S como um resumo do conteúdo de N

Tabela A.26: **Definição da relação *Volitional Cause***

Relação *Volitional Cause*	
Restrição sobre núcleo (N):	Apresenta uma ação volitiva ou uma situação que poderia surgir de uma ação volitiva
Restrição sobre o satélite (S):	Não há
Restrição sobre a combinação N e S:	S apresenta uma situação que pode ter acarretado o fato do agente da ação volitiva em N ter realizado a ação; sem S, o leitor poderia não reconhecer a motivação da ação; N é mais central para a satisfação do objetivo do escritor do que S
Efeito no receptor:	O leitor reconhece a situação apresentada em S como a causa da ação apresentada em N

Tabela A.27: **Definição da relação *Volitional Result***

Relação *Volitional Result*	
Restrição sobre núcleo (N):	Não há
Restrição sobre o satélite (S):	Apresenta uma ação volitiva ou uma situação que poderia surgir de uma ação volitiva
Restrição sobre a combinação N e S:	N apresenta uma situação que pode ter acarretado o fato do agente da ação volitiva em S ter realizado a ação; sem N, o leitor poderia não reconhecer a motivação da ação; N é mais central para a satisfação do objetivo do escritor do que S
Efeito no receptor:	O leitor reconhece a situação apresentada em N como a causa da ação apresentada em S

As Tabelas A.28 a A.31 definem as relacões multinucleares.

Tabela A.28: **Definição da relação *Contrast***

Relação *Contrast*	
Restrição sobre os núcleos (Ns):	Não mais do que dois Ns; as situações nos Ns são (a) compreendidas como similares em vários aspectos, (b) compreendidas como diferentes em vários aspectos e (c) comparadas em relação a uma ou mais dessas diferenças
Efeito no receptor:	O leitor reconhece as similaridades e diferenças resultantes da comparação sendo feita

Tabela A.29: **Definição da relação *Joint***

Relação *Joint*	
Restrição sobre os núcleos (Ns):	Não há
Efeito no receptor:	Não há

Tabela A.30: **Definição da relação *List***

Relação *List*	
Restrição sobre os núcleos (Ns):	Itens comparáveis apresentados nos Ns
Efeito no receptor:	O leitor reconhece como comparáveis os itens apresentados

Tabela A.31: **Definição da relação *Same-unit***

Relação *Same-unit*	
Restrição sobre os núcleos (Ns):	Os Ns apresentam informações que, juntas, constituem uma única proposição
Efeito no receptor:	O leitor reconhece que as informações apresentadas constituem uma única proposição; separadas, não fazem sentido

B

Apêndice B: Listas de palavras

Neste apêndice, são apresentadas as listas de palavras utilizadas durante esta pesquisa, tanto no processo de segmentação discursiva, quanto na extração de atributos para a identificação de relações discursivas e suas nuclearidades. Essas listas foram extraídas do *parser* DiZer (Pardo, 2005).

B.1 Verbos de atribuição

Os verbos de atribuição foram úteis no processo de segmentação, conforme as regras apresentadas no Capítulo 5, considerando a existência da relação *Attribution*. Essas palavras também foram utilizadas nas regras criadas manualmente para a identificação da referida relação retórica. O uso dessas palavras depende da lematização do texto, pois na lista estão apenas as formas lematizadas das palavras, sem flexão.

1. *afirmar*
2. *acrescentar*
3. *anunciar*
4. *citar*
5. *comentar*
6. *complementar*

7. *comunicar*
8. *contar*
9. *declamar*
10. *declarar*
11. *demonstrar*
12. *destacar*
13. *dizer*
14. *enfatizar*
15. *expressar*
16. *explicar*
17. *falar*
18. *frisar*
19. *informar*
20. *lembrar*
21. *mencionar*
22. *negar*
23. *noticiar*
24. *notificar*
25. *observar*
26. *proferir*
27. *relatar*

28. *reportar*

29. *ressaltar*

30. *resumir*

31. *retificar*

32. *salientar*

33. *sugerir*

B.2 Conjunções

As conjunções, seus tipos e subtipos estão apresentadas na Tabela B.1. Os tipos das conjunções consistem em *Coordenativa* ou *Subordinativa*, de acordo com a função sintática da conjunção. Os subtipos podem ser *Aditiva*, *Adversativa*, *Alternativa*, *Conclusiva* ou *Explicativa* para as *Coordenativas*. Como subtipos das conjunções *Subordinativas* tem-se*Causal*, *Comparativa*, *Concessiva*, *Condicional*, *Conformativa*, *Consecutiva*, *Final*, *Proporcional* ou *Temporal*.

Tabela B.1: **Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas.**

Conjunção	Tipo	Subtipo
e	Coordenativa	Aditiva
nem	Coordenativa	Aditiva
também	Coordenativa	Aditiva
como também	Coordenativa	Aditiva
bem como	Coordenativa	Aditiva
mas ainda	Coordenativa	Aditiva
não só	Coordenativa	Aditiva
mas	Coordenativa	Adversativa
entretanto	Coordenativa	Adversativa

Tabela B.1: **Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas.**

Conjunção	Tipo	Subtipo
no entanto	Coordenativa	Adversativa
porém	Coordenativa	Adversativa
todavia	Coordenativa	Adversativa
contudo	Coordenativa	Adversativa
não obstante	Coordenativa	Adversativa
ou	Coordenativa	Alternativa
ora	Coordenativa	Alternativa
já	Coordenativa	Alternativa
quer	Coordenativa	Alternativa
seja	Coordenativa	Alternativa
logo	Coordenativa	Conclusiva
portanto	Coordenativa	Conclusiva
por isso	Coordenativa	Conclusiva
assim	Coordenativa	Conclusiva
por conseguinte	Coordenativa	Conclusiva
então	Coordenativa	Conclusiva
que	Coordenativa	Explicativa
porque	Coordenativa	Explicativa
porquanto	Coordenativa	Explicativa
pois	Coordenativa	Explicativa
porque	Subordinativa	Causal
uma vez que	Subordinativa	Causal
sendo que	Subordinativa	Causal
visto que	Subordinativa	Causal
como	Subordinativa	Causal
já que	Subordinativa	Causal

Tabela B.1: **Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas.**

Conjunção	Tipo	Subtipo
desde que	Subordinativa	Causal
pois	Subordinativa	Causal
como	Subordinativa	Comparativa
tal qual	Subordinativa	Comparativa
que	Subordinativa	Comparativa
do que	Subordinativa	Comparativa
assim como	Subordinativa	Comparativa
mais	Subordinativa	Comparativa
menos	Subordinativa	Comparativa
tanto quanto	Subordinativa	Comparativa
mesmo que	Subordinativa	Concessiva
por mais que	Subordinativa	Concessiva
ainda que	Subordinativa	Concessiva
ainda quando	Subordinativa	Concessiva
quando mesmo	Subordinativa	Concessiva
se bem que	Subordinativa	Concessiva
embora	Subordinativa	Concessiva
conquanto	Subordinativa	Concessiva
posto que	Subordinativa	Concessiva
por muito que	Subordinativa	Concessiva
apesar de que	Subordinativa	Concessiva
que	Subordinativa	Concessiva
malgrado	Subordinativa	Concessiva
dado que	Subordinativa	Concessiva
suporte que	Subordinativa	Concessiva
se	Subordinativa	Condicional

Tabela B.1: **Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas.**

Conjunção	Tipo	Subtipo
caso	Subordinativa	Condicional
contanto que	Subordinativa	Condicional
a menos que	Subordinativa	Condicional
sem que	Subordinativa	Condicional
salvo se	Subordinativa	Condicional
desde que	Subordinativa	Condicional
conforme	Subordinativa	Conformativa
assim como	Subordinativa	Conformativa
segundo	Subordinativa	Conformativa
consoante	Subordinativa	Conformativa
como	Subordinativa	Conformativa
de acordo com	Subordinativa	Conformativa
tal que	Subordinativa	Consecutiva
tão que	Subordinativa	Consecutiva
tanto que	Subordinativa	Consecutiva
tamanho que	Subordinativa	Consecutiva
sem que	Subordinativa	Consecutiva
de modo que	Subordinativa	Consecutiva
de forma que	Subordinativa	Consecutiva
de maneira que	Subordinativa	Consecutiva
a fim de que	Subordinativa	Final
para que	Subordinativa	Final
à medida que	Subordinativa	Proporcional
à proporção que	Subordinativa	Proporcional
ao passo que	Subordinativa	Proporcional
quanto mais	Subordinativa	Proporcional

Tabela B.1: **Listagem das conjunções utilizadas durante o processo de extração de atributos para identificação das relações retóricas.**

Conjunção	Tipo	Subtipo
quanto menos	Subordinativa	Proporcional
quando	Subordinativa	Temporal
enquanto	Subordinativa	Temporal
sempre que	Subordinativa	Temporal
logo	Subordinativa	Temporal
que	Subordinativa	Temporal
depois que	Subordinativa	Temporal
desde que	Subordinativa	Temporal
assim que	Subordinativa	Temporal
até que	Subordinativa	Temporal
cada vez que	Subordinativa	Temporal
sem que	Subordinativa	Temporal