UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Expansão de recursos para análise de sentimentos usando aprendizado semissupervisionado

Henrico Bertini Brum

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

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Henrico Bertini Brum

Expansão de recursos para análise de sentimentos usando aprendizado semissupervisionado

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maria das Graças Volpe Nunes

USP – São Carlos
Maio de 2018

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

B893e Brum, Henrico Bertini
 Expansão de recursos para análise de sentimentos
 usando aprendizado semissupervisionado / Henrico
 Bertini Brum; orientadora Maria das Graças Volpe
 Nunes. -- São Carlos, 2018.
 173 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2018.

 1. Análise de Sentimentos. 2. Anotação de Córpus.
 3. Aprendizado Semissupervisionado. I. Nunes, Maria
 das Graças Volpe, orient. II. Título.

Henrico Bertini Brum

**Extending sentiment analysis resources using
semi-supervised learning**

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

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maria das Graças Volpe Nunes

USP – São Carlos
May 2018

Dedico esse trabalho a Marcia Cristia Cera.
Uma vez ela dançou uma música ridícula na sala pra ensinar um conteúdo.
Sempre soube que existiam bons professores,
mas ali eu percebi que pra ser o melhor tem que amar muito o que faz.
Ela amou. E foi.

AGRADECIMENTOS

Nenhum trabalho bem feito pode ser baseado em somente um indivíduo, apesar do meu nome na capa, muitas pessoas foram essenciais para a conclusão dessa etapa - tanto no contexto acadêmico, quando social, quanto pessoal. Abaixo agradeço alguns grupos que foram pilares na minha jornada, digo grupos por que nomear pessoas sempre carece de precisão e pode gerar reações desnecessárias - todos que estiveram comigo sabem que tem o seu valor na minha vida.

Primeiro ao meu círculo familiar - a família porto-alegrense, a família bageense e a família uruguaianense. Com destaque para os meus pais, sem eles seria impossível eu fazer qualquer coisa (desde caminhar até escrever esses agradecimentos) e a minha namorada que teve que aguentar muita ansiedade nesse processo (e esperamos que eu não tenha que rasurar essa parte no futuro :P). Vocês foram meus verdadeiros pilares mentais.

Meu círculo social, amigos das tarefas espíritas que me acolheram em São Carlos e me deram muito trabalho pra eu aliviar a pressão acadêmica. Pessoal do Paz e Harmonia, do DM-São Carlos e do DM3, sem vocês eu ia ter terminado esse mestrado muito antes, mas eu teria sido uma pessoa muito mais infeliz no processo - e ao meu ver de nada vale trabalhar se não for com felicidade.

Por último ao eixo acadêmico que me deu um suporte intelectual muito bom, ao NILC que também me acolheu, me fez sentir parte de um grupo e me introduziu a pesquisadores sensacionais. Obrigado a todos do laboratório, do ICMC e de todos que eu tive contato nesses anos. Destaco aqui meu orientador da graduação que me ajudou muito com indicações, dicas e alertas; ao time que me ajudou na anotação do TweetSentBR, ainda devo chocolate pra eles; e a minha orientadora que teve que aturar meus atrasos, meu prazos estourados e meu péssimo hábito de não responder e-mails de vez em quando (juro que vou mudar) - sem ela esse trabalho não teria nenhuma Graça.

Ignorando a minha regra, três agradecimentos nominais para a minha família de São Carlos. Por que apesar de amigos, namorada, pais e orientadores, quando batia a ansiedade as três da manhã, eram eles que faziam o café (ruim, diga-se de passagem) e trocavam ideia pra que tudo fosse mais tolerável - Carlos Ronchi, Marcos Treviso e Thales Bertaglia.

RESUMO

BRUM, H. B. **Expansão de recursos para análise de sentimentos usando aprendizado semisupervisionado**. 2018. 173 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

O grande volume de dados que temos disponíveis em ambientes virtuais pode ser excelente fonte de novos recursos para estudos em diversas tarefas de Processamento de Linguagem Natural, como a Análise de Sentimentos. Infelizmente é elevado o custo de anotação de novos corpú, que envolve desde investimentos financeiros até demorados processos de revisão. Nossa pesquisa propõe uma abordagem de anotação semissupervisionada, ou seja, anotação automática de um grande corpú não anotado partindo de um conjunto de dados anotados manualmente. Para tal, introduzimos o TweetSentBR, um corpú de *tweets* no domínio de programas televisivos que possui anotação em três classes e revisões parciais feitas por até sete anotadores. O corpú representa um importante recurso linguístico de português brasileiro, e fica entre os maiores corpú anotados na literatura para classificação de polaridades. Além da anotação manual do corpú, realizamos a implementação de um *framework* de aprendizado semissupervisionado que faz uso de dados anotados e, de maneira iterativa, expande o mesmo usando dados não anotados. O *TweetSentBR*, que possui 15.000 *tweets* anotados é assim expandido cerca de oito vezes. Para a expansão, foram treinados modelos de classificação usando seis classificadores de polaridades, assim como foram avaliados diferentes parâmetros e representações a fim de obter um corpú confiável. Realizamos experimentos gerando corpú expandidos por cada classificador, tanto para a classificação em três polaridades (positiva, neutra e negativa) quanto para classificação binária. Avaliamos os corpú gerados usando um conjunto de *held-out* e comparamos a *F-Measure* da classificação usando como treinamento os corpú anotados manualmente e semi-automaticamente. O corpú semissupervisionado que obteve os melhores resultados para a classificação em três polaridades atingiu 62,14% de *F-Measure* média, superando a média obtida com as avaliações no corpú anotado manualmente (61,02%). Na classificação binária, o melhor corpú expandido obteve 83,11% de *F1-Measure* média, superando a média obtida na avaliação do corpú anotado manualmente (79,80%). Além disso, simulamos nossa expansão em corpú anotados da literatura, medindo o quão corretas são as etiquetas anotadas semi-automaticamente. Nosso melhor resultado foi na expansão de um corpú de *reviews* de produtos que obteve *F-Measure* de 93,15% com dados binários. Por fim, comparamos um corpú da literatura obtido por meio de supervisão distante e nosso *framework* semissupervisionado superou o primeiro na classificação de polaridades binária em *cross-domain*.

Palavras-chave: Análise de Sentimentos, Anotação de Corpú, Aprendizado Semissupervisionado.

ABSTRACT

BRUM, H. B. **Extending sentiment analysis resources using semi-supervised learning**. 2018. 173 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

The high volume of data available in the Internet can be a good resource for studies of several tasks in Natural Language Processing as in Sentiment Analysis. Unfortunately there is a high cost for the annotation of new corpora, involving financial support and long revision processes. Our work proposes an approach for semi-supervised labeling, an automatic annotation of a large unlabeled set of documents starting from a manually annotated corpus. In order to achieve that, we introduced TweetSentBR, a tweet corpora on TV show programs domain with annotation for 3-point (positive, neutral and negative) sentiment classification partially reviewed by up to seven annotators. The corpus is an important linguistic resource for Brazilian Portuguese language and it stands between the biggest annotated corpora for polarity classification. Beyond the manual annotation, we implemented a semi-supervised learning based framework that uses this labeled data and extends it using unlabeled data. TweetSentBR corpus, containing 15.000 documents, had its size augmented in eight times. For the extending process, we trained classification models using six polarity classifiers, evaluated different parameters and representation schemes in order to obtain the most reliable corpora. We ran experiments generating extended corpora for each classifier, both for 3-point and binary classification. We evaluated the generated corpora using a held-out subset and compared the obtained F-Measure values with the manually and the semi-supervised annotated corpora. The semi-supervised corpus that obtained the best values for 3-point classification achieved 62,14% on average F-Measure, overcoming the results obtained by the same classification with the manually annotated corpus (61,02%). On binary classification, the best extended corpus achieved 83,11% on average F-Measure, overcoming the results on the manually corpora (79,80%). Furthermore, we simulated the extension of labeled corpora in literature, measuring how well the semi-supervised annotation works. Our best results were in the extension of a product review corpora, achieving 93,15% on F1-Measure. Finally, we compared a literature corpus which was labeled by using distant supervision with our semi-supervised corpus, and this overcame the first in binary polarity classification on cross-domain data.

Keywords: Sentiment Analysis, Corpus Annotation, Semi-Supervised Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Recorte de tela do <i>Twitter</i> mostrando uma busca por palavra-chave e os <i>Trending Topics</i>	6
Figura 2 – <i>Review</i> de produto com polaridade negativa em nível de documento.	11
Figura 3 – <i>Review</i> de produto analisado em nível de sentença.	12
Figura 4 – <i>Review</i> de produto analisado em nível de entidade ou aspecto.	12
Figura 5 – Exemplo da Rede Neural Recursiva com Tensor classificando um trecho quanto à polaridade.	19
Figura 6 – Documentos representando mesmo domínio (<i>review</i> de filme) e gênero opinativo.	24
Figura 7 – Exemplo de aprendizado semissupervisionado em um espaço bi-dimensional.	30
Figura 8 – Independência condicional na divisão de <i>features</i> do <i>co-training</i>	33
Figura 9 – Interface de anotação contendo 25 possíveis valores em uma barra deslizante.	37
Figura 10 – Instruções e interface de anotação de <i>tweets</i> para o <i>Mechanical Turk</i>	39
Figura 11 – Trecho exemplo do <i>corpus</i> ReLi.	41
Figura 12 – Lista de <i>emojis</i> usados na Supervisão Distante no <i>corpus</i> Pelesent.	43
Figura 13 – Conjunto de palavras de negação, intensificação e redução.	52
Figura 14 – Fluxograma do <i>framework</i> de expansão de <i>corpus</i> proposto.	60
Figura 15 – Fluxograma da primeira etapa do <i>framework</i>	62
Figura 16 – Exibição do programa <i>Master Chef Brasil</i> na emissora Bandeirantes.	63
Figura 17 – Ambiente de anotação para a revisão do <i>corpus</i> de teste.	67
Figura 18 – Emoticons positivos e negativos usados na representação dos documentos.	74
Figura 19 – Exemplo de anotação no <i>Emoji Sentiment Ranking</i>	75
Figura 20 – Segunda etapa do <i>framework</i> de expansão de <i>corpus</i>	79
Figura 21 – Gráfico de <i>F-Measure</i> média obtida com cada <i>corpus</i> expandido usando <i>self-training</i> na avaliação em três classes.	97
Figura 22 – Gráfico de <i>F-Measure</i> média obtida com cada <i>corpus</i> expandido usando <i>self-training</i> na avaliação binária.	99
Figura 23 – Gráfico de <i>F-Measure</i> média obtida com cada <i>corpus</i> expandido com o TTsBR reduzido usando <i>self-training</i> na avaliação em três classes.	101
Figura 24 – Gráfico de <i>F-Measure</i> média obtida com cada <i>corpus</i> expandido com o TTsBR reduzido usando <i>self-training</i> na avaliação binária.	102
Figura 25 – Tela de <i>login</i> da interface de anotação.	127
Figura 26 – Painel principal da interface de anotação.	128

Figura 27 – Manual de anotação da interface de anotação.	129
Figura 28 – Primeiro acesso ao ambiente de anotação.	130
Figura 29 – Documentos anotados na interface de anotação.	130
Figura 30 – Painel de estatísticas da interface de anotação.	131

LISTA DE TABELAS

Tabela 1 – Tipos de opiniões em tarefas de AS (LIU, 2012).	14
Tabela 2 – Matriz-confusão	20
Tabela 3 – Estatísticas dos corpú do SemEval de 2013 até 2015.	38
Tabela 4 – Resumo de corpú de sentimentos em inglês encontrados na literatura.	40
Tabela 5 – Resumo de corpú de sentimentos em português encontrados na literatura.	46
Tabela 6 – Resultados dos trabalhos apresentados avaliados no corpú <i>Stanford Sentiment Treebank</i> .	49
Tabela 7 – Resultados de acurácia e ARI para os dados..	50
Tabela 8 – Resumo dos trabalhos de classificação de polaridades para a língua portuguesa.	57
Tabela 9 – Programas de televisão presentes na extração e suas ocorrências no corpú.	64
Tabela 10 – Dados do corpú compilado.	68
Tabela 11 – Distribuição de <i>hashtags</i> no TTsBR.	69
Tabela 12 – Termos mais relevantes para cada classe no TTsBR usando TF-idf.	69
Tabela 13 – Visão geral dos corpú de SA para Português Brasileiro em comparação ao TTsBR.	82
Tabela 14 – Avaliação do TTsBR na classificação de polaridades em três classes usando configuração Treino/Teste.	83
Tabela 15 – Avaliação do TTsBR na classificação binária de polaridades usando configuração Treino/Teste.	84
Tabela 16 – Classificação de polaridades em três classes com os corpú ReLi e TTsBR.	85
Tabela 17 – Comparação de classificação de corpú usando SVM com duas classes.	85
Tabela 18 – Comparação de classificação de corpú usando Naive Bayes com duas classes.	86
Tabela 19 – Comparação de classificação de corpú usando Regressão Logística com duas classes.	87
Tabela 20 – Comparação de classificação de corpú usando <i>Multilayer Perceptron</i> com duas classes.	87
Tabela 21 – Comparação de classificação de corpú usando Árvore de Decisão com duas classes.	88
Tabela 22 – Comparação de classificação de corpú usando Random Forest com duas classes.	88
Tabela 23 – Visão geral da classificação de polaridades nos corpú da literatura (corpú ReLi e TTsBR possuem duas entradas assinaladas com * para indicar classificação em três classes).	89

Tabela 24 – Melhores resultados obtidos na avaliação de hiperparâmetros com classifica- dor SVM.	92
Tabela 25 – Melhores resultados obtidos na avaliação de hiperparâmetros com classifica- dor Naive Bayes.	93
Tabela 26 – Melhores resultados obtidos na avaliação de hiperparâmetros com Regressão Logística.	93
Tabela 27 – Melhores resultados obtidos na avaliação de hiperparâmetros com <i>Multilayer Perceptron</i>	93
Tabela 28 – Melhores resultados obtidos na avaliação de hiperparâmetros com Árvores de Decisão.	94
Tabela 29 – Melhores resultados obtidos na avaliação de hiperparâmetros com <i>Random Forest</i>	95
Tabela 30 – Resumo da <i>F-Measure</i> média obtida com os corpúscos expandidos usando <i>self-training</i> com três classes.	96
Tabela 31 – Resumo da <i>F-Measure</i> média obtida com os corpúscos expandidos usando <i>self-training</i> na classificação binária.	99
Tabela 32 – Resumo da <i>F-Measure</i> média obtida com os corpúscos expandidos usando <i>self-training</i> com o TTsBR reduzido na classificação em três classes.	100
Tabela 33 – Resumo da <i>F-Measure</i> média obtida com os corpúscos expandidos usando <i>self-training</i> com o TTsBR reduzido na classificação binária.	101
Tabela 34 – Comparação com o TTsBR expandido usando MLP com <i>threshold</i> de 30% com o corpúscos Pelesent na avaliação de <i>cross-domain</i> com os corpúscos da literatura.	106
Tabela 35 – Distribuição das classes nos corpúscos expandidos usando <i>self-training</i> a partir do TTsBR completo.	107
Tabela 36 – Distribuição das classes nos corpúscos expandidos usando <i>self-training</i> a partir do TTsBR reduzido.	108
Tabela 37 – <i>Self-training</i> adicionando 40% dos documentos por iteração e avaliando com três classes.	138
Tabela 38 – <i>Self-training</i> adicionando 30% dos documentos por iteração e avaliando com três classes.	139
Tabela 39 – <i>Self-training</i> adicionando 25% dos documentos por iteração e avaliando com três classes.	140
Tabela 40 – <i>Self-training</i> adicionando 20% dos documentos por iteração e avaliando com três classes.	141
Tabela 41 – <i>Self-training</i> adicionando 10% dos documentos por iteração e avaliando com três classes.	142
Tabela 42 – <i>Self-training</i> adicionando 5% dos documentos por iteração e avaliando com três classes.	143

Tabela 43 – <i>Self-training</i> adicionando 1% dos documentos por iteração e avaliando com três classes.	144
Tabela 44 – <i>Self-training</i> adicionando 40% dos documentos por iteração na classificação binária.	145
Tabela 45 – <i>Self-training</i> adicionando 30% dos documentos por iteração na classificação binária.	146
Tabela 46 – <i>Self-training</i> adicionando 25% dos documentos por iteração na classificação binária.	147
Tabela 47 – <i>Self-training</i> adicionando 20% dos documentos por iteração na classificação binária.	148
Tabela 48 – <i>Self-training</i> adicionando 10% dos documentos por iteração na classificação binária.	149
Tabela 49 – <i>Self-training</i> adicionando 5% dos documentos por iteração na classificação binária.	150
Tabela 50 – <i>Self-training</i> adicionando 1% dos documentos por iteração na classificação binária.	151
Tabela 51 – <i>Self-training</i> adicionando 40% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	154
Tabela 52 – <i>Self-training</i> adicionando 30% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	155
Tabela 53 – <i>Self-training</i> adicionando 25% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	156
Tabela 54 – <i>Self-training</i> adicionando 20% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	157
Tabela 55 – <i>Self-training</i> adicionando 10% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	158
Tabela 56 – <i>Self-training</i> adicionando 5% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	159
Tabela 57 – <i>Self-training</i> adicionando 1% dos documentos por iteração com TTsBR reduzido avaliando com três classes.	160
Tabela 58 – <i>Self-training</i> adicionando 40% dos documentos por iteração com TTsBR reduzido na classificação binária.	161
Tabela 59 – <i>Self-training</i> adicionando 30% dos documentos por iteração com TTsBR reduzido na classificação binária.	162
Tabela 60 – <i>Self-training</i> adicionando 25% dos documentos por iteração com TTsBR reduzido na classificação binária.	163
Tabela 61 – <i>Self-training</i> adicionando 20% dos documentos por iteração com TTsBR reduzido na classificação binária.	164

Tabela 62 – <i>Self-training</i> adicionando 10% dos documentos por iteração com TTsBR reduzido na classificação binária.	165
Tabela 63 – <i>Self-training</i> adicionando 5% dos documentos por iteração com TTsBR reduzido na classificação binária.	166
Tabela 64 – <i>Self-training</i> adicionando 1% dos documentos por iteração com TTsBR reduzido na classificação binária.	167
Tabela 65 – Avaliação do <i>framework</i> de expansão usando o corpus Buscapé-2.	170
Tabela 66 – Avaliação do <i>framework</i> de expansão usando o corpus Mercado Livre.	171
Tabela 67 – Avaliação do <i>framework</i> de expansão usando o corpus Eleições-Dilma.	172
Tabela 68 – Avaliação do <i>framework</i> de expansão usando o corpus Eleições-Serra.	173

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
AS	Análise de Sentimentos
CGU	Conteúdo Gerado por Usuário
CMFS	<i>Comprehensive Measurement Feature Selection</i>
IR	<i>Information Retrieval</i>
PLN	Processamento de Linguagem Natural
SST	<i>Stanford Sentiment Treebank</i>
SVM	<i>Support Vector Machines</i>
TF-IDF	<i>Term frequency – Inverse Document Frequency</i>

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Motivação e contextualização	1
1.2	Objetivos	7
1.3	Organização da monografia	8
2	FUNDAMENTAÇÃO	9
2.1	Análise de Sentimentos	9
2.1.1	<i>Classificação de polaridade</i>	13
2.1.2	<i>Abordagens e métodos</i>	15
2.1.3	<i>Avaliações e Métricas</i>	20
2.2	Córpus de Sentimentos	22
2.3	Aprendizado Semissupervisionado	29
2.3.1	<i>Self-training</i>	30
2.3.2	<i>Co-training</i>	32
3	TRABALHOS RELACIONADOS	35
3.1	Córpus de Sentimentos	35
3.1.1	<i>Córpus em Inglês</i>	35
3.1.2	<i>Córpus em Português Brasileiro</i>	40
3.2	Análise de Sentimentos	46
3.2.1	<i>Métodos usados para inglês</i>	46
3.2.2	<i>Métodos usados para português brasileiro</i>	51
3.3	Discussão da Revisão	56
4	FRAMEWORK DE EXPANSÃO DE RECURSOS	59
4.1	Extração de dados e anotação	61
4.1.1	<i>Extração de Tweets</i>	62
4.1.2	<i>Anotação Manual</i>	64
4.1.3	<i>Revisão da anotação</i>	66
4.1.4	<i>O TweetSentBR</i>	68
4.2	Classificação e expansão	70
4.2.1	<i>Pré-processamento</i>	70
4.2.2	<i>Modelagem dos Dados</i>	73
4.2.3	<i>Métodos de Classificação</i>	77

4.2.4	<i>Expansão de Corpus</i>	79
5	EXPERIMENTOS	81
5.1	Experimentos com o TTsBR	81
5.1.1	<i>Comparativo na tarefa de classificação de polaridades</i>	83
5.1.2	<i>Resumo dos experimentos com o TTsBR</i>	88
5.2	Experimentos com expansão semissupervisionada	89
5.2.1	<i>Seleção de features e hiperparâmetros</i>	91
5.2.1.1	<i>SVM Linear</i>	92
5.2.1.2	<i>Naive Bayes</i>	92
5.2.1.3	<i>Regressão Logística</i>	92
5.2.1.4	<i>Multilayer Perceptron</i>	93
5.2.1.5	<i>Árvore de Decisão</i>	93
5.2.1.6	<i>Random Forest</i>	94
5.2.1.7	<i>Resumo dos hiperparâmetros escolhidos</i>	94
5.2.2	<i>Framework de Expansão com o TTsBR completo</i>	96
5.2.3	<i>Framework de Expansão com o TTsBR reduzido</i>	99
5.2.4	<i>Framework de expansão com corpus da literatura</i>	103
5.2.4.1	<i>Corpus Buscapé-2</i>	103
5.2.4.2	<i>Corpus Mercado Livre</i>	104
5.2.4.3	<i>Corpus Eleições-Dilma</i>	104
5.2.4.4	<i>Corpus Eleições-Serra</i>	105
5.3	Comparação do TTsBR expandido com corpus anotado por super- visão distante	105
6	CONCLUSÕES E TRABALHOS FUTUROS	109
	REFERÊNCIAS	111
	APÊNDICE A MANUAL DE ANOTAÇÃO	123
A.1	Resumo	123
A.2	Origem dos dados	123
A.3	Tarefa	124
A.4	Exemplos	125
	APÊNDICE B TELAS DA INTERFACE DE ANOTAÇÃO	127
B.1	Tela de login	127
B.2	Painel principal	128
B.3	Manual de anotação	128
B.4	Ambiente de anotação	129

B.4.1	<i>Painel de estatísticas</i>	130
APÊNDICE C	<i>TWEETS COM EMPATE NA ANOTAÇÃO</i>	133
C.1	Durante a fase de concordância (7 anotadores)	133
C.2	Durante a fase de revisão (3 anotadores)	134
APÊNDICE D	RESULTADOS DO <i>FRAMEWORK</i> DE EXPANSÃO COM TTSBR COMPLETO	137
APÊNDICE E	RESULTADOS DO <i>FRAMEWORK</i> DE EXPANSÃO COM TTSBR REDUZIDO	153
APÊNDICE F	AVALIAÇÃO DO <i>FRAMEWORK</i> DE EXPANSÃO NOS CÓRPUS DA LITERATURA	169

INTRODUÇÃO

1.1 Motivação e contextualização

Os avanços tecnológicos têm proporcionado sistemas computacionais com comportamento cada vez mais semelhante ao de seres humanos. Isso fica evidente no uso cada vez mais crescente de dispositivos móveis interativos, como *smartphones*, *tablets* e similares. Essa exigência por maior interatividade trouxe mais demanda para a área de Processamento de Linguagem Natural (PLN), que trata do processamento computacional da língua humana escrita ou falada.

Os desafios para PLN, no entanto, não são recentes. Desde os primórdios da Computação, os desafios em lidar com aplicações complexas, como a Tradução Automática, têm alavancado as pesquisas nessa área. Primeiramente com o desenvolvimento de teorias de embasamento linguístico, e, posteriormente, por meio de técnicas de Aprendizado de Máquina (AM), diversas tarefas necessárias ao tratamento da língua, como análise morfossintática, sintática, desambiguação lexical, entre outras, têm alcançado patamares de precisão que possibilitam o desenvolvimento de sistemas cada vez mais ambiciosos. A natureza ambígua e subjetiva da língua humana, no entanto, faz com que questões de natureza semântica devam continuar desafiando os pesquisadores por muito tempo.

A demanda imposta pelo crescente uso de redes sociais sugere um processamento da língua com ênfase no tratamento semântico. De modo geral, as aplicações sobre os Conteúdos Gerados por Usuários (CGU) estão interessadas mais na mensagem veiculada do que na sua forma. Interessa saber, do CGU, do que se trata, sobre o que se fala, se se trata de uma crítica ou um elogio, etc. Em alguma medida essas tarefas tratam de significado, e estão entre os desafios mais complexos de PLN.

Esse trabalho se insere na área de Análise de Sentimentos, que tem por objetivo determinar sentimentos ou emoções vinculados a uma manifestação linguística escrita ou falada. Análise de Sentimentos (AS) é uma área que surgiu na década de 90 ainda sobre a alcunha de *Affective*

Computing, ou Computação Afetiva. O próprio termo, introduzido em [Picard \(1997\)](#), difere muito do que se entende hoje por AS, apresentando uma visão exploratória no uso de emoções agregadas a sistemas inteligentes usados na época, e como essas emoções podem ajudar nas tomadas de decisões mais inteligentes, ou para uma melhor interação com usuários.

No início dos anos 2000, alguns trabalhos já se interessavam pelo estudo de emoções em texto para a classificação de *reviews* de usuários. *Reviews* são textos contendo a opinião de um usuário acerca de uma entidade de um determinado domínio (produtos, filmes, carros). [Turney \(2002\)](#) apresentou um método para classificação de “orientação semântica”, que é a intenção subjacente a um *review* de indicar recomendação ou não do produto avaliado. No mesmo período [Pang, Lee e Vaithyanathan \(2002\)](#) formaram um conjunto de dados com *reviews* de usuários sobre filmes e aplicaram técnicas de classificação para identificar positividade ou negatividade nos mesmos, tal tarefa foi tratada como Classificação de Polaridade.

A motivação para essas pesquisas fortificou-se pela facilidade de aquisição de bases de dados para experimentos, visto que a internet já disponibilizava plataformas de interação com usuários para avaliações de produtos e serviços. Essas avaliações geralmente contêm *reviews* de usuários e uma medida de avaliação, como notas ou estrelas, para os mesmos discretizarem suas opiniões.

Diversos trabalhos surgiram abrindo o leque de aplicações de AS, entre eles, a sumarização de opiniões ([HU; LIU, 2004](#)), que busca classificar e agrupar *reviews* mais importantes sobre um produto ou serviço, destacando elogios e críticas; a análise de posicionamento político ([LAVER; BENOIT; GARRY, 2003](#)), para medir intenção de votos; a detecção de discursos de ódio ([WARNER; HIRSCHBERG, 2012](#)) ou preconceito; a análise de *tweets* para analisar reações de audiências, entre outros. A própria classificação de polaridade sofisticou-se com a possibilidade de lidar com multiclasse ([PANG; LEE, 2005](#)), agregando ao par positivo/negativo as classes “Muito Positivo”, “Muito Negativo” e “Neutro”.

Esse aquecimento da área se deve ao crescimento da internet e da modernização de suas interfaces e ambientes virtuais, além da popularização das redes sociais que transformaram usuários ‘passivos’ em geradores de conteúdo na forma de imagens, textos ou vídeos. Esses ambientes mostraram-se excelentes catalisadores de bases de dados para tarefas de PLN, e a AS se valeu de redes sociais de domínios específicos (livros, filmes, produtos, finanças) e genéricos (*Facebook*, *Twitter*) para avaliar diversas técnicas de classificação e estudar diversos fenômenos envolvendo opiniões e sentimentos em geral. Essas bases de dados contendo textos anotados com a polaridade associada são chamadas de *Córpus de Sentimentos*.

Por constituírem sua fonte as expressões linguísticas, a Análise de Sentimentos depende, em geral, de ferramentas linguístico-computacionais, como etiquetador morfossintático, léxico de sentimentos (palavras associadas às suas polaridades), reconhecedor de entidades nomeadas, desambiguador lexical, entre outros ([LIU, 2012](#)), além de ferramentas mais simples, como tokenizadores e segmentadores de sentenças.

A linguagem usada por usuários de redes sociais tem características próprias, que desafiam o bom funcionamento de ferramentas desenvolvidas para linguagens usadas em ambientes mais formais. Ao tratar GCU, um sistema deve ser capaz de identificar gírias (ex. *mano*, *twittar*), palavras típicas de “internetês” (ex. *vc*), diferentes tipos de erro (ortográfico, sintático, pontuação), de lidar com uma semântica dinâmica (expressões mudam de sentido ou desaparecem com o tempo), com economias morfológicas propositas (caso do *Twitter* que possui limitação de caracteres), influências fonéticas ou vocalizações (ex. “*feicebuq*”), a presença constante de ironia e sarcasmo, entre outros. Essas ocorrências tendem a diminuir a eficiência de outras ferramentas auxiliares, como etiquetadores morfossintáticos e *parsers* por dependência.

Apesar dos desafios, a maioria dos trabalhos da literatura apresenta *córpus* de sentimentos oriundos de redes sociais (PANG; LEE, 2008; PAK; PAROUBEK, 2010; JUNIOR *et al.*, 2017), mostrando que são ambientes ricos em opiniões. O anonimato e a distância derivados do ambiente virtual faz com que as críticas dos usuários em *reviews*, por exemplo, sejam mais sinceras – e também mais ácidas, sarcásticas e agressivas – o que pode ser muito útil tanto para consumidores que buscam informações sobre um serviço ou produto, como para empresas que buscam avaliar a aceitação de um produto novo.

A utilização destes dados torna-se dependente de uma anotação de sentimento (polaridade binária ou multiclasse). Por sua característica subjetiva, a classificação da polaridade pode variar de usuário para usuário. Em alguns casos, o próprio site de comentários fornece opções de classificação por meio de nota ou estrelas. Essa disposição apresenta graus de similaridade para os *reviews* – dizemos que “duas estrelas” está mais próximo de ‘quatro estrelas’ do que de ‘cinco estrelas’ (PANG; LEE, 2005). A dificuldade aparece quando notamos que essa similaridade é subjetiva para cada usuário e, portanto, opiniões similares podem gerar classificações distintas.

A alternativa ao uso desses valores discretos é a anotação manual dessas polaridades. Um processo humano de anotação, como descrito em Hovy e Lavid (2010), demanda criação de manuais, garantia de representatividade e concordância, além de treinamentos e revisões de anotação, o que torna o processo muito custoso.

Esses *córpus* de sentimentos podem variar conforme a tarefa que envolve Análise de Sentimentos. A anotação mais comum num *córpus* de sentimentos é quanto a polaridade (Positivo e Negativo) de suas sentenças. Mas também pode incluir, no caso de avaliação de produtos, quais os aspectos que estão sendo avaliados em cada sentença. Para a maioria das tarefas, estes *córpus* são necessários tanto para o treinamento de modelos de classificação que demandam exemplos – como é o caso de métodos clássicos de Aprendizado de Máquina, como *Support Vector Machines* (SVM) e classificadores Bayesianos – quanto para a avaliação dos métodos de classificação.

É possível encontrar vários *córpus* de sentimentos com anotação de polaridade para o inglês. Destacamos o *córpus* de *reviews* de produtos apresentado por Hu e Liu (2004); o *IMDB corpus*, com *reviews* de filmes (PANG; LEE, 2005); a anotação manual do *IMDB corpus* feita por Socher *et al.* (2013) e rebatizada de *Stanford Sentiment Treebank* (SST); e o *córpus* de sentimentos

do *Twitter* usado no evento SemEval (NAKOV *et al.*, 2016). Para o português brasileiro, temos o ReLi, contendo resenhas de livros (FREITAS *et al.*, 2012); os *corpus* Buscapé (HARTMANN *et al.*, 2014) e Mercado Livre (AVANÇO, 2015), do domínio de *reviews* de produtos; o *Corpus* 7X1 (MORAES; MANSSOUR; SILVEIRA, 2015) no domínio de comentários esportivos no *Twitter*; o Computer-BR (MORAES *et al.*, 2016), com comentários de produtos de informática; e o *corpus* apresentado por Silva *et al.* (2011) contendo comentários políticos.

Entre os métodos de AS mais utilizados estão os baseados em Aprendizado de Máquina (AM). Esses métodos usam diversas abordagens para analisar conjuntos de dados anotados (neste escopo, os *Corpus* de Sentimentos) e aprender os padrões que caracterizam cada classe da classificação de sentimentos.

Algumas abordagens de AM se destacam nos últimos anos por agrupar estruturas neurais multicamadas que conseguem identificar padrões muito mais abstratos do que os métodos convencionais. Esse movimento é conhecido como *Deep Learning* (Aprendizado Profundo) e vem obtendo bons resultados nas tarefas de análise de sentimentos e classificação de polaridade. Sua eficiência vem sendo estudada, porém sua dependência de grandes conjuntos de dados anotados é um empecilho para pesquisas em idiomas como o português brasileiro, para o qual poucas bases disponíveis atingem tamanho suficiente, cerca de dezenas ou centenas de milhares de documentos, que seria o recomendado para essa abordagem.

Socher *et al.* (2013) apresentaram um modelo neural aplicado ao *corpus* SST, obtendo acurácia de 85,4% na classificação binária (positivo e negativo) e 45,7% na classificação em cinco classes (adicionando 'Muito positivo', 'Muito negativo' e 'Neutro') em sentenças em inglês. Esse trabalho apresentou novos horizontes para a área, pois, além de introduzir o *corpus* SST, definiu valores de *baseline* para a tarefa de classificação.

Nos anos seguintes, diversos trabalhos usaram o *Corpus* SST como *benchmark* de Análise de Sentimentos para o inglês. Os resultados de Zhou *et al.* (2016) tornaram-se o estado da arte para o *Corpus* SST em classificação multiclasse, também chamada de *fine-grained*, obtendo 52,4% de acurácia. O estado-da-arte para a classificação binária também foi obtido pelos mesmos autores, com acurácia de 89,5%.

Ainda são poucos os trabalhos que se dedicam à classificação de polaridade para o português brasileiro. A literatura nos apresenta alguns trabalhos usando *reviews* de produtos (MORAES *et al.*, 2016; AVANÇO; BRUM; NUNES, 2016), comentários esportivos (MORAES; MANSSOUR; SILVEIRA, 2015), *reviews* de livros (FREITAS *et al.*, 2012; BRUM; ARAUJO; KEPLER, 2016) e política (TUMITAN; BECKER, 2013; TUMITAN; BECKER, 2014).

A variedade de *corpus* na língua portuguesa traz uma dificuldade na avaliação dos métodos, visto que poucos autores aplicam os mesmos métodos sobre outros *corpus*. Isso é causado pela indisponibilidade dos mesmos ou pela diferença do domínio, levando os pesquisadores a buscarem métodos mais apropriados para um domínio específico, como microblogs ou *reviews*

de produtos.

Poucas pesquisas para o português fazem uso ou apresentam resultados satisfatórios no uso de *Deep Learning* para a classificação de polaridade. Uma das causas é a ausência de um corpus de sentimento que seja adequado à técnica e, ao mesmo tempo, popular no sentido de ser aplicável a outras técnicas, permitindo uma comparação justa entre os sistemas. Ressalta-se que os corpus de sentimento disponíveis para o português ou apresentam anotações automatizadas, baseadas em estrelas e notas, que não necessariamente representam as polaridades reais dos textos, ou são de tamanho insuficiente para abordagens dependentes de muitos exemplos para aprendizado, como o *Deep Learning*.

Essa carência vai ao encontro das possibilidades oferecidas pela internet e seus ambientes virtuais. O microblog *Twitter* indica 313 milhões de usuários ativos por mês na rede social¹. As postagens no *Twitter* (denominadas *tweets*) são caracterizadas pela limitação de 280 caracteres (140 no início desse trabalho), pelo vocabulário característico e pela rapidez com que acontecimentos são difundidos no ambiente. É comum, por exemplo, um usuário validar uma informação (lançamento de seriados, morte de celebridades ou desastres) conferindo os *Trending Topics* (ferramenta da rede que mostra os assuntos mais comentados em tempo real). A rede social também é caracterizada por conter grande número de opiniões e avaliações entre suas postagens (NAAMAN; BOASE; LAI, 2010).

Na **Figura 1** podemos observar um recorte do *Twitter* exibindo uma busca pelos termos **Totalmente Demais**, que durante a data do acesso estava entre os *Trending Topics*. A busca foi limitada somente na região da cidade de São Paulo. Os valores abaixo dos tópicos dão uma ideia do grande fluxo de dados da rede social. Por exemplo, naquele momento, 169.000 *tweets* pediam o “cancelamento” do ano de 2016 (um desabafo cômico mostrando a insatisfação dos usuários com os eventos ocorridos no ano). É possível observar reações aos resultados da eleição presidencial dos Estados Unidos que havia acontecido no dia anterior (*#RIPAmerica*), assim como contratações de jogadores de futebol e lançamentos de álbuns musicais. O assunto **Totalmente Demais** foi escolhido empiricamente para ilustrar alguns fenômenos da rede social: podemos observar erros morfológicos e sintáticos, opiniões comparativas como em “Já podemos chamar Haja Coração de Totalmente Demais 2.0”, inferindo que a novela corrente poderia ser uma nova versão da anterior, e a insatisfação com o “FUNDO FAKE” e o “parque fake” citados em dois *tweets*.

Esses fenômenos se juntam à dificuldade de se determinar os alvos corretos dos *tweets* (a quem o usuário está se referindo), a ironia constante e a linguagem peculiar. Esse cenário apresenta um ambiente rico em fenômenos linguísticos, mas difícil de ser processado por métodos automáticos. Ainda na **Figura 1**, podemos somar 466.652 postagens somente em *Trending Topics* em um intervalo de 24 horas, o que nos dá uma ideia do fluxo de dados na rede

¹ Acessado em 16 de Janeiro de 2017. Valores datados de 30 de Junho de 2016. <<https://about.twitter.com/company>>

Figura 1 – Recorte de tela do *Twitter* mostrando uma busca por palavra-chave e os *Trending Topics*.

Fonte: <<https://twitter.com/search?q=%22Totalmente%20Demais%22&src=tren>>.

Acessado em 9 de Novembro de 2016.

social, e nos incentiva a desenvolver abordagens específicas para esse domínio. As dificuldades em anotação de corpúis limitam muito o estudo da rede social, pois os corpúis em português brasileiro são difíceis de serem encontrados e, em sua maioria, contêm poucos exemplos, o que pode gerar baixa representatividade dos fenômenos. Para o inglês, por exemplo, essa dificuldade de anotação também existe, porém o avanço nas pesquisas de análise semântica e eventos como o SemEval (NAKOV *et al.*, 2016) incentivam e disponibilizam mais dados para pesquisas e experimentos.

Algumas abordagens buscam diminuir o custo de anotação manual, como o uso de supervisão distante (GO; BHAYANI; HUANG, 2009; JUNIOR *et al.*, 2017). Essas abordagens incluem *features* observáveis que podem ser identificadas automaticamente para a classificação de sentenças. No Twitter, uma abordagem recorrente é o uso de *emojicons* para classificar documentos entre positivo e negativo. Essa abordagem foi proposta inicialmente por Go, Bhayani e Huang (2009), porém não traz garantias mais concretas de confiabilidade para o processo, além de limitar a análise ao excluir a classe neutra, como reconhecido pelos autores.

Parece-nos, então, que a criação de um grande corpúis de sentimento anotado, para uso na construção de modelos de classificadores de polaridade e para a comparação de métodos de classificação de polaridade em português, usando o *Twitter* como fonte de dados, deve contribuir para o avanço das pesquisas nessa área. Para contornar o alto custo do processo de anotação

humana, propomos o uso de métodos de Aprendizado Semissupervisionado (ZHU, 2005).

O Aprendizado Semissupervisionado caracteriza-se pelo uso tanto de dados anotados quanto de dados não anotados no treinamento de modelos (ZHU, 2005). A ideia é, de forma iterativa, usar os dados anotados para treinar modelos de classificação e aplicá-los aos dados não anotados, aumentando o conjunto de treino com os exemplos de maior confiabilidade na classificação.

Diversos trabalhos aplicaram abordagens não supervisionadas na tarefa de análise de sentimentos para o inglês (BAUGH, 2013; BECKER *et al.*, 2013; ZHAO; LAN; ZHU, 2014). Silva, Coletta e Hruschka (2016) fazem uma avaliação de métodos semissupervisionados na análise de sentimentos de textos do *Twitter* em inglês, obtendo bons resultados na classificação e discutindo a necessidade do estudo das características usadas na análise dos *tweets*.

1.2 Objetivos

Motivados pela dificuldade na formação de um *corpus* de sentimentos na língua portuguesa, nosso trabalho buscou reunir *tweets* anotados manualmente, e aliados a classificadores de aprendizado de máquina, para a formação de uma grande base de dados anotada em três classes (positiva, neutra e negativa).

Usamos o domínio de programas televisivos para a extração de *tweets* por conta do grande fluxo de dados movimentados por esses programas, pela facilidade na obtenção de dados com baixa ocorrência de ruídos (usando termos específicos de busca) e pelo caráter opinativo dos documentos recuperados com esse viés.

Abordagens semissupervisionadas foram utilizadas na classificação de novos documentos usando como treino o *corpus* anotado manualmente. Tais abordagens fazem uso de classificadores baseados em aprendizado de máquina aliados a arquiteturas iterativas que têm o objetivo de aumentar a eficiência dos métodos usados.

Estabelecemos três eixos de trabalho para o desenvolvimento da pesquisa: a **anotação de um *corpus* de sentimentos**; a **classificação de polaridade**; e o **aprendizado semissupervisionado**.

A principal questão de pesquisa que objetivamos responder pode ser formulada como: em que medida a expansão de um *corpus* anotado manualmente, sob uma abordagem semissupervisionada e iterativa, é capaz de gerar um grande *corpus* cujo uso produza resultados tão bons quanto aqueles gerados pelo *corpus* original na tarefa de classificação de polaridade? Em outras palavras, a anotação automática feita dessa forma preserva ou degrada a qualidade da anotação do *corpus* original na tarefa de classificação de polaridade?

1.3 Organização da monografia

No [Capítulo 2](#) apresentamos definições formais para Análise de Sentimentos, Mineração de Opiniões, assim como para os recursos utilizados, como *cópus*, léxicos de sentimentos e alguns métodos clássicos de classificação de polaridade. Também são apresentados modelos de aprendizado semissupervisionado, definições formais, os cuidados necessários no uso e o funcionamento das técnicas.

Os trabalhos relacionados a nossa pesquisa são apresentados no [Capítulo 3](#), com descrições de *cópus* de sentimentos para inglês e português, assim como métodos de estado-da-arte para ambos os idiomas. No final do capítulo fazemos um breve resumo sobre a área e apresentamos algumas lacunas da literatura para o português.

Apresentamos nosso *framework* de expansão de *cópus* no [Capítulo 4](#), primeiramente introduzindo o TweetSentBR, *cópus* de sentimentos anotado para a tarefa de classificação de polaridades, as etapas de extração dos dados e anotação, assim como algumas avaliações, características e estatísticas do *cópus*, em seguida apresentamos os métodos de classificação de polaridades utilizados pelo *framework* e a abordagem de aprendizado semissupervisionado empregada no projeto.

No [Capítulo 5](#) apresentamos os resultados obtidos no trabalho, os métodos de avaliação e a comparação dos nossos resultados, assim como discussões sobre os resultados obtidos.

Por fim, no [Capítulo 6](#) retomamos nossas questões de pesquisa e objetivos para encerrar o trabalho, concluindo a análise dos resultados e dissertando sobre sugestões e possibilidades para trabalhos futuros.

FUNDAMENTAÇÃO

Nossa pesquisa foi dividida em três eixos principais: a classificação de polaridade como tarefa de análise de sentimentos; a anotação de um corpus de sentimentos usando o *Twitter* como fonte de dados; e o uso de técnicas de aprendizado semissupervisionado para a expansão do corpus de treinamento usando dados não anotados. Neste capítulo trazemos definições formais das áreas, assim como desafios recorrentes e características de cada um dos eixos de trabalho.

2.1 Análise de Sentimentos

Os primeiros trabalhos envolvendo o estudo de avaliações humanas datam o início dos anos 2000. A primeira utilização do termo Análise de Sentimentos (AS) foi em [Nasukawa e Yi \(2003\)](#), descrevendo a tarefa como “busca de expressão de sentimento a um dado sujeito e a determinação da polaridade deste sentimento”. Outro termo semelhante, Mineração de Opinião (MO), foi apresentado em [Dave, Lawrence e Pennock \(2003\)](#) como “processamento de um conjunto de resultados de busca para um item gerando uma lista de atributos do produto e agregando opiniões sobre cada um deles”. Tanto AS quanto MO são atualmente empregados para definir a mesma área, porém as definições usadas nesses trabalhos se referem a tarefas específicas de AS e não formalizam o conceito.

A definição mais aceita de AS é como um “campo de estudo que analisa opiniões de pessoas, sentimentos, avaliações, atitudes e emoções a respeito de entidades, como produtos, serviços, organizações, indivíduos, acontecimentos, eventos, tópicos e seus atributos” ([LIU, 2012](#)). Ela abrange as vertentes da área e dá uma ideia das possibilidades que os estudos envolvendo sentimentos nos proporcionam.

Como subárea do Processamento de Linguagem Natural, a AS carrega suas características e seus desafios. As tarefas de PLN concentram-se em um ou mais dos seis níveis da língua ([JURAFSKY; MARTIN, 2000](#)):

- Fonética e Fonologia: conhecimento linguístico nos sons;
- Morfologia: conhecimento do significado de componentes das palavras;
- Sintaxe: conhecimento de estrutura relacional entre palavras;
- Semântica: conhecimento de significado;
- Pragmática: conhecimento sobre os objetivos e intenções do autor;
- Discurso: conhecimento sobre unidades linguísticas maiores que uma sentença isolada.

Tarefas de AS geralmente se concentram no nível da semântica, de onde se extrai o significado. Podemos dividir esse nível em duas categorias: a semântica lexical e a semântica composicional.

A semântica lexical ajuda a determinar o significado de uma palavra no contexto em que ocorre. Por exemplo, na sentença “O programa estava ok.”, o estrangeirismo *ok* qualifica positivamente programa. Assim, na tarefa de classificação de polaridade, a opinião expressa nessa sentença seria classificada como positiva.

Na semântica composicional buscamos analisar a ocorrência de composições lexicais, por exemplo na sentença “O espetáculo estava de cair o queixo.” temos o verbo *cair*, que semanticamente é geralmente usado para representar algo negativo, e *queixo*, substantivo completamente neutro na sentença, porém a expressão “cair o queixo” indica estupefação ou admiração, o que carrega polaridade positiva para a sentença.

A ambiguidade também é um desafio recorrente em tarefas de AS. A polaridade da sentença “O programa bombou”, por exemplo, poderia ser classificada como negativa ou positiva, dependendo de como “bombar” seja interpretado. Essas divergências, além de serem complexas para a análise automática, causam problemas na anotação manual dos *corpus* de sentimentos. A solução de uma ambiguidade semântica muitas vezes pode demandar conhecimento pragmático (qual o sentido que esse usuário específico pensa quando usa a palavra ‘bombou’) ou discursivos (essa sentença, no meio de outras, indiscutivelmente positivas, num *review*, pode determinar sua polaridade), porém o processamento desses níveis pode aumentar ainda mais a complexidade da tarefa.

A análise semântica também é muito dependente do domínio, visto que o significado das palavras e expressões podem variar dependendo das entidades e do ambiente onde o documento foi obtido (PAN *et al.*, 2010; BOLLEGALA; WEIR; CARROLL, 2013; AVANÇO, 2015).

Para ilustrar essa dependência de domínio, podemos pensar em dois domínios comuns na AS, que possuem características bem distintas: os *reviews* de produtos e os *reviews* de livros. Ambos lidam com opiniões e buscam avaliar um produto único, sendo um produto, por exemplo, um celular, ou um livro. A ocorrência dos adjetivos *interessante* e *profundo* podem ser facilmente

empregados para livros, porém, dificilmente seriam empregados para a avaliação de um celular. Da mesma maneira que atributos como *resistência* ou *durabilidade da bateria* são irrelevantes para livros, porém muito importantes na avaliação de celulares.

Além dos níveis de conhecimento e dos desafios na análise semântica, também é importante definirmos o nível de granularidade da tarefa proposta. Em geral, AS lida com três níveis de granularidade: nível de documento, nível de sentença e nível de entidade ou aspecto.

A análise de um texto completo – um *review* inteiro ou uma matéria jornalística – é considerada uma análise no **nível de documento** (TURNERY, 2002; PANG; LEE; VAITHYANATHAN, 2002). A opinião expressa em várias sentenças de um documento geralmente resume a opinião do autor sobre a entidade analisada. Ela pode incluir sentenças de polaridades variadas e o desafio é determinar qual é a polaridade completa que o documento representa. Na Figura 2 temos um *review* de celular¹ que representa opinião negativa (indicada pelo autor), apesar de também conter sentenças neutras e positivas em sua composição.

Figura 2 – *Review* de produto com polaridade negativa em nível de documento.

Fonte: Elaborada pelo autor.

A análise no **nível de sentença** busca extrair a polaridade somente de uma sentença. Essa análise seria um passo intermediário para a análise de documento, visto que a composição de sentenças pode indicar a opinião que prevalece no texto. Na Figura 3 podemos observar sentenças neutras, apresentadas na cor branca, em meio à avaliação majoritariamente negativa, cor vermelha. Neste nível nos deparamos com a definição de sentenças objetivas ou subjetivas, a classe neutra e a força de opiniões divergentes na mesma sentença (WILSON; WIEBE; HWA, 2004; SOCHER *et al.*, 2013).

O último nível da granularidade em AS é o **nível de entidade ou aspecto**. Nesse nível buscamos identificar atributos da entidade principal da opinião. Na Figura 4 podemos ver o *review* de um celular tratando de 4 atributos distintos: desempenho, resistência, sensores e a usabilidade do aparelho. Esse nível é geralmente usado em tarefas de sumarização de opiniões (HU; LIU, 2004).

¹ <<http://www.buscape.com.br/avaliacoes/smartphone-asus-zenfone-2-laser-ze550kl-16gb>>. Acessado em 26 de Janeiro de 2016.

Figura 3 – Review de produto analisado em nível de sentença.

Fonte: Elaborada pelo autor.

Figura 4 – Review de produto analisado em nível de entidade ou aspecto.

Fonte: Elaborada pelo autor.

Liu (2012) define formalmente uma opinião como um objeto de 5 atributos, $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, associado a um documento d .

O primeiro elemento, e_i , representa a entidade a qual a opinião se refere. O primeiro desafio é conseguir definir sempre o alvo da opinião, ou seja, o que o usuário está avaliando. A presença de um índice indica que podemos ter mais de uma entidade sendo avaliada em um documento.

O segundo elemento, a_{ij} , refere-se ao aspecto da entidade e_i que está sendo avaliado – o tamanho de um celular, por exemplo.

O elemento s_{ijkl} representa o sentimento, ou a classe da opinião. Essa classe pode variar dependendo da tarefa de AS. Por exemplo, pode ser a *polaridade* ou a *relevância* da opinião. Os índices k e l fazem referência aos próximos atributos da opinião.

Os dois últimos atributos referem-se ao autor da opinião (em inglês, *holder*) e ao tempo, respectivamente h_k e t_l . Segundo os autores, esses dados são indispensáveis para uma modelagem de opinião eficiente. A identificação de usuários influenciadores em redes sociais é um campo de pesquisa onde essas características são amplamente utilizadas (CHA *et al.*, 2010).

Para a modelagem da opinião, podem ser considerados os seguintes elementos: as entidades avaliadas (e), um ou mais aspectos das entidades (a), as classes de polaridade extraídas

das sentenças (s), a identificação de autores (h) e a variação de tempo (t) dos documentos. Nesse projeto investigamos AS em nível de documento para a tarefa de classificação de polaridade. Assim, se o documento expressa uma opinião, a análise de todo o texto definirá sua polaridade. Por ser em nível de documento, não são discriminados os aspectos, assim como também não tratamos de autoria e variação de tempo. Apesar do tamanho limitado de caracteres de um *tweet*, ele contém muitas vezes mais de uma sentença, portanto aqui ele é considerado um documento.

2.1.1 Classificação de polaridade

A classificação de polaridade é o problema mais clássico da AS. Nos primeiros trabalhos (TURNERY, 2002; PANG; LEE; VAITHYANATHAN, 2002), as opiniões eram classificadas em duas classes, positiva ou negativa.

Um usuário que avalia um produto ou serviço tende a possuir uma opinião binária sobre a entidade de avaliação. Turney (2002) fala sobre *recomendação*, porém a polaridade de um produto não representa necessariamente a recomendação do mesmo.

Essa divisão binária está presente em diversos trabalhos, pois se baseia no fato de que sempre existirá opinião em uma avaliação de usuário. Um viés dessa visão seria a existência de uma sentença neutra, por exemplo, “Achei esse filme normal, nem bom, nem ruim”. Nesse caso, não temos uma opinião relevante para a entidade alvo, apesar de existir uma avaliação do usuário.

A presença de uma classe neutra cria diversos problemas para a análise. Um deles é a dificuldade em definir uma sentença neutra. Dois conceitos podem contribuir para a análise de neutralidade em uma sentença: subjetividade e emoção.

Subjetividade sentencial é definida como a “presença de sentimento pessoal, visão ou crença” (LIU, 2012). O contrário da subjetividade é a objetividade, quando a sentença apresenta uma informação factual. Um exemplo é a sentença “O filme tem noventa minutos de duração”. Não existe positividade ou negatividade na sentença, pois não existe uma opinião pessoal associada à informação.

Liu (2012) define emoção como “sentimento e pensamento subjetivo”. Nesse conceito temos que sentenças opinativas possam ser consideradas avaliações racionais ou emocionais (CHAUDHURI, 2006). Um exemplo de avaliação emocional seria “Adorei a música nova do DJ”, onde o sentimento “adoração”, indicando positividade, é emocional quanto a música. A sentença “O carro tem ótima eficiência nas rodovias” fala de uma avaliação racional sobre a entidade, independente da opinião do usuário. Em nosso trabalho, não diferenciamos avaliações emocionais ou racionais na classificação, ou seja, mesmo que uma avaliação seja motivada pela razão, consideramos como opinativa. Justificamos essa escolha por considerar que a exposição de um ponto positivo ou negativo, mesmo que racional, expressa que o usuário aprecia ou não essa característica da entidade.

Liu (2012) também classifica opiniões em regulares ou comparativas. As primeiras podem ser divididas em diretas, quando avaliam a entidade diretamente, e indiretas, quando avaliam a entidade fazendo referência a um efeito causado por essa. Sentenças comparativas avaliam uma entidade usando uma comparação com outra entidade. Opiniões podem ser explícitas ou implícitas. Opiniões explícitas são afirmações ou proposições que avaliam uma entidade, seja diretamente ou por comparação. Já opiniões implícitas demandam conhecimento de mundo para serem classificadas corretamente. A Tabela 1 resume os tipos de opinião descritos e apresenta exemplos de cada fenômeno em uma sentença.

Tabela 1 – Tipos de opiniões em tarefas de AS (LIU, 2012).

Característica	Tipo de Opinião	Sub-tipo	Exemplo
Relação	Regular	Direta	<i>O filme é ótimo.</i>
		Indireta	<i>Depois do remédio, meu joelho ficou ainda pior.</i>
	Comparativa	-	<i>O filme é melhor que a série.</i>
Implicância	Explícita	-	<i>O filme é bom.</i>
	Implícita	-	<i>O filme me fez rir muito.</i>
Subjetividade	Objetiva	-	<i>O filme é mudo.</i>
	Subjetiva	-	<i>Adorei o filme.</i>
Emoção	Racional	-	<i>Retrata fielmente a vida do ator.</i>
	Emocional	-	<i>O humor do filme me agrada.</i>

Fonte: Elaborada pelo autor.

Além dos desafios relativos aos tipos de opinião, também a escolha do conjunto de classes oferece desafios. Diferentes configurações de classes são encontradas na literatura:

- **Duas classes:** Positiva e Negativa, também chamada de Classificação Binária (TURNERY, 2002; PANG; LEE; VAITHYANATHAN, 2002; AVANÇO; BRUM; NUNES, 2016);
- **Três classes:** Positiva, Neutra e Negativa (PANG; LEE, 2005; MORAES *et al.*, 2016);
- **Quatro classes:** Muito Positiva, Positiva, Negativa e Muito Negativa. Remoção da classe neutra e adição de superlativos nas classes polares (PANG; LEE, 2005). Essa proposta de classes não obteve tanta popularidade e poucos trabalhos a consideraram;
- **Cinco classes:** Muito Positiva, Positiva, Neutra, Negativa e Muito Negativa, também chamada de *fine-grained* (SOCHER *et al.*, 2013; TAI; SOCHER; MANNING, 2015). A estrutura de análise complexa exige uma anotação muito específica para essa configuração e seus melhores resultados vêm sendo obtidos por trabalhos envolvendo *Deep Learning*.

A adição de classes é um desafio a mais na AS. Além de aumentar a complexidade dos métodos de classificação, aumenta o custo da anotação de corpus de sentimentos, visto que a característica semântica da tarefa pode ser complexa até para seres humanos. Uma anotação binária pode causar menos discordância entre anotadores, agilizando o processo de anotação. Por outro lado, corpus anotados com cinco classes podem ser convertidos para quatro, três ou duas classes somente agrupando ou eliminando documentos das classes desejadas.

Apesar de comum nos trabalhos da área, dificilmente encontramos conjuntos de textos opinativos que não contenham avaliações neutras. Segundo [Go, Bhayani e Huang \(2009\)](#), “em aplicações reais, a classe neutra não pode ser ignorada”. A remoção da classe neutra normalmente é motivada por um aumento significativo na complexidade da tarefa, desde a definição até o desbalanceamento que ela pode causar em alguns domínios ([BRUM; ARAUJO; KEPLER, 2016](#)).

2.1.2 Abordagens e métodos

O número de trabalhos em AS cresceu muito na última década devido à popularização do uso de redes sociais e ao interesse em analisar o que nelas é expressado. Em particular, aos profissionais de mercado interessa saber o que os usuários pensam de seus produtos ou marcas. É sabido que opiniões alheias influenciam muito potenciais consumidores. No entanto, classificar opiniões é uma tarefa difícil mesmo para um humano. Diversas abordagens são propostas em AS para a tarefa de classificar a polaridade de uma sentença.

Podemos dividir as abordagens de classificação de polaridade em duas categorias: baseada em léxico e baseada em Aprendizado de Máquina. Para nosso propósito, em AM, trataremos separadamente as arquiteturas multicamadas (*Deep Learning*).

Classificadores Baseados em Léxico seguem a estratégia mais intuitiva na classificação de polaridade, em que se assume que a combinação das polaridade das palavras de uma sentença resulta em sua polaridade final. A partir da polaridade positiva de algumas palavras (como *gostar*, *amar*, *bonito*, *perfeito*), e negativa de outras (como *odiar*, *terrível*, *lixo*), as abordagens baseadas em léxico inferem a polaridade de sentenças e documentos.

O processo dos classificadores baseados em léxico geralmente parte da atribuição numérica às palavra que carregam polaridade. Para identificar as polaridades lexicais, pesquisadores criam ou recorrem ao uso de Léxicos de Sentimentos, que são conjuntos de palavras anotadas com sua polaridade. O primeiro trabalho que usou essa abordagem é o de [Turney \(2002\)](#) para *reviews* de produtos e serviços, mas também encontramos trabalhos mais recentes como o de [Taboada et al. \(2011\)](#), que aplicaram seu método em *reviews* de filmes e produtos, e [O'Connor et al. \(2010\)](#), que previram resultados de pesquisas de intenção de voto a candidatos à presidência norte-americana através do *Twitter*. No Português podemos citar [Avanço \(2015\)](#), que também propôs alguns classificadores de *reviews* usando abordagens baseadas em léxico, e [Tumitan e Becker \(2013\)](#), que propuseram uma abordagem para classificar sentimentos em *tweets* políticos.

A formação desses léxicos e o cálculo da polaridade lexical são essenciais para as abordagens baseadas em léxico. As classes morfológicas mais ricas em palavras com polaridade são os adjetivos e os advérbios, seguidos de verbos (particípios) e substantivos. Diversos trabalhos propuseram maneiras de se criar léxicos de sentimentos, quer seja manualmente, usando dicionários ou usando diretamente corpú de sentimentos (HATZIVASSILOGLOU; MCKEOWN, 1997; ESULI; SEBASTIANI, 2005; FREITAS, 2013).

Para o português brasileiro, temos à disposição cinco léxicos de sentimentos anotados com polaridades positiva e negativa. Eles são o SentiLex (SILVA; CARVALHO; SARMENTO, 2012), o Opinion Lexicon (SOUZA *et al.*, 2011), OntoPT-Sentic (OLIVEIRA; SANTOS; GOMES, 2014), o Léxico do ReLi (FREITAS, 2013) e o LIWC (BALAGE; PARDO; ALUISIO, 2013).

Apesar dos métodos baseados em léxico demandarem uma observação e um estudo linguístico maior, a vantagem desses métodos em relação aos baseados em AM é que eles não demandam longos processos de anotação de corpú, visto que podem ser aplicados sem qualquer treinamento. Essa economia aliada à redução de tempo de processamento, advinda da eliminação das etapas iterativas de treinamento, faz com que sejam popularmente usados como *baseline* de corpú recém compilados.

Um problema dos métodos baseados em léxico é sua alta dependência da anotação lexical. As mudanças de contexto podem fazer com que palavras alterem sua polaridade. Por exemplo, no Léxico do ReLi (FREITAS, 2013), encontramos o verbo “bobagem” como sendo negativo no domínio de *reviews* de livros. Isso faz sentido quando analisamos o enredo de uma trama, porém ao mudar para o domínio de produtos, o termo “bobagem” pode não apenas ser neutro, como pode descaracterizar uma crítica – como em “*Dizem que a tela dele é muito pequena, mas isso é bobagem*”. Tomá-lo como negativo nesse contexto pode ser prejudicial para a classificação.

Outra desvantagem é a dificuldade no tratamento da composição semântica das sentenças. A língua portuguesa, por exemplo, admite composições sintaticamente complexas que podem se perder em uma análise lexical simples. Composições maiores, comparativas, ou até mesmo com palavras incomuns podem afetar diretamente a classificação dos modelos baseados em léxico.

O ambiente analisado (sites de comércio, jornais, mensagens) também atrapalha um pouco esses métodos, visto que eles são amplamente dependentes dos léxicos em que se baseiam e, por isso, tendem a ter mais dificuldades em ambientes onde a linguagem é mais dinâmica, como redes sociais. É comum palavras caírem em desuso com o tempo, ou ganharem outro sentido, ou outra polaridade após algum acontecimento. Os casos de ironia e sarcasmo também atingem diretamente a eficiência desses classificadores.

Os Classificadores baseados em Aprendizado de Máquina dependem de um corpú anotado para aprenderem a classificar sentenças. Diversas técnicas clássicas são aplicadas à classificação de polaridade, entre elas destacamos: Máxima Entropia, que age como uma

regressão logística multinomial (BERGER; PIETRA; PIETRA, 1996); classificadores Naive Bayes, que classificam sentenças baseando-se em probabilidade condicional (MCCALLUM; NIGAM *et al.*, 1998); Árvores de Decisão, que classificam documentos por meio da combinação de características; e *Support Vector Machines* (SVM), que dividem um espaço usando um hiperplano de corte (CORTES; VAPNIK, 1995).

A maioria dos métodos de AM trabalha com representações numéricas. SVM, por exemplo, usa seus vetores de suporte para gerar um hiperplano que divide pontos em um plano. O desafio é escolher um modelo de representação de sentenças escritas em linguagem natural em vetores multidimensionais, que inclua seu significado.

A maioria dos autores usa representações vetoriais numéricas para representar as sentenças nas tarefas de AS. Esses vetores são formados a partir de *features*, ou características relacionadas à polaridade das sentenças.

Algumas dessas *features* são tradicionalmente empregadas em tarefas de AM, como *Bag-of-words*, que mede a ocorrência ou frequência de palavras em uma sentença. Cria-se um vetor com n posições onde n é o tamanho do vocabulário e representa-se cada sentença, ou documento, como um vetor com a ocorrência ou frequência de cada um dos n termos. Na modelagem do *Bag-of-words* é possível usar valores binários (1 para presença de um termo e 0 para ausência), o número de ocorrências, ou até mesmo *Term frequency – Inverse Document Frequency* (TF-IDF), entre outros. Naturalmente o tamanho do vetor, n , torna-se um problema computacional. Em tarefas de PLN, geralmente ignoram-se *stop-words*, palavras muito frequentes e de classes especiais, como artigos, preposições, pronomes, entre outras. A redução das palavra para sua forma canônica, num processo de *stemming*, também contribui para reduzir o tamanho do vetor de *features*.

Bag-of-words também pode ser empregado para termos compostos. Os *n-gramas* são agrupamentos de termos próximos que podem, ou não, carregar significado semântico. Por exemplo, em um modelo de bigramas (agrupando pares de termos adjacentes), analisando a sentença “Eu não gostei do filme”, teremos um vetor de *features* com os pares (*Eu, não*), (*não, gostei*), (*gostei, do*) e (*do, filme*). Nessa composição, o algoritmo de AM pode aprender, por exemplo, que a tupla (*não, gostei*) possui polaridade negativa na sentença. Não é necessário utilizar somente uma representação de *n-grama* (unigrama ou bigrama, por exemplo) na classificação. Usando *n-gramas* maiores ganhamos maior análise composicional, porém essa análise nem sempre se justifica. Janelas muito grandes demandam ainda mais exemplos para que os algoritmos aprendam os pesos corretos das composições, além de aumentar consideravelmente o tempo de processamento dos algoritmos.

Outra abordagem comum é agregar um léxico de sentimentos como *feature* do AM. Usando um léxico de sentimentos podemos considerar o números de palavras de polaridade positiva ou negativa como novas dimensões do vetor de *features*. O uso de etiquetas morfosintáticas (*PoS-tags*) também é geralmente usado, pois pode ajudar na desambiguação lexical

que pode ocorrer em sentenças opinativas. Na sentença “Eu gosto desse filme”, temos o verbo “gostar” conjugado no presente e indicando positividade quanto ao filme. Já em “Comi e não senti o gosto”, o substantivo “gosto” é neutro.

Liu (2012) ainda cita, como *features* possíveis, regras opinativas (*rules of opinion*) como composições idiomáticas ou padrões linguísticos que podem ser reconhecidos e programados manualmente, e modificadores de opinião (*opinion shifters*), como negação, incremento e decremento de polaridade e dependências linguísticas advindas de *parsers* ou árvores de dependência.

A maioria dos trabalhos do início dos anos 2000 trabalha com classificadores baseados em AM. Pang, Lee e Vaithyanathan (2002), por exemplo, avaliam três classificadores (Máxima Entropia, Naive Bayes e SVM) sobre um corpúsculo de *reviews* de filmes usando combinações de *features* com unigramas, bigramas, etiquetas morfossintáticas e adjetivos.

A facilidade na obtenção dos dados aqueceu muito a tarefa de classificação de polaridade, fazendo com que muitos trabalhos fossem publicados no período. Esses trabalhos variam muito em domínio, número de classes e corpúsculo utilizado, tornando difícil a comparação dos classificadores e respectivos resultados obtidos. Destacamos os domínios de *reviews* de filmes (PANG; LEE; VAITHYANATHAN, 2002; KÖNIG; BRILL, 2006) e *reviews* de produtos (DAVE; LAWRENCE; PENNOCK, 2003), bem como o uso de classificação de polaridade em três e quatro classes (GAMON, 2004).

Nos últimos anos, uma gama de técnicas de AM se sobressaiu das demais. O uso de arquiteturas que agregam camadas neurais (*Multi-layer perceptrons*) ganhou notoriedade no PLN a partir da publicação de Socher *et al.* (2013). Esse movimento foi chamado de **Deep Learning**, ou Aprendizado Profundo, e as técnicas são oficialmente parte do Aprendizado de Máquina. Neste trabalho decidimos apresentá-las separadamente, pois suas características diferem significativamente dos métodos mais tradicionais de classificação baseados em AM.

Deep Learning (DL) é o nome dado a “modelos computacionais compostos por múltiplas camadas de processamento que aprendem representações de dados em múltiplos níveis de abstração” (LECUN; BENGIO; HINTON, 2015). Neurônios combinam diferentes *features* usando pesos iniciados randomicamente para representar os dados de maneira que abstraíam os conceitos necessários para uma tarefa. A saída da arquitetura neural é, então, comparada à saída esperada contida no corpúsculo anotado, e um algoritmo reajusta os pesos (usando *backpropagation*) para reduzir o peso de combinações menos relevantes e aumentar o de combinações mais relevantes.

Socher *et al.* (2013) propuseram uma arquitetura recursiva, onde as sentenças são divididas em sintagmas (bigramas de palavras) e os pares são inseridos em uma Rede Neural Recursiva com Tensor (RNTN) que analisa em *bottom-up*, classificando cada segmento da sentença até atingir seu topo, como pode ser observado na Figura 5.

Os resultados foram superiores na tarefa de classificação de polaridade em comparação

Figura 5 – Exemplo da Rede Neural Recursiva com Tensor classificando um trecho quanto à polaridade.

Fonte: Socher *et al.* (2013).

com outros métodos de AM (Naive Bayes e SVM) usando um corpus proposto no mesmo trabalho, o *Stanford Sentiment Treebank*. A classificação das sentenças pode ser binária ou em cinco classes (*fine-grained*). Os melhores valores de acurácia para a classificação binária foram de 85,4%, enquanto que, para o *fine-grained*, a RNTN atingiu 45,7%, ambos superando classificadores Naive Bayes (binário 81,8%; *fine-grained* 41%) e SVM (binário 79,4%; *fine-grained* 40,7%).

A construção composicional de Socher *et al.* (2013) é muito interessante na AS, porém ela é muito dependente de ferramentas de *parsing* e podem adicionar um ruído a mais na classificação (caso a divisão automática retornada pelo *parser* esteja errada, isso afetará diretamente a classificação do modelo). A prática de anotação composicional acabou sendo pouco aproveitada nos trabalhos seguintes, possivelmente pelo aumento na complexidade (e consequentemente no custo) da anotação de *corpus*.

Outros trabalhos superaram o RNTN, entre eles, o Modelo de Espaço Vetorial (LE; MIKOLOV, 2014), Convolucional (KIM, 2014), arquiteturas de *Long short-term memory* (TAI; SOCHER; MANNING, 2015), redes de memória dinâmica (KUMAR *et al.*, 2016) e redes bidirecionais com neurônios LSTM (ZHOU *et al.*, 2016).

Os métodos baseados em DL têm uma dependência do tamanho do *córpus* utilizado. Em geral são necessárias centenas de milhares de exemplos, ou no mínimo algumas dezenas de milhares, para os resultados superarem classificadores como SVM. Essa limitação ainda prejudica trabalhos para o português, visto que existem poucos *córpus* de sentimentos que atingem esses tamanhos. [Brum, Araujo e Kepler \(2016\)](#) usaram o modelo RNTN ([SOCHER *et al.*, 2013](#)) e o Modelo de Espaço Vetorial ([LE; MIKOLOV, 2014](#)) em um *córpus* de domínio de *reviews* de livros, em português, porém os autores reportaram dificuldade na classificação por conta do tamanho e do balanceamento do *córpus* que não são os ideais para o uso de técnicas de DL.

Podemos citar também algumas combinações de abordagens, como as abordagens híbridas (AVANÇO, 2015), onde classificadores baseados em AM são combinados com classificadores baseados em léxico, e comitês de classificação (*ensembles*), onde são usados conjuntos de classificadores sobre os mesmos documentos, visando uma classificação mais eficiente (AVANÇO; BRUM; NUNES, 2016; SILVA, 2016). Os principais trabalhos da área serão discutidos detalhadamente no Capítulo 3.

2.1.3 Avaliações e Métricas

Supondo que temos um conjunto de documentos D e um conjunto de classes L , onde cada documento $D_i \in D$ possui uma classe associada $L_i \in L$. A tarefa de classificação consiste na busca de um classificador cl que retorne $cl(D_i) = L_i$.

A avaliação de modelos de classificação de polaridade é feita comparando-se a etiqueta L_i anotada no corpúsculo de sentimentos (*Gold*) com a saída obtida do classificador (*Predict*). A literatura de AS apresenta três medidas de avaliação, são elas a **F1-Score** (F_1), a **Acurácia** (AC) e a **Error-Rate** (ER).

A F_1 calcula a média harmônica entre Precisão (P) e Cobertura (C) medidas de um classificador dada uma classe. Os cálculos de Precisão e Cobertura baseiam-se em quatro valores, Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN), que podem ser observados na Tabela 2.

Tabela 2 – Matriz-confusão

		Classificador	
		Sim	Não
Córpus	Sim	VP	FN
	Não	FP	VN

Fonte: Elaborada pelo autor.

A Precisão mede o número de documentos classificados em uma classe que, de fato, pertencem a essa classe. Por exemplo, em um problema de classificação binária de polaridade, a Precisão da classe positiva retornaria a porcentagem de documentos classificados como positivos que foram corretamente classificados. A Equação 2.1 apresenta o cálculo da Precisão.

$$P = \frac{VP}{VP + FP} \quad (2.1)$$

A Cobertura, ou Revocação, mede a quantidade de documentos classificados em uma classe em relação ao número total de documentos desta classe. Seguindo o exemplo anterior, a Cobertura da classe positiva retorna a porcentagem de documentos corretamente classificados

como positivos em relação ao número total de documento positivos. A [Equação 2.2](#) apresenta o cálculo da Cobertura.

$$C = \frac{VP}{VP + FN} \quad (2.2)$$

A F_1 combina os resultados de P e C resultando num valor harmônico entre as duas medidas. Trabalhos da literatura geralmente usam a média das F_1 de todas as classes para comparar seus classificadores com os demais, porém a discriminação da F_1 de cada classe pode nos dar informações ainda mais granulares sobre a eficiência dos modelos. A F_1 é indicada, por exemplo, para avaliação de conjuntos com classes desbalanceadas. A [Equação 2.3](#) apresenta o cálculo da F_1 .

$$F_1 = 2 \cdot \frac{P \cdot C}{P + C} \quad (2.3)$$

Outras métricas muito usadas são a Acurácia (AC) e a *Error-Rate* (ER). Elas são complementares e medem as porcentagens de acerto e erro do classificador, respectivamente. A [Equação 2.4](#) apresenta a equação de AC , enquanto [Equação 2.5](#) apresenta o cálculo de ER , que é basicamente o complemento da AC .

$$AC = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.4)$$

$$ER = 1 - AC \quad (2.5)$$

Além das métricas anteriores, algumas práticas são comuns na avaliação dos modelos de classificação. Por existirem poucos corpú de sentimentos disponíveis, poucos autores dividem seus conjuntos para treinamento e avaliação. Tradicionalmente um único conjunto é usado para treinamento, validação e teste dos classificadores propostos ([SAMMUT; WEBB, 2011](#)).

Primeiramente divide-se o corpú anotado em três conjuntos (treino, validação e teste) formando três corpú menores que serão utilizados em três etapas distintas da avaliação dos modelos. Essa divisão é geralmente aleatória e respeita o balanceamento das classes anotadas. O conjunto de treino é normalmente composto de 80% do corpú completo. Os documentos contidos nele serão responsáveis pelo aprendizado do modelo de classificação. O conjunto de validação é responsável pelo treinamento de hiperparâmetros do classificador. Ele normalmente é composto de 10% do corpú completo, mas nem sempre é essencial no processo. Caso o classificador não forneça hiperparâmetros para serem ajustados, o corpú de validação torna-se irrelevante para a tarefa, adicionando mais 10% do corpú completo ao conjunto de treino. A questão do treinamento de hiperparâmetros ser realizada em um conjunto diferente do de treino reduz a chance de *overfitting*, ou seja, o treinamento excessivo que reduz a capacidade de generalização do modelo. Formado pelos 10% restantes do corpú completo, o conjunto de teste

é responsável por avaliar o modelo treinado seguindo as métricas apresentadas. É possível criar manualmente os conjuntos de teste, escolhendo os documentos mais representativos da tarefa para melhor compor o conjunto que irá avaliar os modelos.

Na AS esse estudo ainda carece de investigação, tanto pela ausência de grandes *córpus* de sentimentos, quanto pela dificuldade de se formalizar os fenômenos da tarefa. Por essa razão, os conjuntos são geralmente divididos aleatoriamente respeitando o balanceamento de classes e a proporção de divisão dos conjuntos.

Tradicionalmente são empregadas estratégias de *folds* durante o treinamento e avaliação dos métodos de AS. A abordagem de *fold* consiste na variação das combinações de treino, teste e validação para aumentar a confiança da eficiência do classificador em validações cruzadas. O *córpus* completo é dividido nos três conjuntos e o processo se dá normalmente. Ao término do teste, os valores são coletados e uma nova divisão é feita, garantindo novos documentos para os conjuntos de teste e validação. Após o término de todos os *folds*, é feita a média dos resultados de todos os *folds* para se obter o resultado do classificador segundo as métricas aplicadas.

2.2 *Córpus* de Sentimentos

Ambientes virtuais, *blogs*, livros e periódicos eletrônicos fazem com que muitos dados estejam disponíveis digitalmente hoje em dia. A área de PLN se beneficia desse material para enriquecer seus métodos e avaliar seus modelos em diversas tarefas.

Kilgarriff e Grefenstette (2003) definem *córpus* como “uma coleção de textos considerada um objeto de estudo linguístico”. Aplicado diretamente a PLN, Hovy e Lavid (2010) ainda dizem que a anotação é uma atividade que enriquece um *córpus*, pois adiciona informação linguística que pode ser usada para vários propósitos, como a investigação de teorias, a criação de dicionários ou de bancos lexicais, entre outros. Ambos os conceitos mostram a importância dos *córpus* na área da PLN, motivando ainda mais sua criação e anotação.

Usaremos os passos descritos em Hovy e Lavid (2010) para comentar alguns conceitos fundamentais da anotação de *córpus* e descrever algumas etapas mais ou menos importantes na classificação de polaridade. Os oito passos serão numerados e discutidos em sequência, respeitando a ordem recomendada pelos autores.

1. *Identificar e preparar uma seleção de textos representativos como material de início para o “córpus de treinamento”.*

O primeiro passo do processo é focado na origem dos dados a serem utilizados. Idealmente, a seleção dos dados deve proporcionar uma fonte de documentos que sejam relevantes para a abordagem e que dentre os documentos extraídos possamos encontrar todos os fenômenos linguísticos característicos da tarefa. Essa representatividade é muito difícil de ser atingida em

tarefas de PLN, especialmente nas que trabalham com os níveis mais avançados de conhecimento como a semântica e a pragmática, portanto é comum definirmos alguns limites para nossa análise, como o **domínio** e o **gênero** dos documentos extraídos.

Domínio caracteriza o assunto geral que buscamos analisar. As categorias de domínio podem variar em tarefas de PLN, visto a subjetividade de definir as nuances entre assuntos disponíveis. O *British National Corpus* (LEECH, 1992), um vasto acervo de documentos em inglês falado e escrito, usa domínio para descrever o assunto mais abrangente dos documentos do conjunto. Entre as categorias de domínio temos Artes, Crenças, Ciências Aplicadas e Ciências Sociais. Essa é uma possível categorização de domínios para um cópus vasto e genérico como o BNC. Trabalhos para o português também usam essa denominação, como o Lácio-Web (ALUÍSIO; OLIVEIRA; PINHEIRO, 2004), que é um conjunto de cópus em português brasileiro que reúne cópus anotados com diferentes etiquetas como morfologia, sintaxe, morfossintaxe, entre outras.

Na AS, o foco em opiniões e avaliações normalmente trata de domínio com uma granularidade maior do que a abrangência do BNC ou do Lácio-Web. Comumente o tipo do alvo da avaliação é considerado o domínio da aplicação, ou seja, o assunto geral onde podemos extrair informações semânticas que representam opiniões.

Algumas categorias de domínio na AS são os *reviews* de filmes (PANG; LEE; VAITHYANATHAN, 2002; SOCHER *et al.*, 2013), os *reviews* de produtos (HARTMANN *et al.*, 2014) e *reviews* de livros (FREITAS *et al.*, 2012). Outros domínios também usados são automóveis, bancos e destinos de viagens. (TURNERY, 2002; GAMON *et al.*, 2005).

Além do domínio, a definição do **gênero** dos textos trabalhados também pode influenciar diretamente nas tarefas de PLN. O gênero é o objetivo subjacente ao documento na sua criação. O conceito de gênero é muito subjetivo e pode ser interpretado por diferentes abordagens. Jurafsky e Martin (2000), por exemplo, citam textos jornalísticos, ficção, conversas telefônicas e páginas da internet, como sendo categorias de gêneros textuais. O autor ainda comenta a necessidade de buscarmos sempre tentar manter o mesmo gênero textual enquanto treinamos e avaliamos classificadores.

Assim como o domínio, o gênero pode influenciar muito nos fenômenos presentes nos documentos. O gênero jornalístico, por exemplo, tem o objetivo de informar leitores acerca de fatos e entidades de diferentes domínios, possuindo um comportamento geralmente mais sintaticamente correto. Já os CGUs podem variar na formalização sintática, o que os difere essencialmente de textos jornalísticos.

Além do domínio e do gênero, outra importante variável na extração de documentos é o ambiente onde estes se encontram. Nesse trabalho usaremos o termo **suporte linguístico** para definir características linguísticas dependentes do ambiente onde os documentos são obtidos. O suporte linguístico diferencia, por exemplo, CGUs de diferentes redes sociais. A Figura 6 ilustra

essa variação linguística em dois documentos de diferentes ambientes, sendo ambos pertencentes ao domínio de *reviews* de filme e do gênero opinativo.

Figura 6 – Documentos representando mesmo domínio (*review* de filme) e gênero opinativo.

Fonte: Elaborada pelo autor.

É muito comum em ambientes virtuais, como redes sociais e blogs, a extração de documentos retornar elementos que escapem do escopo do trabalho. Por essa razão é importante definir qual o domínio a ser trabalhado e o gênero buscado para que possam ser tomadas medidas que reduzam a ocorrência de documentos ruidosos no *corpus*.

A definição de fenômenos também é importante para a representatividade do *corpus*, porém ainda é muito difícil de se estabelecer padrões em tarefas de cunho semântico, como a AS. Liu (2012) propõe alguns tipos de opinião, como visto na Tabela 1 apresentada na Subseção 2.1.1. Garantir que todos os fenômenos estejam presentes no *corpus* gerado contribui para a melhoria de modelos treinados no *corpus*.

2. Instanciar dada teoria linguística ou conceito linguístico, especificando o conjunto de etiquetas a serem usadas, suas condições de aplicabilidade, etc. Este passo inclui o início da escrita do manual de anotação (geralmente chamado de Codebook ou Manual).

Além das escolhas referentes à origem do *corpus*, a **instanciação da teoria** busca indicar como a anotação da tarefa será realizada. Considerando o nível de conhecimento trabalhado, essa tarefa pode ser mais simples ou mais complexa, visto que a subjetividade de tarefas de PLN, como análise discursiva, similaridade semântica e a própria AS podem apresentar dificuldades em que até mesmo seres humanos podem entrar em desacordo com facilidade.

Uma boa teoria linguística pode reduzir o índice de discordância entre anotadores, o que economiza muito tempo na anotação de um *corpus*. Para a classificação de polaridade, definir os quesitos que denotam positividade, neutralidade e negatividade é essencial para o início da anotação.

Os primeiros trabalhos da área usavam a anotação discreta realizada por usuários, como notas ou estrelas (TURNERY, 2002; PANG; LEE; VAITHYANATHAN, 2002), porém alguns ambientes não apresentam essa opção, como as redes sociais e os *blogs*. Avanço (2015), por exemplo, usa o *corpus* Buscapé (HARTMANN *et al.*, 2014) para análise de sentimentos. O *corpus* Buscapé é composto de *reviews* e de notas entre 0 e 5, fornecidas pelo usuário. Os autores analisaram uma amostra do *corpus* e definiram que *reviews* seriam considerados negativos se

possuíssem notas 1 e 2, enquanto os positivos seriam os que possuíssem notas 5. Essa definição ilustra os cuidados necessários com o uso de valores discretos feitos por usuários. Os autores admitiram que avaliações com nota 0, que deveriam representar os usuários mais insatisfeitos, não correspondiam de fato a textos negativos. *Reviews* com as notas 3 e 4 foram, inclusive, descartados, talvez pela heterogeneidade de opiniões ocorrentes.

Alguns fenômenos devem ser observados durante a composição de um manual de anotação. Por exemplo, se opiniões objetivas serão classificadas como neutras ou polares. Uma opinião objetiva, “O programa está atrasado”, não constitui uma avaliação por parte do usuário, mas sim uma constatação de um fato ocorrente. Na falta de uma definição oficial de anotação, anotadores diferentes podem se basear em suas experiências para anotar uma sentença objetiva.

O manual deve conter diretrizes do escopo da análise, até quando os anotadores podem considerar o contexto do fenômeno descrito. Por exemplo, em [Moraes, Manssour e Silveira \(2015\)](#), que apresenta a anotação de um cópus de sentimentos em um período de tempo, os autores mostram um exemplo de sentença que no início do período analisado tem conotação positiva e, a medida que acontecimentos externos acontecem, a mesma sentença é usada para indicar sentimento negativo.

Outro ponto importante é o **desenvolvimento de uma interface de anotação**. Essa interface, geralmente construída para operar na *web*, deve apresentar as sentenças ou documentos a serem anotados e as possibilidades de anotação.

A literatura nos traz alguns exemplos de interfaces de anotação distribuída, como o *AGTK-toolkit* ([MAEDA; STRASSEL, 2004](#)) e o *GATE TeamWare* ([BONTCHEVA et al., 2010](#)). Outros autores preferem produzir seus próprios ambientes de anotação ([HARTMANN et al., 2014](#); [MORAES et al., 2016](#)). Algumas ferramentas de trabalho colaborativo (*crowdsourcing*) também são usadas para anotação de cópus. O *Amazon Mechanical Turk* ², por exemplo, é uma rede de trabalho colaborativo onde usuários podem requisitar serviços distribuídos a outros usuários e já foi usado por trabalhos de anotação de polaridade ([SOCHER et al., 2013](#)). Além dele, o *CrowdFlower* ³ também é outra alternativa usada por trabalhos da área ([NAKOV et al., 2016](#)).

Um produto direto dessa etapa é a formação de um manual de anotação. O manual de anotação deve conter entradas com diretrizes para a anotação do cópus e deve ser apresentado aos anotadores antes da anotação.

3. Anotar algum trecho do cópus de treino, a fim de determinar a viabilidade tanto da instanciação da teoria quanto do manual de anotação.

A anotação de um trecho do cópus pode prevenir problemas de instanciação de teoria

² <https://www.mturk.com>. Acessado em 24 de Fevereiro de 2017.

³ <https://www.crowdfunder.com/>. Acessado em 24 de Fevereiro de 2017.

que só apareceriam em avançadas etapas do processo de anotação. A partir desse passo simples, podemos medir concordância, perguntar aos anotadores os casos mais frequentes para adicionar ao manual de anotação e remover alguma indicação que gere ambiguidade na interpretação dos anotadores.

Outra observação importante que pode ser feita nessa etapa é o tempo da anotação, visto que ainda existe uma parcela grande do corpus a ser anotada nesse estágio, podemos inferir o tempo médio de anotação de um documento, ou o número de documentos anotados em uma faixa de tempo para gerar uma estimativa do término da anotação.

Uma questão que pode surgir nessa etapa do processo é a **seleção e o treinamento dos anotadores**. Apesar do manual contendo as diretrizes de anotação, identificar quais anotadores devem ser usados no processo pode ser importante para que a tarefa seja bem executada. Algumas tarefas, como análise sintática ou morfologia, podem demandar que o anotador seja um especialista no assunto.

Kilgarriff (1999), por exemplo, cita como uma questão crítica a escolha de indivíduos qualificados para a tarefa de anotação de desambiguação lexical. Na AS, os trabalhos de classificação de polaridade não especificam muito essa questão. Alguns autores relatam o uso de *crowdsourcing* (SOCHER *et al.*, 2013; NAKOV *et al.*, 2016) para a anotação de polaridade. Nakov *et al.* (2016) ressaltam que anotadores confiáveis são necessários para a tarefa, mas não cita a necessidade em especialistas da área na anotação de polaridade de *tweets*.

4. Medir os resultados (comparando as decisões dos anotadores) e decidir quais medidas são apropriadas e como elas devem ser aplicadas.

Uma métrica de concordância para anotação pode nos trazer importantes informações sobre o manual e sua legibilidade. Um valor alto de concordância pode representar que os anotadores se encontram alinhados na tarefa de anotação, assim como as instruções causam poucas dúvidas ou a tarefa é objetiva o suficiente para não causar divergências severas entre anotadores.

A área de SA nos apresenta diversas medidas de concordância, o que aumenta a dificuldade na comparação das diferentes anotações. Geralmente as métricas retratam um valor entre 0 e 1 indicando a uniformidade na anotação, porém os valores ideais para cada tarefa são ainda hoje motivo de estudo no PLN, especialmente para tarefas que envolvem alto grau de subjetividade.

Alguns trabalhos não apresentam a métrica calculada e outros ainda apresentam um valor sem identificar qual método foi utilizado para o seu cálculo. Essas ocorrências dificultam ainda mais a padronização de uma escala absoluta de garantia de concordância.

A medida mais clássica é o coeficiente *kappa* (κ). O *kappa* mede a concordância entre

anotadores usando a seguinte equação:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \quad (2.6)$$

$\bar{P}$ é a concordância direta entre anotadores, enquanto $\bar{P}_e$ é a probabilidade hipotética de chance de concordância. Existem duas variações do coeficiente κ que podem ser utilizadas - o Cohen's Kappa (COHEN, 1960) e o Fleiss' Kappa (FLEISS; COHEN, 1973). Ambos usam a mesma equação para calcular a concordância, mas variam no cálculo de $\bar{P}$ e $\bar{P}_e$, sendo o Cohen's Kappa limitado somente a pares de anotadores, enquanto o Fleiss' consegue flexibilidade no número de anotadores.

Outra métrica usada é o Krippendorff's Alpha (α), descrito em Krippendorff (2004). Essa métrica tem como vantagem a flexibilidade da variação do número de anotadores, etiquetas e da possibilidade de existirem documentos não anotados por um ou outro anotador.

A equação do Krippendorff's Alpha é descrita por:

$$\alpha = 1 - \frac{D_o}{D_e} \quad (2.7)$$

D_o é a discordância entre anotadores, enquanto D_e é a chance de discordância esperada.

Na AS não existe um padrão único para avaliação de concordância, alguns trabalhos fazem uso do Kappa (HARTMANN *et al.*, 2014; DOSCIATTI *et al.*, 2015), enquanto outros usam o Alpha (BERMINGHAM; SMEATON, 2011; ARRUDA; ROMAN; MONTEIRO, 2015),

5. Determinar qual nível de concordância é considerado satisfatório (pouca concordância significa pouca consistência na anotação para permitir que algoritmos de AM sejam treinados com sucesso). Se a concordância não é satisfatória, o processo pode se reiniciar a partir da segunda etapa, com mudanças apropriadas na teoria, sua instanciização, no Manual e nas instruções de anotação. Caso contrário, o processo segue para a próxima etapa.

Após a escolha da métrica de concordância e tendo sido feita a anotação de um trecho do cópus (etapa 3), podemos definir um *threshold* de concordância que será considerado satisfatório para a anotação.

A concordância pode variar para o tipo da tarefa de PLN, número de classes utilizadas e para cada especificidade da anotação. Anotações de cópus semelhantes podem atingir valores diversos na concordância, Freitas *et al.* (2012), por exemplo, relatam 79.9% de concordância na anotação das opiniões do cópus ReLi, enquanto (MORAES *et al.*, 2016) relata concordância de 53% entre anotadores na classificação de polaridade em *tweets*.

Como descrito no enunciado da etapa, quanto menor a concordância entre anotadores, mais duvidosa será a anotação, o que pode acarretar em problemas no aprendizado de algoritmo

de AM. Na AS, entretanto, é normal que exista discordância entre anotadores, especialmente quando a classe neutra é presente na classificação. Isso decorre da característica semântica da tarefa, que compreende uma visão muito mais subjetiva da polaridade preterida pelo usuário que externou a opinião.

6. Anotar uma grande porção do corpus, possivelmente durante vários meses ou anos, com muitas checagens intermediárias, melhoramentos, etc.

Com a concordância definida e as modificações no manual realizadas, os anotadores podem estender a anotação para todo o corpus extraído. O término da etapa culmina na entrega de um corpus anotado.

Uma questão importante sobre o corpus é a sua **disponibilização**. Corpus anotados em PLN costumam ser recursos raros e normalmente são utilizados para mais de uma tarefa. O corpus ReLi (FREITAS *et al.*, 2012), por exemplo, é usado em Brum, Araujo e Kepler (2016) para classificação de polaridade em três classes, já em Balage, Pardo e Aluisio (2013) podemos ver o mesmo corpus sendo usado para classificação binária de polaridades. O corpus Buscape (HARTMANN *et al.*, 2014) apresentou ainda mais variação na utilidade como recurso, Bertaglia e Nunes (2016) usam o corpus para normalização textual de CGUs, Almeida *et al.* (2016) usaram o mesmo corpus para avaliação de um corretor gramatical fonético, enquanto Anchiêta *et al.* (2015) e Avanço, Brum e Nunes (2016) usam para classificação de polaridade.

Para que seja utilizado como recurso, o corpus deve ser disponibilizado observando-se as políticas de distribuição providas pela origem dos dados, construindo um *site* de acesso e usando formatos comuns na PLN, como XML. Essas definições são importantes para que um corpus continue sendo utilizado e incentive novas pesquisas na área.

7. Quando material suficiente for anotado, treinar um método de PLN baseado em AM na porção de treinamento e medir sua performance no restante (ex. comparando resultados obtidos pelos classificadores em um trecho, geralmente chamado de “held-out”, com as anotações manuais).

Para essa etapa, classificadores podem ser empregados no treinamento de modelos de classificação, usando o corpus gerado. Os autores propõem a utilização de um *held-out* para a validação dos dados, porém a literatura de AS tradicionalmente realiza *folds*, como explicado na Subseção 2.1.3, que trata das avaliações e métodos da classificação de polaridade. Da mesma maneira, as métricas utilizadas podem ser as mesmas da literatura, *F-Measure* e Acurácia.

8. Se a concordância é satisfatória, os modelos podem ser aplicados para dados adicionais, não anotados, materiais semelhantes, auxiliando assim futuras análises. Se a concordância não é satisfatória, o processo se repete, possivelmente a partir da segunda etapa, ou da sexta se mais dados de treinamento forem necessários.

A última etapa proposta é utilizar o modelo treinado no *corpus* anotado para anotar novos documentos. Essa abordagem afirma que um classificador ensinado a analisar uma tarefa de PLN conseguirá reproduzi-la tão bem quanto um ser humano.

Para tal é necessário que um classificador tenha uma boa capacidade de generalização, ou seja, consiga aprender padrões genéricos usando poucos documentos e que esses padrões sejam eficientes para classificar dados não conhecidos.

2.3 Aprendizado Semissupervisionado

Uma limitação dos métodos de AM supervisionado é sua dependência de dados anotados para treinamento de modelos. Na PLN, essa questão ainda é mais crítica pela diversidade de possíveis domínios a se explorar, dos diferentes idiomas existentes e da variedade de tarefas da área.

Na [Seção 2.2](#) citamos que esse processo de anotação pode ser muito custoso, especialmente quando lidamos com uma tarefa subjetiva como a AS. Além do custo de anotação, que demanda recursos humanos e tempo, ainda é necessário o tempo de estruturação do processo, a formação da teoria linguística, manutenção do manual e desenvolvimento de interfaces próprias para a tarefa.

Esse custo faz com que alguns domínios mais específicos, ou idiomas menos populares sofram com escassez de recursos para treinamento de modelos em algumas tarefas.

Abordagens de aprendizado semissupervisionado “fazem uso de uma grande quantidade de dados não anotados em conjunto com dados anotados para construir melhores classificadores” ([ZHU, 2005](#)). Essas abordagens geralmente combinam uma parcela de dados anotados para o treinamento de modelos, enquanto enriquecem os mesmos usando informações incertas (dados não anotados).

Segundo [Zhu \(2005\)](#), abordagens semissupervisionadas utilizam dados não anotados para modificar ou priorizar probabilidades já obtidas por meio de dados anotados.

Para ilustrar o processo de semissupervisão, consideramos a [Figura 7](#). A figura apresenta quatro quadros representando um espaço 2D contendo documentos positivos e negativos, indicados pelos sinais de “mais” e “menos”, enquanto documentos não anotados são representados por círculos.

No quadro superior esquerdo podemos observar o espaço contendo dados anotados (4 positivos e 5 negativos) e um corte horizontal produzido por um classificador treinado com esses dados que divide o espaço em dois subespaços distintos. No quadro ao lado podemos observar os dados não anotados dispostos sobre o mesmo espaço.

No quadro inferior esquerdo apresentamos o corte horizontal classificando esses dados não anotados, podemos perceber que uma parcela desses dados se encontra na região de incerteza

Figura 7 – Exemplo de aprendizado semissupervisionado em um espaço bi-dimensional.

Fonte: Elaborada pelo autor.

do classificador (mais próxima do corte). Esses dados incertos são descartados enquanto os demais são adicionados aos dados já anotados (apresentados no quadro superior esquerdo) para a geração de um novo corte no espaço. No quadro inferior direito podemos observar esse novo corte dividindo os subespaços, assim como os dados anotados (em negrito) e os dados classificados com confiança na etapa anterior.

Essas abordagens são usadas geralmente em processos onde a anotação é muito custosa, como tarefas que façam uso de anotação no nível de palavra (QI *et al.*, 2009), classificação de documentos para domínios diversos (DASGUPTA; NG, 2009) ou em classificação multi-idíomas (RAINA *et al.*, 2007; WAN, 2009).

Dentre as abordagens mais clássicas de semissupervisão, temos o uso de modelos gerativos, como *Expectation-Maximization* (DEMPSTER; LAIRD; RUBIN, 1977), onde a distribuição probabilística gerada pelo modelo pode ser enriquecida com o uso de dados não anotados; as abordagens iterativas, onde dados não anotados são adicionados gradativamente ao treinamento com base na confiança da classificação (TANHA; SOMEREN; AFSARMANESH, 2017); e abordagens baseadas em grafos, que representam documentos como vértices e medidas de similaridade como arestas.

2.3.1 Self-training

O *self-training* é um modelo iterativo de aprendizado semissupervisionado onde um classificador baseado em AM age como um anotador, classificando dados e aprendendo com suas próprias anotações.

A abordagem parte de um classificador treinado com um conjunto de dados anotados ma-

nualmente para ser aplicado a um conjunto de dados sem anotação. Em seguida, os documentos que possuem maior confiança nessa classificação são inseridos no conjunto de dados anotados, enquanto é feito um novo treinamento sobre este novo conjunto. [Zhu e Goldberg \(2009\)](#) definem *self-training* como um modelo de aprendizado que usa suas próprias previsões para ensinar a si mesmo.

Apesar de simples, diversos autores relataram bons resultados obtidos com *self-training* na literatura em PLN, como na tarefa de desambiguação lexical ([YAROWSKY, 1995](#)) e no melhoramentos de classificação em *cross-domain* nas tarefas de *parsing* ([MCCALLUM; NIGAM et al., 1998](#)) e anotação de papéis semânticos ([SAGAE, 2010](#)).

O funcionamento da abordagem se baseia na adição de novos elementos com características ainda não analisadas pelo classificador na base de treinamento, para exemplificar citaremos um exemplo a seguir.

Suponhamos que um classificador de polaridades binário (positivo e negativo) aprenda que a incidência do verbo “amei” indique forte positividade, enquanto nenhum documento do treinamento indique que “gostei” funcione de maneira semelhante. Ao classificar a sentença “Amei aquele filme e adorei aquele ator”, o modelo atesta uma probabilidade alta desse documento pertencer à classe positiva, portanto ele é adicionado ao conjunto de treinamento. Sendo retreinado agora, o modelo se modifica e aprende que o termo “gostei” também possui uma pequena chance de indicar positividade, o que favorece a classificação de novas sentenças.

O método de *self-training* funciona iterativamente e pode ser implementado de diversas maneiras, porém certos cuidados devem ser tomados a fim de que o método funcione corretamente.

O primeiro cuidado é a definição de uma regra de adição ao corpus de treinamento. Dependendo da tarefa na qual o *self-training* é utilizado, será necessário observarmos a eficiência dos próprios classificadores utilizados, isto é, um classificador que obtém 70% de acurácia em uma tarefa acertará a classificação de sete a cada dez documentos que classificar.

Um valor muito baixo de *threshold* pode fazer com que documentos classificados equivocadamente sejam adicionados ao conjunto de treino, propagando o erro ao longo da execução. Isso se agrava ainda mais se na primeira iteração do algoritmo essa situação for muito incidente.

Já o oposto, a definição de um *threshold* muito alto pode aumentar consideravelmente o número de iterações e até mesmo enviesar a classificação para classes onde os classificadores possuam mais facilidade em analisar. Em nosso caso, a classe neutra é consideravelmente mais complexa de ser definida (tanto linguisticamente quanto pelos classificadores baseados em AM) do que a positiva e a negativa, isso pode fazer com que em cada iteração existam menos exemplos da classe sendo adicionados ao conjunto de treino.

O próprio modelo de classificação também tem um papel muito importante no *self-training* ([ZHU; GOLDBERG, 2009](#)). Um modelo de difícil generalização ou um treinamento

que faça *overfitting* pode influenciar negativamente na adição de novos documentos, visto que os dados adicionados serão sempre muito semelhantes aos já anotados, diminuindo a variação de novos fenômenos no conjunto de treino.

A utilização dessa abordagem é muito comum na tarefa de classificação de polaridades entre competidores do evento SemEval (BAUGH, 2013; BECKER *et al.*, 2013; ZHAO; LAN; ZHU, 2014). Além disso ela já foi empregada com bons resultados na tarefa em inglês (XIANG; ZHOU, 2014; SILVA *et al.*, 2016).

2.3.2 Co-training

Uma desvantagem do *self-training* é que a adição de novos documentos é completamente dependente do classificador usado. Intuitivamente, a utilização de mais de um classificador poderia oferecer maior confiança para que um documento seja adicionado ou não ao conjunto de treinamento.

O *co-training* é uma abordagem iterativa semelhante ao *self-training*, mas que utiliza mais de um modelo de classificação durante o processo de aprendizado semissupervisionado (BLUM; MITCHELL, 1998).

O treinamento de um classificador baseado em AM pode ser visto como a minimização de uma função de custo que divide um espaço. Dependendo da observação de tal espaço, um método de *self-training* consegue aprender com mais confiança um conjunto de documentos. No *co-training*, dois espaços ou mais são formados usando diferentes representações para o mesmo conjunto de dados, dessa maneira, em cada iteração serão formados dois ou mais conjuntos de dados mais confiáveis. A ideia do *co-training* é que essa visualização espacial múltipla consiga trazer para modelos distintos dados classificados com confiança pelos demais modelos.

A Figura 8 ilustra o exemplo acima. Nela, temos no lado esquerdo um espaço bidimensional com dados sendo classificados em duas classes, sendo que os dados mais distantes do corte são considerados mais confiáveis. Ao lado vemos outro classificador analisando os mesmos dados, porém usando uma representação espacial que os dispõe de maneira diferente no espaço. Observando onde os dados (facilmente separáveis no espaço da esquerda) são dispostos nesse espaço, podemos supor que muitos dos pontos classificados com confiança pelo primeiro classificador seriam descartados pelo segundo. Da mesma forma, a recíproca também é válida.

Segundo Zhu (2005), são necessárias três condições para o uso de *co-training*: 1) deve ser possível dividir em dois grupos as *features* usadas na modelagem dos documentos; 2) cada subdivisão de *feature* deve ser suficiente para treinar um classificador; 3) os subconjuntos são condicionalmente independentes para cada classe.

Diversos trabalhos da AS usam *co-training* em suas tarefas. Wan (2009), por exemplo, faz classificação binária de polaridades para *reviews* em chinês. Yu (2014) investigou o uso de *co-training* na AS em domínio de notícias, *reviews* e *blogs*.

Figura 8 – Independência condicional na divisão de *features* do *co-training*

Fonte: [Zhu \(2005\)](#).

Ambas as abordagens iterativas são muito utilizadas na literatura, porém outras técnicas podem ser agregadas a fim de facilitar o processo de classificação e para garantir mais confiança no aprendizado. Uma delas é o *Active Learning* ([COHN; ATLAS; LADNER, 1994](#)).

No *Active Learning*, a abordagem semissupervisionada iterativa obtém dois conjuntos de documentos em cada iteração, um com confiança para ser adicionado no treinamento e um com maior incerteza para ser anotado manualmente. O esforço manual necessário aqui é direcionado somente para os documentos onde o classificador obteve mais dificuldade na classificação; isso reduz o esforço humano na anotação e aumenta a eficiência do classificador nas próximas iterações ([DASGUPTA; NG, 2009](#)).

Outra técnica que pode ser aplicada em ambos os casos é a inserção de um passo de avaliação para o classificador em cada iteração, avaliando se o modelo ganha ou perde eficiência após a adição de novos documentos. Esse passo pode facilitar na identificação de documentos que tenham sido classificados erroneamente antes que esse erro se propague no aprendizado.

Em nosso trabalho implementamos um modelo de aprendizado semissupervisionado usando *self-training*. Apresentamos mais características da nossa implementação, assim como dos classificadores e representações utilizadas, no [Capítulo 4](#).

TRABALHOS RELACIONADOS

Neste capítulo apresentamos os trabalhos que melhor se relacionam com algumas das tarefas essenciais para a nossa proposta. Dividimos o capítulo em duas tarefas principais, primeiramente, apresentando alguns *corpus* de sentimentos presentes na literatura para inglês e português brasileiro, assim como suas referentes anotações, e, em seguida, alguns métodos de classificação de polaridade utilizados para a língua inglesa e a portuguesa.

3.1 *Corpus* de Sentimentos

Nesta seção apresentamos alguns *corpus* de sentimentos encontrados na literatura e discutimos algumas técnicas de anotação. É importante ressaltar que é comum encontrarmos trabalhos onde os autores apresentam um método inovador para uma tarefa e propõem um *corpus* para sua avaliação, e isso resulta em *corpus* que nem sempre são comparáveis. Nesta seção tratamos somente da construção dos *corpus*, mas não das técnicas de classificação, exceto se estas forem importantes para a anotação.

3.1.1 *Corpus* em Inglês

Turney (2002) fez o primeiro trabalho sobre Análise de Sentimentos da literatura. O autor buscava propor um algoritmo de aprendizado não supervisionado que classificasse *reviews* em “recomendado” e “não recomendado”.

O *corpus* formado era composto de 410 *reviews* de produtos e serviços obtidos no site *Epinions*, que reúne *reviews* de carros (*Honda Acoord* e *Volkswagen Jetta*), bancos (*Bank of America* e *Washington Mutual*), filmes (*Matrix* e *Pearl Harbor*) e destinos de viagens (*Cancun* e *Puerto Vallarta*). A anotação usada no *corpus* é a nota do usuário fornecida no *review*. As notas são medidas com estrelas, sendo que o autor considera *reviews* com até 3 estrelas como não recomendados, e acima disso como recomendados.

Pang, Lee e Vaithyanathan (2002) decidiram focar no domínio de *reviews* de filmes. Os autores citam que uma das motivações foi o trabalho apresentado anteriormente, que identificou os *reviews* de filmes como sendo o domínio de maior complexidade na análise. O trabalho reuniu *reviews* do IMDB ¹, sendo que os autores usaram uma escala numérica para avaliar os *reviews* (estrelas ou notas). Após a coleta, os documentos eram convertidos para uma das três classes, positiva, negativa ou neutra. Apesar da conversão em três classes, o trabalho manteve o foco na classificação binária, ignorando a classe neutra do conjunto usado nos experimentos. Uma das preocupações dos autores foi o grande número de *reviews* de um mesmo usuário, o que enviesaria o corpus em um padrão linguístico individual, ou de um pequeno grupo de usuários muito atuantes na rede. Para contornar isso, os autores estabeleceram um limite máximo de 20 *reviews* de cada usuário para cada polaridade. Ao todos eles coletaram *reviews* de 144 usuários, obtendo 752 negativos e 1301 positivos.

Em Pang e Lee (2005), os autores consideram uma variação na binarização das classes de sentimento. Eles propõem que as opiniões podem obedecer escalas, aumentando sua classificação para três e quatro classes. Os autores coletaram 5.331 *reviews* positivos e a mesma quantidade de negativos.

Hu e Liu (2004) apresentam um corpus composto por 314 *reviews* sobre 5 produtos extraídos do site Amazon ² contendo anotação de polaridades para aspectos. A anotação foi feita em nível de aspecto, pois o trabalho teve o objetivo de sumarizar opiniões de produtos. O corpus ainda foi expandido com *reviews* de mais 12 produtos nos anos seguintes (DING; LIU; YU, 2008; LIU *et al.*, 2015), totalizando 7.356 sentenças.

Blitzer *et al.* (2007) investigaram a adaptação de classificadores de sentimentos para diferentes domínios. Para essa tarefa, os autores compilaram um corpus com *reviews* de quatro tipos de produtos (livros, DVDs, eletrônicos e artigos para cozinha). Eles usaram as notas de usuários para a anotação, presentes em uma escala de 0 a 5. Os *reviews* com nota abaixo de 3 foram anotados como negativos, enquanto os com notas maiores foram anotados como positivos. *Reviews* considerados neutros foram descartados. O corpus final conta com 1.000 *reviews* positivos e 1.000 *reviews* negativos.

Socher *et al.* (2013) utilizaram o corpus apresentado em Pang e Lee (2005) para propor uma anotação mais completa para os *reviews*, rebatizando o corpus com o nome de *Stanford Sentiment Treebank* (SST). Os autores usaram um *parser* (KLEIN; MANNING, 2003) para extrair as árvores sintáticas das 10.662 sentenças do corpus, totalizando 215.154 sintagmas. Após a divisão, os autores usaram o *Amazon Mechanical Turk* para anotar os sintagmas com uma interface contendo uma barra deslizante com 25 valores possíveis, como pode ser observado na Figura 9. Os anotadores recebiam trechos da árvore sintática em diferentes níveis e anotavam a polaridade do trecho usando essa interface.

¹ <www.imdb.com> Acessado em 24 de Fevereiro de 2017.

² <amazon.com> Acessado em 26 de Fevereiro de 2017.

Figura 9 – Interface de anotação contendo 25 possíveis valores em uma barra deslizante.

Fonte: Socher *et al.* (2013).

Essa anotação mais composicional foi motivada pelo método de classificação proposto pelos autores no trabalho. A *Recursive Neural Tensor Network* (RNTN) realiza uma análise composicional usando como entrada a árvore sintática da sentença analisada, portanto era necessário que cada sintagma da composição fosse anotado quanto à polaridade.

O cópus final obtido tem 11.855 sentenças, e isso se deve ao fato de que, durante o *parsing* sintático, algumas sentenças foram divididas, originando novas. Isso fez com que o cópus aumentasse em 1.193 documentos.

Os valores coletados eram normalizados para cinco classes, o que foi denominado pelos autores de análise *fine-grained*. Os dados estão disponíveis em <http://nlp.stanford.edu/sentiment/>, junto com uma interface de demonstração.

O SemEval³ é um evento que vem há anos contribuindo para o avanço da área de PLN. O evento traz competições das mais diversas tarefas, como similaridade semântica, *entailment*, tradução e AS. Geralmente o evento disponibiliza um cópus para treino e grupos de pesquisa interessados se inscrevem e desenvolvem modelos de classificação para esses cópus. Os modelos são enviados e um conjunto de testes desconhecido é utilizado para avaliar os modelos de classificação. Os cópus usados no evento são disponibilizados aos usuários, aumentando a gama de recursos disponíveis para as tarefas.

As tarefas de AS do SemEval trabalham com o *Twitter*, pela facilidade de se obter documentos, pela ampla variedade de assuntos disponíveis e pelos desafios característicos do ambiente. Uma compilação das estatísticas dos cópus já anotados no evento pode ser vista na Tabela 3. Os cópus apresentados são referentes às três últimas edições do evento.

A primeira edição a apresentar uma tarefa de AS foi em 2013 (NAKOV *et al.*, 2013), com duas tarefas: desambiguação de polaridade lexical e classificação de polaridade de mensagens. A primeira tarefa apresentava um fragmento de um *tweet* (uma instância de palavra ou um

³ <http://alt.qcri.org/semeval2018>. Acessado em 26 de Abril de 2018.

Tabela 3 – Estatísticas dos corpúis do SemEval de 2013 até 2015.

Dataset	Positiva	Negativa	Neutra	Total
Twitter2013-train	3.662	1.466	4.600	9.728
Twitter2013-dev	575	340	739	1.654
Twitter2013-test	1.572	601	1.640	3.813
SMS2013-test	492	394	1.270	2.093
Twitter2014-test	982	202	669	1.853
Twitter2014-sarcasm	33	40	13	86
LiveJournal2014-test	424	304	411	1.142
Twitter2015-test	1.040	365	987	2.392

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

sintagma) e os participantes deveriam construir classificadores que determinassem a polaridade (positiva, neutra ou negativa) do fragmento. A segunda tarefa consistia na classificação de mensagens (SMS) nas três polaridades da tarefa anterior. A tarefa deixou como recurso um corpúis anotado para treino e um para validação, assim como dois corpúis para avaliação de cada uma das tarefas (um com *tweets* e um com SMS). Ambos os corpúis de treino e validação consistiam de *tweets*.

Na edição do ano seguinte, as mesmas duas tarefas foram propostas (ROSENTHAL *et al.*, 2014). A organização do evento incentivou os usuários a usarem os corpúis apresentados no ano anterior para treinar seus modelos. Para a avaliação, foram apresentados três corpúis. O primeiro é semelhante ao apresentado no ano anterior, com *tweets* diversos anotados quanto à polaridade. O segundo trata de *tweets* contendo sarcasmo, também anotados em três polaridades. O terceiro corpúis usado na avaliação é composto por sentenças retiradas do *LiveJournal*⁴, outra rede social onde usuários podem criar *blogs* e fóruns sobre assuntos diversos. O corpúis foi criado para identificar se os classificadores treinados com *tweets* poderiam ser usados para classificar sentenças de outras redes sociais.

A edição de 2014 ainda contou com uma tarefa de extração de aspectos, apresentando alguns assuntos relacionados a classificação de polaridade, entretanto, por essa tarefa apresentar mais desafios na extração de aspectos, decidimos não tratar delas nessa monografia.

A popularidade da tarefa em 2014 fez com que os organizadores propusessem mais tarefas de AS para o SemEval de 2015 (ROSENTHAL *et al.*, 2015). A primeira e a segunda tarefas foram mantidas, desambiguação de polaridade contextual e classificação de polaridade de mensagem. Uma tarefa de classificação de polaridade para tópicos em mensagens foi adicionada. Nessa tarefa os classificadores construídos deveriam, dado um tópico e uma mensagem, analisar a sentença para encontrar a polaridade sob a qual o tópico específico estava sendo avaliado. Outra nova tarefa foi a de detecção de tendência para tópico, na qual os modelos construídos deveriam analisar um

⁴ <<http://www.livejournal.com/>> Acessado em 26 de Fevereiro de 2017.

conjunto de *tweets* de um certo período sobre um tópico específico e classificar o sentimento global em cinco classes – fortemente positivo, fracamente positivo, neutro, fracamente negativo, ou fortemente negativo. A última tarefa adicionada foi a de determinar o grau de positividade de um termo. Nessa tarefa eram apresentadas palavras, ou sintagmas, e os classificadores deveriam atribuir um valor entre 0 e 1 indicando o quão positivo esse termo seria.

Na [Figura 10](#) podemos ver uma instrução para a anotação dos *tweets*, assim como uma *screenshot* da interface de anotação usada. A anotação foi abrangente o suficiente para que o mesmo cópus criado pudesse ser utilizado para todas as tarefas propostas.

Figura 10 – Instruções e interface de anotação de *tweets* para o *Mechanical Turk*.

Instructions: Subjective words are ones which convey an opinion or sentiment. Given a Twitter message, identify whether it is objective, positive, negative, or neutral. Then, identify each subjective word or phrase in the context of the sentence and mark the position of its start and end in the text boxes below. The number above each word indicates its position. The word/phrase will be generated in the adjacent textbox so that you can confirm that you chose the correct range. Choose the polarity of the word or phrase by selecting one of the radio buttons: positive, negative, or neutral. If a sentence is not subjective please select the checkbox indicating that “There are no subjective words/phrases”. If a tweet is sarcastic, please select the checkbox indicating that “The tweet is sarcastic”. Please read the examples and invalid responses before beginning if this is your first time answering this hit.

Sentence: A¹ #Christmastree² ..³ Really?⁴ That⁵ can⁶ be⁷ debated...⁸ Merry⁹ XMas¹⁰ to¹¹ Paris.¹² May¹³ it¹⁴ be¹⁵ a¹⁶ jolly¹⁷ holiday¹⁸ ;) ¹⁹
<http://t.co/LDEQwHb62V>²⁰

Overall, the tweet is ☐ Objective ☒ Positive ☐ Negative ☐ Neutral
 The sentiment towards the topic **paris** is ☐ Objective ☒ Positive ☐ Negative ☐ Neutral

☐ The tweet is sarcastic.

☐ There are no subjective words/phrases.

Subjective Phrase 1: to That can be debated... ☐ Positive ☐ Negative ☒ Neutral
 Subjective Phrase 2: to May it be a jolly holiday ;) ☒ Positive ☐ Negative ☐ Neutral
 Subjective Phrase 3: to ☐ Positive ☐ Negative ☐ Neutral

Fonte: [Rosenthal et al. \(2015\)](#).

A última edição ([NAKOV et al., 2016](#)) contou com cinco tarefas distintas: a classificação de polaridade em três classes; a classificação de opinião sobre um determinado tópico; a classificação em cinco classes sobre um determinado tópico; a distribuição de *tweets* positivos e negativos sobre um tópico; e a distribuição em cinco classes de *tweets* sobre um determinado assunto.

A edição mais recente do evento⁵ apresenta três tarefas na área de AS: a classificação de emoção, onde um *tweet* é usado para prever uma intensidade emocional; a previsão de *emojis*, onde *tweets* em inglês e espanhol são disponibilizados e um *emoji* deve ser selecionado como sumário da emoção do documento; e a detecção de ironia.

Na [Tabela 4](#) temos um resumo dos cópus apresentados nessa subseção. Podemos observar que a variação de domínios, tamanhos e anotações tornam difícil a comparação dos cópus de sentimentos. O destaque maior é o *Stanford Sentiment Treebank*, que é o cópus mais

⁵ <http://alt.qcri.org/semeval2018/index.php> Acessado em 27 de Outubro de 2017.

utilizado nas aplicações de *Deep Learning* que vêm atingindo os melhores resultados nos últimos anos (KIM, 2014; TAI; SOCHER; MANNING, 2015; KUMAR *et al.*, 2016).

Tabela 4 – Resumo de corpúis de sentimentos em inglês encontrados na literatura.

Corpúis	Domínio	Anotação	Tamanho do corpúis
Turney (2002)	<i>Reviews</i> diversos	2 classes	410 <i>reviews</i>
Pang, Lee e Vaithyanathan (2002)	<i>Reviews</i> de filmes	2 classes	2.053 <i>reviews</i>
Pang e Lee (2005)	<i>Reviews</i> de filmes	4 classes	10.662 <i>reviews</i>
Hu e Liu (2004)	<i>Reviews</i> de produtos	Aspectos, 5 classes	7.356 sentenças
Blitzer <i>et al.</i> (2007)	<i>Reviews</i> de produtos	2 classes	2.000 <i>reviews</i>
Socher <i>et al.</i> (2013)	<i>Reviews</i> de filmes	5 classes	11.855 sentenças
Nakov <i>et al.</i> (2013)	<i>Tweets</i> e SMS	3 classes	17.288 documentos
Rosenthal <i>et al.</i> (2014)	<i>Tweets</i> e <i>LiveJournal</i>	3 classes	3.081 documentos
Rosenthal <i>et al.</i> (2015)	<i>Tweets</i>	3 classes	2.392 <i>tweets</i>
Nakov <i>et al.</i> (2016)	<i>Tweets</i>	5 classes	22.761 <i>tweets</i>

Fonte: Elaborada pelo autor.

3.1.2 Corpúis em Português Brasileiro

Nesse trabalho vamos apresentar oito corpúis que mais se destacam na AS em português. Assim como acontece nos corpúis criados para a língua inglesa, a variação de domínios, gêneros e o suporte linguístico dos corpúis dificulta a comparação direta entre eles.

Freitas *et al.* (2012) descrevem a anotação de um corpúis de sentimentos do domínio de Resenhas de Livros (ReLi). O ReLi é composto por 1.600 resenhas de 14 livros diferentes, anotadas em nível de sentença em três polaridades. Os autores anotaram manualmente 12.470 sentenças. As resenhas foram obtidas do *website* Skoob⁶, uma rede social de trocas de livros.

Os autores escolheram os livros variando temáticas, estilos de escrita e época de publicação. Essa variação foi feita para que diversos tipos de público fossem analisados, variando desde leituras consideradas mais pesadas (George Orwell, José Saramago) até livros mais populares entre jovens e adolescentes (Talita Rebouças, Stephanie Meyer). Essa decisão tornou mais visíveis siglas, gírias, neologismos e *emoticons* em alguns *reviews* de alguns livros.

O corpúis é desbalanceado, ou seja, a distribuição de sentenças por classe não é equilibrada. Ao todo são 8.994 sentenças anotadas como neutras, 2.883 positivas e 593 negativas. O desbalanceamento das classes em um corpúis pode ocasionar problemas em métodos de AM.

⁶ <www.skoob.com.br> Acessado em 26 de Fevereiro de 2017.

Brum, Araujo e Kepler (2016) usaram o cópus para avaliar classificadores de polaridade e reportaram que o enviesamento de classes prejudicou consideravelmente os resultados obtidos. Balage, Pardo e Aluisio (2013) e Avanço (2015) também citam desafios na avaliação de métodos usando o cópus por causa do desbalanceamento das classes.

Apresentamos um trecho de resenha retirado do cópus na Figura 11. A resenha completa contém diversas sentenças, porém separamos duas para ilustrar como a anotação é feita no ReLi.

Figura 11 – Trecho exemplo do cópus ReLi.

```
#Livro_Se-Houver-Amanhã
#Resenha_113
#Nota_3.0
#Título_113
Frustrante    NPROP    0    op00-    -    0
.             .         0    op00-    -    0

Me            PROPESS   0    op00-    -    0
decepsionei  V         0    op00-    -    0
de            ADV        0    0         -    0
novo          ADV        0    0         -    0
.             .         0    0         -    0

0            ART        0    0         -    0
romance      N          OBJ02 0         -    0
entre        PREP       OBJ02 0         -    0
Tracy        NPROP     OBJ02 0         -    0
e            KC         OBJ02 0         -    0
o            ART        OBJ02 0         -    0
outro        PROADJ     OBJ02 0         -    0
personagem   N          OBJ02 0         -    0
também       PDEN       0    0         -    0
é            V          0    0         -    0
muito        ADV        0    0         -    0
óbvio        ADJ        0    op02-    -    0
```

Fonte: Elaborada pelo autor.

O cópus é composto de sete arquivos de texto, um para cada autor, nos quais encontramos diversas resenhas. No cabeçalho de cada resenha temos o título do livro, uma *id* para a sentença, a nota dada pelo autor e o título da resenha, sendo o último, opcional. Em seguida temos todas as sentenças da resenha e cada palavra desta com cinco etiquetas.

A primeira anotação é a etiqueta morfossintática (*POS-tag*) da palavra. Em seguida temos o alvo da opinião. Os autores definem como alvo entidades referidas como *OBJ*. O livro em questão sempre será o *OBJ00*. No trecho mostrado acima, temos “romance entre Tracy e o outro personagem” como sendo um aspecto do livro (*OBJ02*).

A terceira etiqueta é a polaridade da palavra em relação a um aspecto. A etiqueta é composta da *string op* concatenada com o *id* do aspecto que está avaliando e a polaridade da

opinião (“+” ou “-”). No título da resenha exemplo e na primeira sentença, temos uma indicação de negatividade anotada com “op00-”. A indicação faz referência ao aspecto *OBJ00* (o próprio livro) e traz a polaridade negativa.

A quarta etiqueta diz respeito a polaridade da sentença. Essa etiqueta sempre será igual para todas as palavras da sentença, mesmo que existam mais de uma polaridade em nível de aspecto. Somente 212 sentenças possuem mais de uma polaridade em sua composição.

A última etiqueta do *ReLi* indica a dificuldade da sentença. Em algumas ocorrências ela pode ter o valor “*HELP*”, indicando que a anotação da sentença foi considerada difícil. Existem somente 26 ocorrências desse tipo no *ReLi*.

O *ReLi* foi anotado por três anotadores e os autores usaram a métrica *agr* (WIEBE; WILSON; CARDIE, 2005) para calcular a concordância da anotação. A média de concordância na atribuição de polaridade foi de 98%, enquanto que, para a identificação das frases, objetos e sintagmas, foi, respectivamente, de 81%, 72% e 79%. Isso indica que nesse *ReLi* é mais difícil identificar os alvos das opiniões do que o sentimento associado.

O *ReLi* ainda originou um léxico de sentimentos (FREITAS, 2013) com adjetivos, substantivos, verbos e expressões multipalavras e suas polaridades associadas. Esse léxico contém 616 entradas, sendo 371 adjetivos, 94 substantivos, 88 verbos e 63 expressões multipalavras⁷.

Uma importante abordagem de formação de *ReLi* é a Supervisão Distante. Nessa abordagem, *features* observáveis e definidas manualmente são utilizadas como forte indício de uma informação a ser anotada. Read (2005) propôs um modelo de supervisão distante para o *Twitter* usando *emoticons* para identificar automaticamente polaridade em *tweets*. A proposta foi primeiramente adotada por Go, Bhayani e Huang (2009), quando os autores formularam uma lista de *emoticons* para as polaridades positivas e negativas. A técnica traz como vantagem a rápida anotação de grandes conjuntos de dados, mas carrega consigo a presença de ruídos, visto que o grande número de documentos dificulta muito a revisão dos textos anotados. Posteriormente a técnica foi usada em trabalhos envolvendo o *Twitter* (PAK; PAROUBEK, 2010).

Junior *et al.* (2017) usam essa abordagem para compilar o *Corpus Pelesent*, um vasto *ReLi* de sentimentos do *Twitter* com 980.067 documentos em português, sendo 554.623 positivos e 425.444 negativos. O *ReLi* foi compilado usando um *crawling* de cerca de 41 milhões de *tweets* em português. Não houve qualquer filtragem por domínio, somente pelo idioma e por posição geográfica (usando o Brasil como centro da busca).

Além dos *emoticons* apresentados em Go, Bhayani e Huang (2009) para classificar automaticamente os documentos, os autores construíram uma lista de *emojis* para a mesma finalidade. A diferença entre *emoticons* e *emojis* é que os primeiros são cadeias de caracteres (ex. “:”), “:-D”, “:-(”), enquanto os demais são caracteres especiais que são interpretados por aplicativos

⁷ Disponível em <www.linguateca.pt/Repositorio/ReLi>. Acessado em 8 de Janeiro de 2018.

e sistemas *web* mais modernos. A lista de *emojis* foi empiricamente definida pelos autores, eliminando símbolos que possuísem conotação duvidosa (aparecessem em documentos tanto positivos quanto negativos) em muitos casos. A Figura 12 mostra os *emojis* usados no trabalho.

Figura 12 – Lista de *emojis* usados na Supervisão Distante no córpus Pelesent.

Fonte: Junior *et al.* (2017).

Apesar de representar um grande conjunto de documentos, o Córpus Pelesent não oferece garantias de anotação. A Supervisão Distante pode facilmente acarretar ruídos no *dataset*, seja por ironias no uso dos *emojis*, quanto pelo emprego dos símbolos em documentos neutros ou objetivos. Visando reduzir esses ruídos, os autores removeram documentos que possuísem *emojis* dos dois grupos na mesma postagem.

Alves *et al.* (2014) apresentam um córpus de sentimentos extraído do *Twitter* sobre o torneio de futebol Copa das Confederações 2013. Os *tweets* foram coletados durante quatro meses, usando como palavras-chave alguns termos específicos sobre o torneio (“#copa2014”, “#Brasil2014”, “Copa do Mundo de 2014”, “Copa das Confederações” e “#copadascondeferacoes”). Os autores coletaram cerca de 300.000 *tweets* e realizaram a anotação com duas abordagens, usando Supervisão Distante para a classificação binária, e um trecho anotado manualmente em três classes.

Além da abordagem distante, 1.500 *tweets* foram anotados manualmente por 10 anotadores. Os autores escolheram aleatoriamente os *tweets* anotados pelos usuários, sendo que alguns continham *emoticons*. As anotações manuais conferiram com a indicação dos *emoticons* presentes nos *tweets*, o que os autores interpretaram como validação da anotação por meio de *emoticons*. Ao todo foram anotados 3.070 *tweets*, sendo 1.794 positivos (58%), 463 neutros (15%) e 813 negativos (26%). A baixa incidência de *tweets* neutros pode ter sido ocasionada pelo processo de anotação envolvendo *emoticons*, visto que essa classificação só encontra *tweets* positivos e negativos, não sendo possível inferir que os *tweets* restantes seriam neutros.

Hartmann *et al.* (2014) apresentam um córpus de *reviews* de produtos obtido do site Buscapé⁸. O objetivo do córpus é prover recursos para tarefas de PLN, em especial o estudo linguístico de conteúdos gerados por usuários (CGU). Alguns trabalhos fizeram uso desse córpus na normalização lexical dos CGUs (DURAN *et al.*, 2015; BERTAGLIA; NUNES, 2016). O córpus completo traz 85.910 *reviews*. Apesar de não ser especificamente anotado para análise de

⁸ <www.buscape.com.br> Acessado em 26 de Fevereiro de 2017.

sentimentos, o *corpus* provê notas de usuários nos *reviews* escritos. Essa nota pode servir como base para a identificação de polaridade nas sentenças.

Outro *corpus* muito semelhante ao anterior é o *corpus* Mercado Livre. Ele foi introduzido em [Avanço \(2015\)](#), como um *corpus* de *reviews* de produtos que foi usado tanto na tarefa de normalização quanto na classificação de polaridade. O *corpus* conta com 43.818 *reviews* de produtos extraídos do site Mercado Livre ⁹.

Ambos os *corpus* anteriores (Buscapé e Mercado Livre) não possuem anotação direta para a polaridade das sentenças, entretanto [Avanço \(2015\)](#) fez uma análise manual dos dados para usá-los como *corpus* de sentimentos na tarefa de classificação de polaridades. O autor assumiu que *reviews* com notas 1 e 2 podiam ser considerados negativos, enquanto as de nota 5 seriam considerados positivos. Os *reviews* que receberam nota 0 foram descartados, pois o autor considerou que eles não correspondiam a documentos negativos. Os demais foram considerados nem negativos, nem positivos. Como o objetivo do trabalho era realizar somente a classificação binária de polaridades, a classe neutra não foi abordada no trabalho.

O *corpus* Buscapé ainda possui mais uma etiqueta anotada, a recomendação ou a não-recomendação do produto. [Avanço \(2015\)](#) realizou duas anotações automáticas no *corpus* Buscapé, chamando-as de Buscapé-1 e Buscapé-2. A primeira, já descrita, usa as notas atribuídas pelos usuários. A segunda faz uso somente do campo de recomendação, assumindo como positivos os *reviews* onde o usuário recomendaria o produto, e negativo quando não recomendaria.

Ao todo, o *corpus* Mercado Livre soma 21.499 *reviews* positivos e 21.819 *reviews* negativos, sendo feita a anotação proposta. O *corpus* Buscapé-1 totalizou 6.812 *reviews* positivos e 6.873 *reviews* negativos, enquanto o Buscapé-2 é composto por 2.000 *reviews*, sendo 1.000 positivos e 1.000 negativos.

[Moraes, Manssour e Silveira \(2015\)](#) apresentam o *corpus* 7x1, *corpus* de sentimentos com *tweets* de usuários sobre a seleção brasileira durante a semi-final da Copa do Mundo de 2014. Foram anotados 2.728 *tweets* em português durante a partida “Brasil vs. Alemanha”, que é popularmente lembrada pelo placar elástico de 7 a 1 para a seleção alemã. O *corpus* foi construído com dados da base *WorldCupBrasil2014*, na qual estão disponíveis 851.292 *tweets* em três idiomas (português, inglês e espanhol) durante a Copa do Mundo.

Os *tweets* foram anotados por dois anotadores quanto à polaridade associada a cada mensagem e os anotadores puderam utilizar informações do tempo de jogo e placar para se guiar durante a anotação. Essa informação é importante, pois os acontecimentos durante a partida influenciaram fortemente nos *tweets* obtidos, haja vista a ocorrência de ironias e piadas à medida que a seleção brasileira sofria mais gols.

Foram anotados 157 *tweets* como positivos (6%), 1.771 como neutros (65%) e 800 como sendo negativos (29%). A anotação foi avaliada usando o coeficiente Kappa, atingindo 53% de

⁹ www.mercadolivre.com.br. Acessado em 26 de Fevereiro de 2017.

concordância entre anotadores. Em um trabalho mais recente (MORAES *et al.*, 2016), os autores refizeram a anotação de maneira a aumentar a concordância entre anotadores e atingiram 69%. A distribuição de classes foi modificada, 1.178 *tweets* foram classificados como negativos (43%), 1.101 como neutros (40%) e 449 como positivos (17%).

Moraes *et al.* (2016) apresentam o Computer-BR, também extraído do *Twitter*. Os *tweets* foram encontrados com o uso de palavras-chave relacionadas a computadores, como *notebook*, *análise* e *teste*, e anotados quanto à polaridade em três classes. Quatro anotadores participaram do processo de anotação, sendo três da área da Ciência da Computação e um da Linguística, que serviu como monitor do processo, resolvendo desacordos e possíveis empates. O Computer-BR conta com 2.192 *tweets*, sendo 443 negativos (19%), 1.677 neutros (72%) e 197 positivos (9%). Os autores pré-processaram os *tweets* usando normalizadores, transformando *emoticons* em palavras e removeram *links* e caracteres especiais.

Outras áreas da AS também realizaram compilações de cópus que podem ser aproveitados na classificação de polaridade. Silva *et al.* (2011) analisam a variação de polaridade sobre uma determinada entidade, denominando a tarefa de *Sentiment Stream Analysis*. Essa área difere da classificação de polaridade clássica pois trabalha com grandes fluxos de dados contínuos, preocupando-se mais com a variação geral de polaridade do que com a classificação individual de um documento específico.

Os autores anotaram manualmente um *dataset* contendo 76.358 *tweets* usando como busca dois candidatos à presidência do Brasil nas eleições de 2014 - Dilma Rouseff e José Serra. Ambos os *datasets* possuem anotação manual de polaridade positiva ou negativa em relação aos candidatos. O trecho relacionado a Dilma Rouseff possui 66.640 documentos, sendo 46.805 positivos e 19.835 negativos; já o trecho relacionado a José Serra possui 9.718 documentos, sendo 1.371 positivos e 8.347 negativos.

A diferença entre áreas acaba fazendo com que esse cópus contenha alguns fenômenos indesejados à classificação de polaridade, tais como, a redundância de documentos e o desbalanceamento de classes.

A Tabela 5 resume os cópus de sentimentos para português brasileiro encontrados na literatura. Podemos ressaltar que alguns dos maiores cópus apresentados (Pelesent, Buscapé e Mercado Livre) não são anotados manualmente, assim como boa parte do cópus da Copa das Confederações. Comparando os cópus disponíveis para português com os disponíveis para o inglês, podemos observar uma grande lacuna tanto no tamanho dos cópus quanto no número de cópus anotados manualmente.

Outro ponto a ser ressaltado é a disponibilidade dos cópus oriundos do *Twitter*. Alguns cópus como o 7x1, o cópus da Copa das Confederações e o Cópus Eleições Presidenciais são protegidos contra replicação, o que acaba dificultando seu uso por conta das políticas de privacidade da rede social.

Tabela 5 – Resumo de corpus de sentimentos em português encontrados na literatura.

Córpus	Domínio	Anotação	Tamanho do corpus
Córpus ReLi	Resenhas de livros	3 classes	12.470 sentenças
Córpus Pelesent	<i>Tweets</i> variados	2 classes	980.067 <i>tweets</i>
Copa das Confederações 2013	<i>Tweets</i> sobre futebol	3 classes	3.070 <i>tweets</i>
Buscapé-1	<i>Reviews</i> de produtos	2 classes	13.685 <i>reviews</i>
Buscapé-2	<i>Reviews</i> de produtos	2 classes	2.000 <i>reviews</i>
Mercado Livre	<i>Reviews</i> de produtos	2 classes	43.818 <i>reviews</i>
7 x 1	<i>Tweets</i> sobre futebol	3 classes	2.728 <i>tweets</i>
Computer-BR	<i>Tweets</i> de informática	3 classes	2.317 <i>tweets</i>
Córpus Eleições Presidenciais	<i>Tweets</i> de política	2 classes	76.358 <i>tweets</i>

Fonte: Elaborada pelo autor.

3.2 Análise de Sentimentos

Nesta seção discutiremos trabalhos relacionados com a tarefa de classificação de polaridades. Dividiremos em duas subseções representando os idiomas inglês e português, apresentando os melhores resultados e as abordagens mais influentes na área.

É importante ressaltar que muitos dos trabalhos fazem uso dos corpus já citados, assim como alguns dos trabalhos que apresentaram os corpus também propõem classificadores para essa tarefa.

3.2.1 Métodos usados para inglês

O primeiro trabalho a propor um método de classificação de polaridade foi apresentado em [Turney \(2002\)](#). O autor sugere um método baseado em léxico para a classificação de “recomendação”, ou não, dado um *review* de produto. O autor se refere a essa divisão como “orientação semântica”.

Inicialmente, o autor aplica um etiquetador morfossintático e identifica padrões de bigramas entre os cinco considerados pelo autor – 1) Ocorrência de um adjetivo e um substantivo (singular ou plural); 2) Um advérbio seguido de um adjetivo; 3) Um adjetivo seguido de outro; 4) Um substantivo seguido de adjetivo; 5) Um advérbio seguido de verbo. Os padrões 2, 3 e 4 não podem ser seguidos de um substantivo.

O segundo passo consiste em estimar a orientação semântica dos padrões encontrados usando o algoritmo *Point-wise mutual information* (PMI). O PMI calcula a associação semântica

entre duas palavras por meio da seguinte equação (CHURCH; HANKS, 1990):

$$PMI(word_1, word_2) = \log_2 \left(\frac{P(word_1) \cap P(word_2)}{P(word_1)P(word_2)} \right) \quad (3.1)$$

Para o cálculo da orientação semântica, o autor usa como $word_1$ os bigramas extraídos da sentença, e como $word_2$ dois termos que representam positividade e negatividade. O termo representando positividade é a palavra “*excellent*” e o termo representando negatividade é a palavra “*poor*”. As probabilidades de cada palavra são obtidas via um operador *NEAR*, presente em uma ferramenta de *Information Retrieval* (IR) usada pelo autor. Assumindo um padrão de bigrama encontrado no *review*, *phrase*, podemos calcular a orientação semântica de *phrase* com:

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor") \quad (3.2)$$

No terceiro passo são extraídas as orientações semânticas de todos os bigramas do *review* e é calculada a média. Caso a média seja positiva, o *review* é considerado “recomendado”, em caso contrário, “não recomendado”.

O autor mediu a acurácia da classificação nos *reviews* de carros, bancos, filmes e destinos de viagens, sendo que os melhores resultados foram obtidos nos *reviews* de carros (84,00%) e os piores em filmes (65,83%). Usando todo o corpus, a acurácia total foi de 74,39%.

Pang, Lee e Vaithyanathan (2002) usaram uma abordagem baseada em AM para classificar *reviews* de filmes em positivos e negativos. Os autores usaram três classificadores (Máxima Entropia, Naive Bayes e SVM) com diversas combinações de *features*.

Os autores usaram uma *bag-of-words* com *features* de representação de documentos. Essa abordagem testa diversas combinações de *features* para obter os melhores resultados para cada classificador. As *features* usadas na modelagem foram unigramas, bigramas, *POS-tagging*, adjetivos e a posição das palavras no *review*. Os autores mediram a acurácia dos classificadores (fazendo *cross-validation*) nas oito seguintes combinações de *features*:

- **Frequência de unigramas:** Representar os *reviews* com uma *bag-of-words* com a frequência de cada termo. Os melhores resultados foram obtidos com Naive Bayes (78,7% de acurácia).
- **Presença de unigramas:** Representar os *reviews* com uma *bag-of-words* com a presença (0 ou 1) de cada unigrama. O melhor resultado foi obtido com SVM (82,9% de acurácia).
- **Unigramas e bigramas:** Representar os *reviews* com bigramas e unigramas concatenados. Foi usada somente a presença/ausência dos termos na modelagem e a melhor acurácia foi relatada com SVM (82,7%).

- **Presença de bigramas:** Representar os *reviews* somente usando presença/ausência de bigramas. O melhor resultado foi obtido com Máxima Entropia (77,4%)
- **Unigramas e POS-tags:** Representar os *reviews* com unigramas concatenando as palavras com sua etiqueta morfossintática. Os autores propuseram essa alternativa para reduzir a ambiguidade dos termos, por exemplo, diferenciando “*I love this movie*” (positivo) de “*This is a love story*” (neutro). No exemplo citado, a primeira instância de *love* seria um verbo, enquanto a segunda um adjetivo. O melhor resultado obtido foi com o classificador SVM (81,9%), mas ainda ficou abaixo da modelagem usando somente unigramas.
- **Adjetivos:** Nessa modelagem foram usados somente os adjetivos para representar os *reviews*. O melhor resultado foi obtido com Máxima Entropia (77,7%), mas ainda assim foi mais baixo do que esperado pelos autores.
- **Sub-conjunto de unigramas:** Os autores buscaram eliminar unigramas menos frequentes da modelagem, reduzindo os 16.165 unigramas menos frequentes para uma representação com 2.633. O corte conseguiu obter resultados quase tão bons quanto com todos os unigramas usando SVM (81,4%).
- **Unigramas e posição:** Os autores propuseram um modelo de representação que leva em consideração a posição das palavras no *review*. Segundo eles, uma estrutura de *review* de filme inicia com a discussão do enredo e termina resumizando as opiniões do autor. Eles adicionaram junto dos unigramas a informação de posição (dividindo o *review* em quatro partes iguais). O melhor resultado foi obtido com SVM (81,6%).

O trabalho é um dos pioneiros na AS, fazendo um estudo linguístico e experimentando muitas combinações de *features* para a classificação binária de polaridades. O melhor resultado obtido foi usando SVM modelado com unigramas (82,9% de acurácia).

Um ramo da AM que vem crescendo em popularidade é o *Deep Learning* (DL). As arquiteturas multi-camadas encontraram na AS um desafio que combina a complexidade da PLN com as características subjetivas da semântica.

Socher *et al.* (2013) foi um dos trabalhos mais importantes para a popularização da AS entre os modelos de DL. Os autores apresentam um *corpus* anotado em cinco classes de polaridade para *reviews* de filmes, junto de uma arquitetura neural recursiva que faz uso de um tensor para analisar composicionalmente as sentenças e inferir sua polaridade.

Os autores propuseram duas maneiras de avaliar o *corpus*, usando as cinco classes analisadas, e usando somente duas (pos/neg). O *corpus* rapidamente se tornou uma espécie de *benchmark* para modelos de DL. A seguir apresentaremos sucintamente alguns métodos de DL que usaram o *corpus* SST.

Le e Mikolov (2014) usaram uma modificação do *word2vec* (MIKOLOV *et al.*, 2013), um modelo de espaço vetorial que apresenta indícios de captura de informações semânticas e sintáticas das palavras dado seu contexto. Os autores propuseram um modelo semelhante ao *word2vec*, chamado *Paragraph Vector*, que extrai representações vetoriais de documentos. Os autores usaram Regressão Logística para predizer as polaridades das sentenças do SST usando as representações geradas pelo *Paragraph Vector*.

Além desses modelos, outras abordagens de DL se destacaram na classificação de polaridades, como a rede neural convolucional proposta por Kim (2014), a arquitetura em árvore usando neurônios de *long short-term memory*, proposta em Tai, Socher e Manning (2015), a rede de memória dinâmica proposta em Kumar *et al.* (2016), e a arquitetura bidirecional proposta em Zhou *et al.* (2016). A Tabela 6 apresenta os resultados obtidos com os modelos aqui apresentados usando a acurácia como medida comparativa.

Tabela 6 – Resultados dos trabalhos apresentados avaliados no corpus *Stanford Sentiment Treebank*.

Trabalho	<i>Fine-grained</i>	Classificação binária
Socher <i>et al.</i> (2013)	45,7%	85,4%
Le e Mikolov (2014)	48,7%	87,8%
Kim (2014)	48,0%	88,1%
Tai, Socher e Manning (2015)	51,0%	88,0%
Kumar <i>et al.</i> (2016)	52,1%	88,6%
Zhou <i>et al.</i> (2016)	52,4%	89,5%

Fonte: Elaborada pelo autor.

Alguns trabalhos apresentam uma abordagem semissupervisionada para gerar classificadores que possam treinar com dados anotados e não anotados. Dados não anotados são geralmente mais numerosos que dados anotados e mais fáceis de serem encontrados.

Dasgupta e Ng (2009) propõem uma abordagem semissupervisionada na classificação binária de polaridade usando *active learning* (COHN; ATLAS; LADNER, 1994; TONG; KOLLER, 2001) para anotar *reviews* com grande incerteza na anotação.

Os autores propõem uma abordagem “Minere os fáceis e classifique os difíceis”. A definição de *reviews* “fáceis” vem da ausência de ambiguidade na polaridade do documento, enquanto o oposto é considerado uma anotação “difícil”. Para classificar esses *reviews*, os autores propõem o uso de um algoritmo de agrupamento espectral (NG *et al.*, 2001) para agrupar *reviews* não ambíguos em dois *clusters* (positivos e negativos). Esse processo é iterativo e inicia com o algoritmo agrupando todos os *reviews* no plano. A cada iteração, os *reviews* mais ambíguos vão sendo removidos até que restem dois *clusters* de *reviews*, um positivo e um negativo.

Após a formação desse corpus anotado automaticamente, os autores usam um classifica-

dor SVM para classificar os demais documentos. Nesse ponto é feita a incorporação do *active learning*, onde um processo de aprendizado semissupervisionado recebe alguns documentos anotados para os quais o classificador obteve maior incerteza na classificação. Esses documentos seriam aqueles cuja classificação os tenha colocado mais próximos do hiperplano de corte.

O SVM é treinado com os dados anotados e aplicado no restante do corpus. Os 10 documentos mais próximos do hiperplano (com maior incerteza) são anotados manualmente e o modelo é retreinado com esses dados adicionados ao conjunto de treinamento. Além disso, os autores propõem a utilização de um *ensemble* com cinco classificadores, treinados com trechos diferentes do corpus anotado.

Para a avaliação do método, os autores usaram os corpus propostos em Pang, Lee e Vaithyanathan (2002), de *reviews* de produtos, e Blitzer *et al.* (2007), do mesmo domínio. As métricas utilizadas foram a acurácia dos classificadores e *Adjusted Rand Index* (usada com tarefas de agrupamento). Os autores compararam com três *baselines*: o algoritmo de *clustering* espectral proposto em Kamvar *et al.* (2003); um classificador SVM transdutivo (WANG; SHEN; PAN, 2007); e uma abordagem de *active learning* descrita em Tong e Koller (2001).

A Tabela 7 apresenta os resultados obtidos em cinco corpus, sendo o primeiro (MOV), o corpus de *reviews* de filmes apresentado em Pang, Lee e Vaithyanathan (2002) e, os demais, os conjuntos apresentados em Blitzer *et al.* (2007). As primeiras três linhas se referem aos *baselines*. A linha 4 trata do uso do SVM nos primeiros dados, a linha 5, após o uso de *active learning*, e a linha 6, com o uso do *ensemble* de cinco classificadores.

Tabela 7 – Resultados de acurácia e ARI para os dados..

System Variation	Accuracy					Adjusted Rand Index				
	MOV	KIT	ELE	BOO	DVD	MOV	KIT	ELE	BOO	DVD
1 Semi-supervised spectral learning	67.3	63.7	57.7	55.8	56.2	0.12	0.08	0.01	0.02	0.02
2 Transductive SVM	68.7	65.5	62.9	58.7	57.3	0.14	0.09	0.07	0.03	0.02
3 Active learning	68.9	68.1	63.3	58.6	58.0	0.14	0.14	0.08	0.03	0.03
4 Our approach (after 1st step)	69.8	70.8	65.7	58.6	55.8	0.15	0.17	0.10	0.03	0.01
5 Our approach (after 2nd step)	73.5	73.0	69.9	60.6	59.8	0.22	0.21	0.16	0.04	0.04
6 Our approach (after 3rd step)	76.2	74.1	70.6	62.1	62.7	0.27	0.23	0.17	0.06	0.06

Fonte: Dasgupta e Ng (2009).

Em Silva *et al.* (2016) os autores propõem um *framework* para aprendizado semissupervisionado para classificação de *tweets*. A abordagem faz uso do algoritmo C³E (ACHARYA *et al.*, 2011). O algoritmo combina classificação e *clustering*, usando uma matriz de similaridade para definir a distribuição de probabilidades de classes para cada instância dos documentos.

Os autores utilizaram corpus do eventos SemEval (NAKOV *et al.*, 2016) e compararam a abordagem com um classificador SVM, uma abordagem de *self-training* e uma abordagem de *co-training* (ambas utilizando SVM) – uma abordagem baseada em léxico também foi avaliada. Os dados usados para o treinamento inicial do modelo variaram de 5% até 40%, sendo que quanto mais dados foram utilizados, maior foi a *F-Measure* obtida na avaliação.

A abordagem dos autores obteve melhores resultados nos corpúis *LiveJournal* e *SMS2013* usando a partir de 10% dos dados como treinamento inicial. No corpúis *Twitter2014* o método superou a *F-Measure* dos demais usando 5% dos dados, 20%, 30% e 40%. A *F-Measure* mais alta obtida foi 68,51% usando a abordagem dos autores no corpúis *LiveJournal*.

3.2.2 Métodos usados para português brasileiro

A classificação de sentimentos ainda não é tão popular no idioma português quanto no inglês. A maioria dos trabalhos usam abordagens baseadas em léxico e AM, em contraponto ao uso de *Deep Learning* para o inglês. Isso pode ser decorrência do fato de que não existem disponíveis muitos corpúis de sentimentos para português. Os desafios da comparação entre trabalhos também se aplicam ao idioma, visto que os corpúis e os tipos de classificação variam muito nos diferentes trabalhos.

[Avanço \(2015\)](#) implementou uma série de classificadores de polaridade para português brasileiro. O autor teve como objetivo desenvolver classificadores para duas classes no domínio de *reviews* de produtos. Foram desenvolvidos oito classificadores, usando-se três abordagens diferentes: classificadores baseados em léxico, baseados em AM e um classificador híbrido que combina as abordagens anteriores.

Os classificadores baseados em léxico não possuem etapa de treinamento e se baseiam no uso de léxicos de sentimentos para realizar a classificação. Os léxicos usados pelo autor no trabalho foram o Sentilex ([SILVA; CARVALHO; SARMENTO, 2012](#)), o Opinion Lexicon ([SOUZA et al., 2011](#)), um subconjunto do OntoPT ([OLIVEIRA; SANTOS; GOMES, 2014](#)) e uma tradução do inglês do léxico LIWC ([BALAGE; PARDO; ALUISIO, 2013](#)). Os léxicos foram agrupados em um arquivo único e serviram como base de cálculo para alguns dos métodos discutidos a seguir. A seguir apresentamos os 3 classificadores propostos no trabalho.

- **Baseline:** O classificador mais intuitivo construído. As palavras de uma sentença são percorridas e buscadas no léxico de sentimentos. Em seguida é agregada a polaridade 1 para as palavras positivas e -1 para as palavras negativas. As polaridades são somadas e, caso a polaridade seja maior que zero, a sentença é considerada positiva, do contrário, é considerada negativa.

O método intuitivo ignora algumas características da língua, como negação, por exemplo. A sentença “Eu não achei bom”, seria classificada como positiva, visto que a única palavra de sentimento presente é “bom” e o método não trata a negação na sentença.

- **Classificador de opiniões baseado em léxico (CBL):** O autor também desenvolveu um classificador baseado em léxico seguindo a proposta de [Taboada et al. \(2011\)](#). No modelo, são propostas três alternativas para o tratamento de negações, intensificações e reduções de polaridade.

O método percorre a sentença em busca das palavras de sentimento. Encontrando uma palavra, ele busca por palavras vizinhas (em uma janela de tamanho 3) que indiquem negação, intensificação ou redução, com base na lista de palavras da [Figura 13](#).

Figura 13 – Conjunto de palavras de negação, intensificação e redução.

Negação	Intensificação	Redução
<i>jamaís, nada, nem, nenhum, ninguém, nunca, não, tampouco</i>	<i>mais, muito, demais, completamente, absolutamente, totalmente, definitivamente, extremamente, frequentemente, bastante</i>	<i>pouco, quase, menos, apenas</i>

Fonte: [Avanço \(2015\)](#).

Caso seja encontrada uma palavra de negação vizinha à palavra de sentimento, o sentimento associado (1 ou -1) é multiplicado por -1, invertendo a polaridade. Encontrando uma palavra de intensificação, a polaridade é multiplicada por 3 e, em caso de redução, dividida pelo mesmo valor. Caso sejam encontradas palavras de negação e de intensificação na janela da palavra de sentimento, o classificador inverte a intensificação da polaridade. Por exemplo, na sentença “O filme não foi muito bom”, temos uma negação (não) e uma intensificação (muito), portanto a polaridade associada com “bom” é dividida por 3.

- **Classificador baseado em léxico usando modelo de espaço vetorial (CBL-MEV:** Os modelos de espaço vetorial buscam representar palavras baseando-se no seu contexto, gerando representação que, geralmente, podem carregar semelhanças sintáticas e semânticas entre palavras. O autor usou o modelo proposto por [Mikolov et al. \(2013\)](#) para extrair representações vetoriais das palavras.

O autor elencou “palavras-semente” positivas e negativas, sendo as primeiras, as 10 palavras mais similares (usando distância de cosseno) à palavra “bom”, e as últimas, as 10 mais similares à palavra “péssimo”. Usando o léxico de sentimentos, o classificador busca nas sentenças as palavras que estejam no léxico, porém, em vez de usar a polaridade associada descrita nele, é calculada a similaridade (novamente usando distância de cosseno) da palavra com todas as “palavras-semente” positivas e negativas. A similaridade da palavra com todas as palavras-semente positivas é somada, assim como com as negativas, e a que obtiver o maior valor é considerada a polaridade da palavra.

Esse método foi proposto baseando-se na mudança de polaridade que as palavras podem sofrer dependendo do contexto, em que está inserida. Podemos ter uma palavra que representa sentimento positivo no léxico, mas que, entretanto, no domínio de *reviews* de

produtos apresenta características negativas - um exemplo é o adjetivo *barato*, que consta nos léxicos como negativo, entretanto é um indicador de positividade em diversos *reviews*.

Os classificadores baseados em AM usados pelo autor foram o Naive Bayes e o SVM. Para a modelagem das *features*, o autor usou *bag-of-words* com presença/ausência de termos, quantidade de emoticons positivos e negativos, quantidade de palavras positivas e negativas (usando léxico de sentimentos) e a quantidade de adjetivos, advérbios, substantivos e verbos. Também foi feita uma seleção de *features* no SVM, reduzindo a dimensão do vetor de *features* de 7.000 para 1.600.

O autor também construiu um classificador híbrido, que combina o C-SVM (que obteve o melhor resultado nos experimentos com AM) com o CBL (que obteve os melhores resultados entre os baseados em léxico). Esse classificador usa o C-SVM para classificar um *review* e assume incerteza quando a classificação do SVM está muito próxima ao hiperplano de corte. Nesse caso, o classificador híbrido usa o CBL para classificar o *review*.

Os classificadores baseados em léxico e AM foram avaliados nos corpúscos Buscapé e Mercado Livre, sendo medidas a *F-Measure* e Acurácia das classificações (usando *cross-validation*). Os melhores resultados obtidos no corpúscos Mercado Livre foram usando o C-SVM com seleção de *features*, atingindo 95,6% para *F-Measure* e Acurácia. O melhor resultado para o corpúscos Buscapé foi com o mesmo classificador, também realizando seleção de *features* e atingindo 91,77% de *F-Measure* e 91,78% de Acurácia.

O classificador híbrido foi usado em experimentos com domínios diferentes, no caso, no corpúscos ReLi, de resenhas de livros. O melhor resultado obtido foi 63,04% de *F-Measure*, o que é significativamente abaixo do resultado obtido quando os conjuntos de treino e teste pertencem ao mesmo corpúscos. A Acurácia não pôde ser medida, pois o corpúscos ReLi é desbalanceado.

O trabalho foi expandido em [Avanço, Brum e Nunes \(2016\)](#), com a adição de um comitê de classificadores (*ensemble*). A abordagem combina a saída de diferentes classificadores para fazer a predição da classe. Os autores combinaram os oito classificadores apresentados em [Avanço \(2015\)](#) usando votação majoritária (C-ENS), onde cada classificador possui a mesma influência na classificação, e usando votação ponderada (C-WENS), onde foram usados os valores de acurácia para dar mais influência a classificadores que apresentaram resultados melhores na avaliação individual.

Foram realizados experimentos nos corpúscos Buscapé e Mercado Livre. Os melhores resultados foram obtidos usando o C-ENS, com 95,83% de *F-Measure* e Acurácia no corpúscos Mercado Livre. O melhor valor encontrado para o corpúscos Buscapé foi 89,35% de *F-Measure* e Acurácia.

[Balage, Pardo e Aluisio \(2013\)](#) propõem uma avaliação dos léxicos de sentimentos para português brasileiro. Realizando uma avaliação extrínseca, os autores realizaram classificação de sentimentos no corpúscos ReLi para avaliar os léxicos.

O método utilizado foi o SO-CAL (TABOADA *et al.*, 2011), semelhante ao apresentado em Avanço (2015). Os autores fizeram experimentos com três léxicos de sentimentos diferentes: o subconjunto do LIWC, o Opinion Lexicon e o Sentilex. Os experimentos avaliaram a *F-Measure* e a Acurácia da classificação no corpus ReLi.

O LIWC obteve o melhor resultado de Acurácia, 57,33%, e *F-Measure* de 51,71%. Os experimentos com o Opinion Lexicon obtiveram 47,43% de acurácia e 49,27% e o Sentilex atingiu 44,17% e 53,01% de Acurácia e *F-Measure*, respectivamente.

Brum, Araujo e Kepler (2016) realizaram experimentos com dois classificadores de sentimentos no corpus ReLi para três classes. Os autores investigaram o modelo de espaço vetorial proposto em Le e Mikolov (2014), que cria representações vetoriais para um documento inteiro, em vez de para cada palavra, usando Regressão Logística para a classificação. O outro classificador usado foi a RNTN proposta em Socher *et al.* (2013), onde uma análise composicional é feita usando uma arquitetura multi-camada, classificando os sintagmas da composição da sentença.

Os autores investigaram o desbalanceamento das classes do corpus ReLi, propondo duas alternativas para melhorar a classificação das resenhas, de acordo com Monard e Batista (2002). A primeira foi o *over-sampling*, onde os documentos do conjunto de treinamento das classes menos ocorrentes são replicados até que o corpus seja balanceado. A segunda foi o *under-sampling*, onde documentos da classe mais representada são descartados no treinamento do modelo. O melhor resultado foi atingido usando-se *over-sampling* e a RNTN, obtendo 82,85% de Acurácia. Entretanto a medida não é recomendada neste corpus, por conta do desbalanceamento das classes.

Alves *et al.* (2014) usaram o corpus proposto no mesmo trabalho sobre a Copa das Confederações para avaliar dois classificadores baseados em AM. Os autores propõem um classificador SVM e um Naive Bayes para a classificação de polaridade em duas classes.

O corpus utilizado foi anotado em três classes, portanto a classe neutra foi usada na avaliação. Isso adicionou uma etapa de classificação para identificar se os *tweets* eram subjetivos (contêm polaridade) ou objetivos (neutros).

A anotação do corpus foi realizada em parte manualmente, e também automaticamente (por meio de *emoticons*). Os autores realizaram experimentos com o corpus completo e com as duas partes individuais. Os classificadores foram avaliados quanto a *F-Measure* e Acurácia na classificação de subjetividade e polaridade.

O melhor resultado na classificação de subjetividade foi atingido pelo classificador SVM, com 84% de Acurácia e 82,1% de *F-Measure*. O classificador Naive Bayes obteve 82% e 81,9% de Acurácia e *F-Measure*, respectivamente. A avaliação da classificação de polaridade realizada no corpus completo teve Acurácia de 80% com o classificador SVM e 77% com Naive Bayes. A *F-Measure* foi de 80% e 78,3%, respectivamente. Nos experimentos com a parte anotada manualmente, o SVM obteve maior Acurácia (65,6%), apesar de uma

F-Measure baixa (66,1%). O classificador Naive Bayes obteve Acurácia muito próxima (65%) e *F-Measure* superior (67,2%). No cópús anotado automaticamente, o SVM obteve resultados muito superiores aos do Naive Bayes. O primeiro atingiu 87% de Acurácia e 87,3% de *F-Measure*, enquanto o outro atingiu 72,7% de Acurácia e 73,3% de *F-Measure*.

Moraes *et al.* (2016) apresentam dois métodos de classificação de polaridades para *tweets* de domínios diferentes. Os autores usam um modelo baseado em léxico e um classificador SVM para a classificação. Semelhante a Alves *et al.* (2014), são propostas duas etapas na classificação, sendo a primeira para separar *tweets* objetivos de subjetivos (que contêm polaridade), e a segunda para definir as polaridades dos *tweets* subjetivos.

O classificador baseado em léxico usou o Sentilex (SILVA; CARVALHO; SARMENTO, 2012) e o WordNetAffectBr (PASQUALOTTI; VIEIRA, 2008) como recursos. Para a primeira etapa, são considerados objetivos todos os *tweets* que não contenham palavras associadas a positividade ou negatividade nos léxicos de sentimentos. Para a segunda etapa, a classificação de polaridade, o método soma a polaridade das palavras (atribuindo 1 para as positivas e 0 para as negativas) dos *tweets* e, caso a soma resulte em número positivo, o *tweet* é classificado como tal, do contrário, o *tweet* é classificado como positivo.

Para a representação dos *tweets* no modelo de AM, os autores usaram a *Comprehensive Measurement Feature Selection* (CMFS), proposta por Yang, Qu e Liu (2014). Essa medida é calculada usando a probabilidade condicional de uma palavra w_k , dada uma classe c_j , multiplicada pela probabilidade da classe c_j , dada a palavra w_k , dividida pela probabilidade a priori da palavra.

$$CMFS(w_k, c_j) = \frac{P(w_k|c_j)P(c_j|w_k)}{P(w_k)} \quad (3.3)$$

A CMFS é usada para avaliar a relevância das palavras de cada classe. Os autores usaram a medida para ranquear as palavras mais relevantes de cada classe e remover as menos relevantes durante a classificação. Foram realizados experimentos removendo palavras comuns às duas classes, o que se mostrou mais eficiente.

Os autores representaram as sentenças usando *bag-of-words* com os termos mais relevantes, com presença/ausência de termos para a classificação de subjetividade. Para a classificação de polaridade, foi usado um léxico de sentimentos na composição do vetor de *features*. O léxico de sentimentos usado na tarefa é composto pelo Sentilex (SILVA; CARVALHO; SARMENTO, 2012) e pelo WordNetAffectBR (PASQUALOTTI; VIEIRA, 2008).

Além da abordagem em duas etapas, os autores também apresentaram uma abordagem direta de classificação, em etapa única. Ambas foram avaliadas quanto a Acurácia nos cópús 7x1 e Computer-BR. O melhor resultado obtido no cópús 7x1 foi 57,03% de Acurácia; no cópús Computer-BR o melhor resultado foi 76,37% da mesma medida. Ambos os resultados foram obtidos com o modelo de duas etapas classificando com SVM.

[Junior et al. \(2017\)](#) trazem uma proposta de Supervisão Distante na criação do *Cópus Pelesent*, citado anteriormente na [Subseção 3.1.2](#). Os autores comparam quatro abordagens de AM na tarefa de classificação de polaridade sobre os *cópus* Eleições Presidenciais, Buscapé e Mercado Livre. Os autores sugerem que o treinamento com dados anotados à distância consegue se igualar ou superar os valores obtidos pelos métodos do estado-da-arte testados em *cross-validation* sobre os dados.

Os quatro classificadores usados foram um Regressor Logístico (usando *bag-of-words* com TF-IDF, *word2vec* e *doc2vec* como modelos de representação dos dados), o modelo de CNN proposto por [Kim \(2014\)](#), um modelo de CNN recorrente, usado em [Treviso, Shulby e Aluísio \(2017\)](#), para segmentação de sentenças, e o modelo híbrido proposto por ([AVANÇO, 2015](#)).

Os experimentos sobre o *cópus* de Eleições Presidenciais ([SILVA et al., 2011](#)) atingiram os melhores valores de *F-Measure* e Acurácia com o Regressor Logístico usando *bag-of-words* com TF-IDF, respectivamente 64,77% e 71,28%. Sendo o *cópus* proposto para uma tarefa diferente (*Sentiment Stream Analysis*), não se pode comparar os resultados com aqueles obtidos na tarefa de classificação de polaridades.

Os experimentos sobre o *cópus* Buscapé ([AVANÇO, 2015](#)) resultaram no valor máximo de 76,68% de *F-Measure* e 76,95% de Acurácia com o Classificador Híbrido. Esses valores não superaram os valores originais de estado-da-arte obtidos em [Avanço, Brum e Nunes \(2016\)](#).

No *cópus* Mercado Livre o classificador Híbrido também obteve os melhores resultados usando o *cópus* Pelesent, 86.14% de *F-Measure* e Acurácia. Esse valor também não superou o estado-da-arte de 95,83% descrito em [Avanço, Brum e Nunes \(2016\)](#).

A [Tabela 8](#) apresenta um resumo dos melhores resultados obtidos nos trabalhos discutidos. O trabalho de [Junior et al. \(2017\)](#) apresenta avaliações sobre os três *cópus* avaliados no artigo - o *cópus* *Reviews* de produtos¹ representa o *Cópus* Buscapé ([HARTMANN et al., 2014](#)), enquanto *Reviews* de produtos² representa o *Cópus* Mercado Livro ([AVANÇO, 2015](#)).

3.3 Discussão da Revisão

A AS, apesar de ser recente, apresentou um rápido desenvolvimento nos últimos 10 anos. Quando observamos os *cópus* de sentimentos presentes na literatura, o idioma Inglês se sobressai ao Português graças a eventos ([NAKOV et al., 2016](#)) e esforços de grandes centros de pesquisa ([SOCHER et al., 2013](#)), originando diversas bases de dados com dezenas de milhares de documentos.

A anotação dessas bases de dados ainda compreende um espectro maior de tarefas,

Tabela 8 – Resumo dos trabalhos de classificação de polaridades para a língua portuguesa.

Trabalho	Métodos	Córpus	Classificação	F-Measure	Acurácia
Avanço (2015)	Léxico, SVM, Naive Bayes e abordagem híbrida	Reviews de produtos	2 classes	95,60%	95,60%
Avanço, Brum e Nunes (2016)	Ensemble (votação majoritária e votação ponderada)	Reviews de produtos	2 classes	95,86%	95,86%
Balage, Pardo e Aluisio (2013)	SO-CAL	Resenhas de livros	2 classes	51,71%	57,33%
Brum, Araujo e Kepler (2016)	Modelo de espaço vetorial e rede neural recursiva	Resenhas de livros	3 classes	-	82,85%
Alves <i>et al.</i> (2014)	SVM e Naive Bayes	Tweets de futebol	3 classes	80,00%	80,00%
Moraes <i>et al.</i> (2016)	Léxico e SVM	Tweets de informática	3 classes	-	76,37%
Junior <i>et al.</i> (2017)	Regressor logístico, CNN, RCNN e abordagem híbrida	Tweets de política	2 classes	64,77	71,28%
		Reviews de produtos ¹	2 classes	79,17	79,34%
		Reviews de produtos ²	2 classes	86,14	86,14%

Fonte: Elaborada pelo autor.

destaques ao sarcasmo (ROSENTHAL *et al.*, 2014) e à marcação da classe neutra (NAKOV *et al.*, 2013), o que faz com que alguns corpú da língua inglesa possibilitem estudos mais complexos dos fenômenos opinativos e sua aplicação em conjunto de outras tarefas, como a sumarização e a identificação de aspectos (HU; LIU, 2004).

A padronização dos corpú também facilita a comparação direta entre métodos, motivando pesquisadores na proposta de novos métodos (como as arquiteturas de *Deep Learning*). Os bons resultados ainda possibilitam que problemas mais complexos, como a análise em 5 classes, sejam investigados (SOCHER *et al.*, 2013).

Apesar da diferença entre idiomas, o Português Brasileiro possui um número considerável de bases de dados, porém a dificuldade na obtenção das bases de dados e algumas características de balanceamento, redundância de dados e ausência de garantia de confiabilidade de anotação fazem com que cada vez mais autores construam seus próprios recursos.

Essa preferência pela construção de uma base nova favorece a área, dando maior diversidade de dados para análise, porém dificulta consideravelmente a comparação de métodos. Além disso, as diferentes anotações em cada corpú dificultam alguns métodos construídos especificamente para um domínio ou para uma tarefa.

Outro quesito a ser observado é a presença de grandes bases de dados que possuem anotação automática baseada em pontuações objetivas fornecidas por usuários (estrelas ou notas) que nem sempre representam o verdadeiro alinhamento semântico dos documentos (JUNIOR *et al.*, 2017; AVANÇO, 2015; HARTMANN *et al.*, 2014).

A diversidade de domínios e gêneros não se repete em relação ao número de classes - poucos autores investigam além da classificação binária. Essa ausência pode indicar a dificuldade

que a classe neutra representa e pode ser uma boa alternativa para que melhores resultados sejam obtidos, porém afasta os métodos de uma classificação mais realista.

Sobre os métodos, podemos observar que nos últimos anos os métodos de *Deep Learning* vêm se sobressaindo na literatura de AS para a língua inglesa. Talvez uma explicação sejam as bases de dados bem construídas que podem ser encontradas para tal idioma.

Em Português observamos o mesmo fenômeno, porém ainda em escala menor. O uso de SVM (ALVES *et al.*, 2014), abordagens baseadas em Léxico (MORAES *et al.*, 2016) e *Ensembles* (AVANÇO; BRUM; NUNES, 2016) ainda são preferidos, obtendo bons resultados nos corpúis existentes. Ainda podemos observar que os melhores valores obtidos por Junior *et al.* (2017) foram atingidos usando SVM aliado a um classificador baseado em léxico, sendo que experimentos com redes neurais convolucionais e recursivas não conseguiram superar a abordagem.

Pelos resultados obtidos, os corpúis Buscapé e Mercado Livre são os que apresentaram maiores valores de F-measure e Acurácia na tarefa de classificação de polaridades, isso pode indicar que as representações usadas por Avanço (2015) e Avanço, Brum e Nunes (2016) sejam suficientes para determinar de maneira eficiente a diferença entre documentos positivos e negativos. O mesmo não pode ser dito em bases de dados oriundas do Twitter (ALVES *et al.*, 2014; MORAES *et al.*, 2016).

O Português Brasileiro ainda carece de experimentos em abordagens semissupervisionadas na classificação. Alguns autores (DASGUPTA; NG, 2009; SILVA, 2016) conseguiram atingir bons resultados usando tais abordagens em bases de dados em Inglês, portanto a exploração dessas técnicas pode ser uma boa alternativa para o Português.

FRAMEWORK DE EXPANSÃO DE RECURSOS

Como visto na [Seção 1.2](#), buscamos a construção de um *córpus* de sentimentos confiável e anotado semi-automaticamente, usando classificadores de AM aliados à abordagens semissupervisionadas para nos dar maior garantia de confiança na classificação.

Optamos por dar foco à classificação de polaridades em três classes (positiva, neutra e negativa), assim sendo, um dos primeiros objetivos foi a busca por recursos linguísticos que nos permitissem realizar treinamento, expansão e validação de nossas questões de pesquisa.

Apesar dos *córpus* apresentados no [Capítulo 3](#), decidimos construir uma base de dados inédita para a tarefa. Isso nos possibilitou maior controle sobre as garantias de confiança na anotação do *córpus*, além de gerar um novo recurso para a comunidade de AS no Brasil.

Além do *córpus* de treinamento, foi importante definirmos os classificadores utilizados na expansão, portanto buscamos na literatura por técnicas, modelagens e abordagens que melhor se adaptassem ao domínio de trabalho e à tarefa de classificação de polaridades.

Por último utilizamos abordagens de aprendizado semissupervisionado iterativas para expandir o *córpus* anotado manualmente com dados não anotados de mesmo domínio dos contidos no *córpus* de treinamento.

Construímos nosso trabalho em dois eixos (extração de dados e anotação; e classificação e expansão). A [Figura 14](#) apresenta o fluxograma de processos do *framework* proposto. A seguir discutiremos cada uma das etapas e, ao longo do capítulo, detalharemos os processos.

A primeira etapa da [Figura 14](#) compreende a extração de *tweets* e a construção de um *córpus* de sentimentos anotado. O principal objetivo dessa etapa foi definir com clareza a metodologia de extração de documentos (domínio, termos de busca, origem dos dados), elaborar um manual de anotação e aplicar os conceitos de anotação de *córpus* vistos na literatura.

Figura 14 – Fluxograma do *framework* de expansão de córpus proposto.

Fonte: Elaborada pelo autor.

Considerando que a posterior expansão do córpus anotado manualmente depende diretamente dos dados não anotados, nessa etapa também realizamos a coleta dos documentos que compuseram o córpus expandido. Estabelecemos o objetivo de construir um córpus anotado com cerca de 10.000 documentos balanceados entre as três classes e cerca de 100.000 documentos não anotados para serem usados na posterior etapa de expansão - sendo essas duas as saídas esperadas dessa etapa da pesquisa. Detalharemos essa etapa na [Seção 4.1](#).

Na etapa 2 da [Figura 14](#) temos a classificação de sentimentos. Para essa etapa, classificação e expansão, temos como entrada os documentos anotados e não anotados obtidos na etapa anterior. Ela está discutida na [Seção 4.2](#).

Outro desafio é encontrar *features* discriminativas que favoreçam aos métodos utilizados na classificação. Buscamos na literatura diversas representações como modelagens de *bag-of-*

words, PoS *tagging* e *embeddings* para melhor representar os dados durante a classificação.

A partir disso, buscamos investigar classificadores que consigam melhores valores de F-Measure na tarefa de classificação multiclasse até obter bons modelos treinados (usando como medida a *F-Measure*) para serem usados na classificação dos documentos não anotados. Detalharemos os processos na [Seção 4.2](#).

No final da etapa 2 teremos dados já classificados e divididos em dois grupos - os dados anotados com mais confiança e os anotados com menos confiança. Os primeiros são adicionados novamente ao conjunto de treino, enquanto os demais serão novamente classificados pelo modelo treinado com os dados anotados manualmente e com os dados mais confiáveis.

4.1 Extração de dados e anotação

Apesar dos *corpus* existentes para AS em Português, os recursos disponíveis no idioma ainda não satisfaziam as necessidades de nossos experimentos. Nosso objetivo de analisar 3 classes nos limitou ao uso de somente quatro *corpus* encontrados na literatura, o ReLi ([FREITAS et al., 2012](#)), o *corpus* da Copa das Confederações ([ALVES et al., 2014](#)), o *7x1* ([MORAES; MANSSOUR; SILVEIRA, 2015](#)) e o *Computer-BR* ([MORAES et al., 2016](#)).

O *corpus* da Copa das Confederações possui baixo número de documentos anotados (3.070), sendo que somente 15% desses foram anotados manualmente; as demais anotações foram obtidas por meio de supervisão distante usando *emojicons*. Buscávamos uma alternativa mais confiável, assim como mais documentos para o treinamento do modelo inicial de classificação.

Os *corpus* *7x1* e *Computer-BR* apresentam a mesma deficiência que o anterior, com cerca de 2.000 documentos em cada. A incompatibilidade de domínio ainda faz com que os *corpus* não possam ser combinados sem que isso acarretasse ruídos no treinamento dos modelos de classificação. Uma combinação do *7x1* com o *corpus* da Copa das Confederações seria possível, mas ainda assim o diacronismo dos dados e a natureza dos *7x1* (composto majoritariamente de ironias) poderia enviesar os resultados.

O ReLi poderia representar uma alternativa interessante, porém possui o desbalanceamento de classes evidente, sendo que um *under-sampling* (retirada de documentos da classe majoritária) geraria um *corpus* final com menos de 2.000 documentos. O mesmo fenômeno ocorre com os *corpus* já citados.

Pelas razões apresentadas, optamos por anotar um novo *corpus* de sentimentos para o Português, o TweetSentBR (TTsBR). A [Figura 15](#) ilustra as etapas de construção do *corpus*, iniciando pela extração de *tweets* e resultando em duas saídas - o TTsBR e o conjunto de dados não anotados que será utilizado nas posteriores etapas do projeto.

Nas subseções seguintes discutiremos cada etapa apresentada na figura.

Figura 15 – Fluxograma da primeira etapa do *framework*.

Fonte: Elaborada pelo autor.

4.1.1 Extração de Tweets

O primeiro passo da anotação foi a definição de um domínio para a coleta de dados. O *Twitter* é uma rede social onde usuários trocam mensagens, expressam emoções e se posicionam acerca de acontecimentos do cotidiano de maneira abrangente, por essa razão foi necessário determinar qual o escopo de informações a ser considerado em nossa pesquisa.

Nosso foco foi obter documentos categorizados como CGU (Conteúdo Gerado por Usuário), que é a denominação dada ao conteúdo criado de forma espontânea por indivíduos comuns, sem vínculos com meios de comunicação (BERTAGLIA, 2017). Os CGUs representam boa parte do conteúdo presente em ambiente virtuais e podem tratar sobre diversos assuntos, desde política até comentários pessoais.

Buscando opiniões e avaliações, nosso objetivo foi o de encontrar um domínio que incentivasse os usuários a se posicionarem positiva ou negativamente em relação a alguma entidade. Diversos trabalhos da literatura já exploraram diferentes domínios, como política (TUMASJAN *et al.*, 2011), produtos (HARTMANN *et al.*, 2014) e notícias (DOSCIATTI *et al.*, 2015). O uso de um contexto opinativo pode facilitar a representatividade das classes analisadas e diminuir a chance de uma classe majoritária.

Outro fator considerado foi a carência de recursos anotados na língua portuguesa, como já discutido no Capítulo 3 e na seção anterior. Devido ao alto custo de anotação, buscamos introduzir um cópus com domínio diferente dos já encontrados na literatura, visando ao aumento das possibilidades para pesquisas de PLN no Brasil. Um exemplo que ilustra essa necessidade seria o uso de eventos pontuais (como um torneio esportivo ou um festival musical), que poderiam gerar um grande número de documentos em um determinado período, mas talvez não uma boa quantidade de documentos para os experimentos de expansão de anotação.

Escolhemos o domínio de programas televisivos devido à periodicidade das exibições, que proporciona um bom volume de dados semanais, à abrangência do canal televisivo, que

aumenta a representatividade de fenômenos linguísticos presentes nos documentos recuperados, e aos assuntos discutidos, que geralmente despertam a necessidade opinativa dos usuários de redes sociais. Escolhemos alguns programas televisivos com boa repercussão e exibidos na grade da TV aberta.

O *Twitter* possui um sistema de agrupamento de mensagens pelo uso de *hashtags*, que são sequências de caracteres precedidos por # que indicam a temática de uma postagem. Usamos as *hashtags* para agrupar *tweets* (documentos ou postagens do *Twitter*) referentes aos programas escolhidos. As *hashtags* mais recorrentes na rede social em um determinado intervalo são exibidas em uma área chamada *trending topics*, o que faz com que emissoras e programas incentivem seus espectadores a utilizar as *hashtags* para aumentarem sua visibilidade na rede social, assim como fornecer aos usuários interações ao vivo nas atrações dos programas. Uma prática comum é a exibição das *hashtags* no canto superior das telas durante o programa, como pode ser visto na [Figura 16](#).

Figura 16 – Exibição do programa *Master Chef Brasil* na emissora Bandeirantes.

Fonte: Elaborada pelo autor.

A extração dos *tweets* foi feita por meio de um *crawler* construído com o uso da API do *Twitter* para *python*¹. A API permite que buscas sejam feitas na rede social, assim como alguns métodos de tratamento de *tweets* (como encontrar origem, número de *retweets*, verificar usuários). Existe uma limitação dessa ferramenta para a busca de *tweets* de um período superior a 7 dias, portanto o *crawler* foi construído para funcionar de maneira contínua. A posterior disponibilização do *corpus* foi feita por meio do *id* dos *tweets*, o que aumenta a capacidade da API recuperar *tweets* mais antigos. Ao todo foram extraídos 132.097 *tweets*. A distribuição de *tweets* por programas pode ser vista na [Tabela 9](#)

Foram escolhidos somente programas televisivos que divulgassem suas *hashtags* por meio de um canal oficial, seja por uma página própria ou durante a programação (exibindo na tela). Durante a extração, alguns programas fizeram uso de *hashtags* dinâmicas para determinados convidados. Um exemplo foi o programa Encontro com Fátima Bernardes, que possui em sua

¹ <<https://github.com/bear/python-twitter>>

Tabela 9 – Programas de televisão presentes na extração e suas ocorrências no corpus.

Programa	Hashtag Associada	Emissora	Quantidade	Porcentagem
Encontro com F. B.	#encontro	Rede Globo	16.430	12.43%
Vídeo Show	#videoshowaovivo	Rede Globo	18.844	14.26%
Altas Horas	#altashoras	Rede Globo	14.013	10.60%
Conversa com P. B.	#conversacombial	Rede Globo	10.345	7.83%
Mais Você	#maisvoce	Rede Globo	9.123	6.90%
É de Casa	#édecasa	Rede Globo	3.513	2.65%
Popstar	#palcopospstar	Rede Globo	1.741	1.31%
Domingo Legal	#domingolegal	SBT	6.936	5.25%
The Noite com D. G.	#thenoite	SBT	9.227	6.98%
Master Chef	#masterchefBR	Rede Bandeirantes	41.925	31.73%

Fonte: Elaborada pelo autor.

conta oficial a *hashtag* “#encontro”, mas na ocorrência de um convidado *Fulano*, também veicula “#encontroComFulano”. Esses casos foram ignorados em nossa pesquisa, utilizamos somente as *hashtags* contidas na Tabela 9 na formação do corpus.

O uso das *hashtags* ainda favorece a eliminação de ruídos, visto que elas geralmente agrupam *tweets* relacionados ao programa. Poucos casos foram identificados onde os documentos não tratavam do programa e usavam sua *hashtag*.

Para efetuar algumas garantias no corpus - como eliminar conteúdos repetidos -, filtramos postagens com textos iguais já presentes no corpus e *retweets*, que são compartilhamentos de textos criados por outros usuários; para obtermos somente CGUs, utilizamos um método da API que identifica quando usuários são “verificados”, ou seja, representam uma entidade (programa, emissora ou instituições) ou pessoas públicas (celebridades em geral), e os eliminamos do corpus.

4.1.2 Anotação Manual

A primeira parte do processo de anotação é a definição das teorias linguísticas que norteiam a anotação do corpus. Em nosso caso, precisamos definir quais as diferenças entre as sentenças positivas e negativas, assim como estabelecer o que deveria ser marcado como neutro. Um manual de anotação foi desenvolvido e testado nessa primeira etapa, porém ainda seria modificado durante as próximas etapas. Sete voluntários realizaram a anotação, sendo que suas áreas de atuação variavam entre linguísticas, jornalistas e acadêmicos de ciência da computação.

Adicionamos um campo opcional de incerteza. Durante a anotação, cada anotador poderia marcar o *check box* “Não tenho certeza” em algum determinado documento. A orientação foi de

que, em caso de dificuldade na anotação, os anotadores elegeassem uma classe mais representante do sentimento do *tweet* e marcassem a *check box*. Essa informação é útil para a análise dos erros dos nossos classificadores, assim como pode vir a ser útil para trabalhos futuros na mesma área ou em áreas afins.

Após a escrita de um manual de anotação, os anotadores receberam 100 *tweets* para uma anotação preliminar. Nessa anotação todos deveriam etiquetar os *tweets* respeitando o manual, sem que uma reunião presencial fosse feita. Nesse primeiro experimento, houve anotação unânime (todos os anotadores marcaram a mesma etiqueta para *tweets*) em 48% dos documentos. Em 23% dos documentos, somente um anotador divergiu da anotação dos outros e somente em 29% dos casos houve discordância entre mais de dois anotadores.

Quanto aos anotadores que marcaram “Não tenho certeza”, a média ficou em 4,66%, com desvio padrão de 4,7%. Ao todo, 24% dos documentos foram considerados “difíceis” por pelo menos um anotador e somente 4% foram considerados difíceis por mais de um anotador.

Para medir a concordância, usamos o *Krippendorff's Alpha* nominal e intervalar. A principal diferença entre eles é que o segundo penaliza menos anotações que divergem em um grau na escala, por exemplo, quando os anotadores assinalam *neutro* e *negativo*, em vez de *negativo* e *positivo*. A métrica nominal resultou em 63.70, enquanto a intervalar ficou em 80.80.

Essa etapa também serviu para calcularmos o tempo médio de anotação. A média foi de 12 minutos para cada 100 *tweets*, ou um *tweet* a cada 7,2 segundos. Esse tempo indica a facilidade que anotadores humanos encontram para alinhar um documento em uma das três classes.

Com base nas anotações, realizamos uma reunião presencial para treinar os anotadores. Melhoramos a teoria, reescrevendo o manual com base nas dúvidas e problemas discutidos em reunião e iniciamos o processo de anotação de concordância. Para essa etapa foram separados 300 *tweets* para todos os anotadores. Essa nova versão já compreendia exemplos melhorados e condizentes com as dificuldades relatadas pelos anotadores durante a reunião.

O *Krippendorff's alpha* (KRIPPENDORFF, 2004) nominal nessas etapa resultou em 52,90%, enquanto a métrica intervalar ficou em 70%. O kappa (FLEISS; COHEN, 1973) obtido foi 52,84%. Os valores reduziram em comparação com os obtidos na anotação preliminar. Acreditamos que isso se deve ao aumento nos dados.

O valor obtido na métrica nominal está de acordo com o obtido em outros trabalhos da área, como o corpus 7x1 (MORAES; MANSSOUR; SILVEIRA, 2015) que apresentou 53% de concordância. Entretanto, ficou abaixo do corpus *Computer-BR*, com 69%. Ambos os trabalhos usaram três anotadores para a tarefa, metade do que foi utilizado na nossa pesquisa, e isso pode ter impactado diretamente no baixo valor de concordância.

As últimas alterações foram feitas no manual e ele foi disponibilizado para todos os anotadores. Uma versão completa do manual pode ser conferida no [Apêndice A](#).

Para a etapa de anotação individual, cada anotador recebeu 2.333 *tweets* divididos em 5 etapas com periodicidade sugerida de uma semana para cada fase. A sugestão de manter-se a média de 100 *tweets* por dia foi feita visando a diminuição da carga para cada anotador e consequentemente da chance de erros por exaustão ou pressa na anotação. Um dos anotadores não terminou sua anotação, abandonando o projeto antecipadamente.

Um ambiente *web* específico para a anotação foi desenvolvido e pode ser conferido no [Apêndice B](#).

Ao todo foram anotados 15.047 *tweets* nesta primeira etapa, sendo 300 destes anotados por todos os anotadores (os 100 etiquetas da fase anterior à reunião presencial foram descartadas). Os *tweets* não anotados por mais de um anotador foram anotados por um único anotador.

Dentre os *tweets* anotados na fase de concordância, 17 resultaram em empate, ou seja, a votação majoritária resultou em número de votos igual para duas classes diferentes (positiva e negativa, neutra e positiva ou neutra e negativa). Optamos por removê-los dos experimentos, porém eles estão disponíveis na versão final do TTsBR devidamente marcados como empate. Ocorreram 2 empates entre as classes positiva e negativa, 7 empates entre a positiva e neutra e 8 empates entre neutra e negativa. Curiosamente, nesses casos de empate nenhum anotador marcou a opção “*não tenho certeza*”. Os *tweets* empatados estão listados no [Apêndice C](#).

4.1.3 Revisão da anotação

Para a avaliação dos modelos de classificação e extensão semissupervisionada, precisamos separar uma parcela do *corpus* anotado. Esse trecho será tratado nessa seção como *corpus de teste*.

O *corpus* de teste é uma parcela de aproximadamente 10% do *corpus* completo onde os modelos serão avaliados quanto a sua eficiência, portanto realizamos uma revisão das anotações desse trecho com dois supervisores. Um grupo de documentos já anotados foi selecionado aleatoriamente e anotado novamente por dois anotadores especialistas. Durante essa anotação, os supervisores tiveram acesso a etiqueta já agregada a cada documento, na forma da cor de fundo do documento: verde, se positivo; vermelho, se negativo; cinza, se neutro, como pode ser visto na [Figura 17](#).

Os supervisores poderiam modificar ou manter a etiqueta previamente estabelecida pelo anotador. Os documentos utilizados nessa etapa foram sorteados da anotação feita por um dos outros quatro anotadores, porém ambos os supervisores receberam os mesmos documentos. Dessa maneira, os documentos do conjunto de teste possuem três anotações, referentes ao anotador e aos dois supervisores.

Novamente, alguns *tweets* obtiveram empate nas anotações (30 documentos). Esses *tweets* estão listados no [Apêndice C](#). Em 16 casos, a opção *Não tenho certeza* não foi marcada por nenhum anotador. Podemos observar que no empate em três anotadores, cada um etiquetou

Figura 17 – Ambiente de anotação para a revisão do corpus de teste.

The interface displays five tweets for sentiment annotation:

- Tweet 1 (Green box):** "Parabéns a amiga da Sheron Menezes que fez o enxoval pra ela em NY. Amigos. Estou sem tempo é preciso de brusinhas. #encontro". Buttons: Negativo, Neutro, Positivo. Checkbox: Não tenho certeza.
- Tweet 2 (Pink box):** "@DaniloGentili citou o referendo de 2005 AAAAAAAAAAAAA e ainda citou o luladrão KKKK #TheNoite". Buttons: Negativo, Neutro, Positivo. Checkbox: Não tenho certeza.
- Tweet 3 (Grey box):** "A sardinha que ganhar vai ser meu prato pós festa de amanhã #MasterChefBR". Buttons: Negativo, Neutro, Positivo. Checkbox: Não tenho certeza.
- Tweet 4 (Light Green box):** "Maluma é muito talentoso né #TheNoite".

Dicas:

1. Todos os tweets devem ser anotados.
2. **Classe positiva:** O tweet reflete uma intenção do autor em mostrar um sentimento positivo?
3. **Classe negativa:** O tweet reflete uma intenção do autor em mostrar um sentimento negativo?
4. **Classe neutra?** O tweet não reflete nem sentimento positivo, nem sentimento negativo do autor.
5. Caso se sinta em dúvida, marque a caixa "Não tenho certeza"
6. **Programas avaliados:**
 - #encontro #adnight #esquenta
 - #thevoicekids #DomingoLegal
 - #altasHoras #TheNoite
 - #videoShowAoVivo #maisvoce
 - #masterchefbr #édecasa
 - #masterchefBR #HellsKitchenBR
 - #thevoicebrasil #XFactorBR
 - #ConversaComBial

[Ver manual completo](#)

Fonte: Elaborada pelo autor.

com uma categoria cada um dos *tweets*. Abaixo temos alguns exemplos desse grupo:

"higiene mental é a palavra-chave"

"fé no pai que hj a miriam sai"

"mds tenho escola de manhã e ainda to aki tbm quem manda ser tão emocionante"

Nos casos acima podemos observar que somente as informações dos *tweets* não são suficientes para se identificar inequivocamente qual o alinhamento semântico da intenção do autor. Isso ilustra um pouco do desafio que o domínio proporciona na classificação de polaridades em três classes.

Os *tweets* da etapa de concordância também foram utilizados na composição do conjunto de treino, visto que cada um dos 300 *tweets* possui seis anotações, o que aumenta a confiança da polaridade escolhida.

4.1.4 O TweetSentBR

O corpus final totalizou 15.000 documentos, sendo 47 descartados por conta do empate na anotação.

O corpus de treinamento possui 12.990 *tweets*, enquanto o de teste soma 2.010. Para definir as etiquetas dos documentos do conjunto de treinamento usamos votação majoritária entre as múltiplas anotações. A [Tabela 10](#) apresenta a distribuição de classes para cada conjunto e para cada classe.

Tabela 10 – Dados do corpus compilado.

Cópus	Positivos	Neutros	Negativos	Total
Treino	5.741 (44,2%)	3.410 (26,3%)	3.839 (29,5%)	12.990
Teste	907 (45,1%)	516 (25,7%)	587 (29,2%)	2.010
Total	6.648 (44,3%)	3.926 (26,1%)	4.426 (29,6%)	15.000

Fonte: Elaborada pelo autor.

Observamos na [Tabela 10](#) um desbalanceamento entre as classes, algo que propicia ruído em classificadores baseados em AM ([BRUM; ARAUJO; KEPLER, 2016](#); [MONARD; BATISTA, 2002](#)), porém, usando a técnica de *under-sampling* para reduzir o número de documentos da classe majoritária, ainda teremos 10.230 documentos no conjunto de treino, o que satisfaz nosso objetivo inicial.

Os *tweets* anotados foram escolhidos aleatoriamente dentre os obtidos na extração inicial; na [Tabela 11](#) podemos ver a distribuição de *hashtags* de busca no TTsBR.

Nenhum *tweet* sobre o programa “Popstar” está presente no corpus de teste, porém existem documentos sobre o programa no corpus de treinamento.

Dos 15.000 *tweets* do corpus, 12.990 (86,6%) possuem anotação individual, enquanto 1.727 (11,5%) possuem anotação tripla e 283 (1,9%) possuem sete etiquetas. Decidimos disponibilizar no corpus todas as anotações, assim como a quantidade de usuários que marcaram a opção “Não tenho certeza” e a etiqueta final obtida por votação majoritária.

Realizamos uma avaliação simples das palavras contidas no TTsBR usando TF-idf para medir quais termos são mais relevantes para as classes mais polares do corpus. Os cinco termos mais relevantes para as classes estão apresentados na [Tabela 12](#).

Podemos observar a presença de palavras positivas (“amo”, “fofura”) na classe positiva, assim como adjetivos pejorativos na classe negativa. Um fenômeno interessante é a presença do termo “tirem”, característico de *reality shows* onde os usuários pedem que um participante seja retirado do programa por não simpatizarem com o indivíduo.

O ranqueamento feito com a classe neutra resultou em termos como “consolidados”, que

Tabela 11 – Distribuição de *hashtags* no TTsBR.

Programa	Hashtag Associada	Quantidade	Porcentagem
Encontro com F. B.	#encontro	2.166	14.44%
Vídeo Show	#videoshowaovivo	2.006	13.37%
Altas Horas	#altashoras	1.119	7.46%
Conversa com P. B.	#conversacombial	1.314	8.76%
Mais Você	#maisvoce	1.184	7.89%
É de Casa	#édecasa	259	1.72%
Popstar	#palcopopstar	0	0.00%
Domingo Legal	#domingolegal	293	1.95%
The Noite com D. G.	#thenoite	1.252	8.34%
Master Chef	#masterchefBR	5.407	36.04%

Fonte: Elaborada pelo autor.

representa alguns *tweets* indicando programas com grande audiência gerados automaticamente, nomes de emissoras (“redeTV”, “globo”) e verbos genéricos (“apresentado”, “marcaram”).

Tabela 12 – Termos mais relevantes para cada classe no TTsBR usando TF-idf.

#	Classe Positiva	Classe Negativa
1	amo	ridículo
2	fofura	péssimo
3	adorando	lixo
4	emocionada	tirem
5	linda	mala

Fonte: Elaborada pelo autor.

A lista de termos positivos ainda compreendia *emojis* positivos (como carinhas felizes, corações e indicativos de “top”), enquanto na classe negativa foram encontradas carinhas zangadas e desinteressadas, termos de deboche (“mimimi”), palavrões e termos irônicos como “aham” (que indica concordância, mas nesse caso dá tom sarcástico aos *tweets*).

Uma das preocupações com recursos linguísticos como *corpus* obtidos em redes sociais é com a política de privacidade das redes. O *Twitter* proíbe a reprodução de textos oriundos de usuários da rede social, portanto disponibilizamos somente os *ids* dos *tweets*, junto com uma ferramenta de *crawling* baseada na API do *Twitter*, que permite a extração dos dados. Dessa maneira é possível disponibilizar a base anotada para que outros pesquisadores possam utilizá-la legalmente. A seguir temos um trecho do TTsBR disponibilizado:

id	hashtag	labels	hard	sent	group
863044774588272640	#encontro	[1,1,1,1,1,1,1]	0	1	test
865217606143418368	#encontro	[-1]	0	-1	train
864667631642705920	#masterchefbr	[-1,-1,0]	2	-1	test
865063232201011201	#TheNoite	[1,0,1,1,1,0,0]	2	1	test
864668391008763905	#masterchefbr	[0,0,0,0,0,0,0]	0	0	test
864647552087543808	#masterchefbr	[0]	0	0	train
865582747703816192	#encontro	[1]	0	1	train
862169393844846593	#masterchefbr	[0,1,1,0,1,-1,0]	0	-	
864702535134851073	#TheNoite	[1,0,-1]	0	-	
862147013667934210	#masterchefbr	[1,-1,0]	0	-	

A primeira coluna apresenta o *id* do *tweet*, na segunda temos a *hashtag* de onde o *tweet* foi originado. A coluna seguinte traz o conjunto de etiquetas dos anotadores: 1, 3 ou 7 valores -1 (negativo); 0 (neutro) ou 1 (positivo). Em seguida, o número de usuários que assinalaram a opção “Não tenho certeza”, a etiqueta final do *tweet* e, por último, o grupo ao qual o documento pertence: teste, treino ou nenhum dos anteriores (para documentos empatados).

O TTsBR está disponível livremente na internet² para replicação e uso em AS e outras tarefas de PLN.

Além do TTsBR, durante a extração de *tweets* ainda coletamos 117.050 documentos sem etiquetas para serem utilizados na etapa de expansão de corpus, descrita na [Subseção 4.2.4](#).

4.2 Classificação e expansão

A segunda etapa do nosso trabalho compreendeu a investigação de classificadores de sentimentos. Nessa etapa utilizamos o corpus TTsBR para treinar e testar classificadores clássicos já utilizados em trabalhos da literatura para Português Brasileiro.

Nessa seção discutiremos a etapa de pré-processamento dos dados, a modelagem das representações utilizadas e os algoritmos implementados para a classificação de polaridade multiclasse.

4.2.1 Pré-processamento

A etapa de pré-processamento compreende o conjunto de modificações que são aplicadas aos dados antes de seu uso pelos métodos de classificação. Essa etapa foi repetida tanto para os dados anotados quanto para os não anotados.

² <https://bitbucket.org/HBrum/tweetsentbr/>. Acessado em 11 de Janeiro de 2018.

Alguns trabalhos exploraram o papel do pré-processamento e da normalização na AS, como [Avanço \(2015\)](#) e [Junior et al. \(2017\)](#). No primeiro, os testes efetuados com a ferramenta UGCNormal ([DURAN et al., 2015](#)) foram aplicados em um *cópus* de *reviews* de produtos com suporte linguístico de um site de *e-commerce*, onde os usuários se encontram em um ambiente favorável à troca de opiniões. Ainda assim, os autores concluíram que a normalização não representou diferença estatística significativa na classificação nos métodos de AM - somente houve diferença significativa no método de classificação com abordagem lexical.

Em [Junior et al. \(2017\)](#) os autores dispunham de uma base de dados muito semelhante à utilizada nesse trabalho, portanto optamos por utilizar o mesmo método de pré-processamento.

A ferramenta utilizada foi o Enelvo ([BERTAGLIA, 2017](#)), ferramenta livre disponível na internet ³ que oferece suporte para pré-processamento e normalização de dados textuais em Português.

A seguir descrevemos o pré-processamento dos dados nesse trabalho:

- **Tokenização:** o processo de *tokenização* é o agrupamento lexical dos dados, separando aglutinações e removendo o número de ruídos nos dados trabalhados. A linguagem natural faz com que *tokens* ou unidades lexicais sejam aglutinadas a pontuação ou marcadores, como em “*O menino bom estava correndo. Bom, e agora?*”, onde temos, entre outros, dois *tokens* diferentes sendo formados para o adjetivo “bom” - temos “bom” e “Bom,”. O processo de *tokenização* dividiria essa sentença em unidades lexicais, transformando-a em “O menino bom estava correndo . Bom , e agora ?”.

Ainda no exemplo anterior, dois *tokens* seriam formados para o mesmo adjetivo - “bom” e “Bom”. Para tal, convertemos todos os *tweets* em caixa baixa para que ambas palavras fossem mapeadas para a mesma unidade lexical.

- **Remoção de pontuação:** Além da *tokenização*, removemos a pontuação dos documentos. O próprio Twitter tem como característica a informalidade na escrita e a despreocupação com a norma culta. Assumimos que a ausência da pontuação não prejudicaria a classificação de sentimentos.

Uma exceção foi feita quando um elemento de pontuação fosse parte de um *emoticon* (conjunto de caracteres que representa uma figura). Por se tratarem de bons indicativos de polaridade ([GO; BHAYANI; HUANG, 2009](#)), mantivemos qualquer símbolo de pontuação que fosse parte integrante de um *emoticon*. Para tal utilizamos o dicionário de *emoticons* usado em [Junior et al. \(2017\)](#).

Emojis, caracteres especiais que também representam figuras, foram mantidos da mesma forma, já que também são fortes indícios de polaridade ([JUNIOR et al., 2017](#)).

³ <<https://github.com/tfcbertaglia/enelvo/>>. Acessado em 12 de Janeiro de 2018.

- **Substituições de termos:** Realizamos substituições de termos a fim de normalizar algumas expressões que pudessem sobrecarregar métodos de representação como o *bag-of-words*.

Números como datas e números de telefones foram substituídos pela etiqueta “*NUMBER*”. Outra substituição foi feita em links, transformando-os para a etiqueta “*URL*”.

Os nomes de usuários (representados no *Twitter* pros cadeias de caracteres iniciadas em “@”) foram substituídos pela etiqueta “*USERNAME*” para garantir a privacidade dos usuários e para eliminar ruídos dos documentos classificados. Em *reality shows*, por exemplo, muitas vezes um participante costuma ser o preferido do público, fazendo com que seu usuário seja constantemente incluído em *tweets* positivos e de apoio. Dessa maneira conseguimos reduzir um pouco o enviesamento causado por esse fenômeno.

Quanto ao nome próprio do usuário, seria necessária a aplicação de alguma ferramenta de REN (Reconhecimento de Entidade Nomeada) para tal finalidade, mas optamos por descartar essa possibilidade assumindo que o mesmo nome do participante hipotético seria usado em diversos contextos, diferente de seu usuário na rede social que é único.

As *hashtags* foram mantidas, pois podem carregar informação semântica (como em *#viva*, que denota felicidade, *#triste*, que pode indicar tristeza e *#sqn* que é um forte indicativo de ironia). A única exceção são as *hashtags* usadas na extração de dados, ou seja, as *hashtags* dos programas televisivos. Isso poderia enviesar nosso método se houvesse, por exemplo, um programa muito adorado pelo público o qual em todos os documentos a *hashtag* do programa estivesse presente. A presença da *hashtag* nesse exemplo seria entendida pelos métodos de AM como uma *feature* muito relevante para a classificação, o que seria completamente irrelevante em qualquer cenário envolvendo outros programas.

- **Repetição de caracteres:** outra característica comum das redes sociais é a repetição de caracteres causada pelo entusiasmo durante a escrita dos *tweets*. Isso faz com que sejam produzidas unidades lexicais como “nossa”, “nooossa” e “noooooooooossa”. Para reduzir esse fenômeno, mas manter a semântica agregada à repetição, definimos um limite de 4 repetições de caracteres e reduzimos todos os casos excedentes. Dessa forma, “nooossa” continuou existindo, porém “noooooooooossa” foi mapeado para “nooossa”.

Para ilustrar a etapa de pré-processamento, apresentamos o exemplo a seguir:

Original: “Adorei o programa!estou vendo pela internet em *www.globo.com*, vamos assistir juntos @suelen @mtreviso ? por favooooooooor!! :D #tmj #encontro”

Pré-processada: “adorei o programa estou vendo pela internet em *URL* vamos assistir juntos *USERNAME USERNAME* por favooooor :D #tmj”

4.2.2 Modelagem dos Dados

Apesar do pré-processamento, classificadores de AM ainda têm muita dificuldade no processamento de texto. Para que os textos sejam compreendidos com mais clareza é necessário que se estabeleça uma representação vetorial do objeto de análise - no nosso caso, documentos para a classificação de polaridades.

Usamos como base os métodos mais clássicos da área para representar os dados. A seguir descrevemos como cada abordagem foi implementada e utilizada.

- **Bag-of-words:** O *bag-of-words* é uma das representações mais intuitivas na AS, já sendo utilizada desde o primeiro trabalho que usou AM na área (PANG; LEE; VAITHYANATHAN, 2002). Ele consiste de um vetor esparsos com n posições onde n é o tamanho do vocabulário.

Cada documento é representado como um vetor de tamanho n onde um valor é adicionado em cada posição cuja palavra esteja presente no texto. As variações de *bag-of-words* podem se dar pela utilização de um vetor binário (MORAES *et al.*, 2016), onde só é marcado com 1 as posições de palavras presentes; um vetor de frequências (AVANÇO; BRUM; NUNES, 2016); ou um vetor com o peso de cada palavra definido por TF-IDF (term frequency-inverse document frequency), usado em Junior *et al.* (2017). Outra variação é o uso de n-gramas para uma análise mais profunda das composições semânticas nos documentos (PANG; LEE; VAITHYANATHAN, 2002).

Um dos maiores desafios do *bag-of-words* é seu alto custo de memória, visto que o tamanho do vocabulário impacta diretamente no armazenamento de cada vetor dos conjuntos de treino e teste. Em nosso caso essa situação se ampliou pelo grande número de dados sendo classificados.

Para contornar a situação, usamos somente o modelo de *bag-of-words* por palavra, ou seja, sem o uso de modelo n-grama. Utilizamos também a contagem de ocorrência no vetor, semelhante a o que é feito em Avanço, Brum e Nunes (2016).

Além disso, utilizamos uma estrutura de *Hash* para armazenar os dados e reduzir o tamanho máximo do *bag-of-words* para 5.000 dimensões⁴. Essa redução faz com que alguns elementos sejam agrupados aleatoriamente na mesma posição de memória durante a construção do *bag-of-words*.

- **Palavras de negação:** Uma representação utilizada em Avanço (2015) foi a de um sinalizador de palavras de negação. Essas palavras podem indicar a mudança de polaridade em sentenças, como por exemplo, *esse filme não é bom*.

⁴ <http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.HashingVectorizer.html>. Acessado em 15 de Janeiro de 2018.

Para reconhecer essas palavras usamos o mesmo conjunto lexical apresentado por [Avanço \(2015\)](#), composto por “jamais”, “nada”, “nem”, “nenhum”, “ninguém”, “ninguem”, “nunca”, “não”, “nao” e “tampouco”.

Mais uma dimensão é adicionada ao vetor de representação com a contagem de quantas negações estão representadas no documento. É importante ressaltar que mesmo com o *bag-of-words*, essa dimensão a mais dá informação morfossintática aos classificadores, pois mesmo com a representação de quantos “não” ou “jamais”, o agrupamento de léxicos de negação indica ao classificador que existe uma relação entre esse grupo.

- **Emoticons:** Como citado na [Subseção 4.2.1](#), os *emoticons* são indicativos fortes de polaridade em documentos do Twitter. Isso pode ser observado em trabalhos como [Go, Bhayani e Huang \(2009\)](#) e [Pak e Paroubek \(2010\)](#), onde córpis foram anotados somente usando informações de *emoticons* por Supervisão Distante.

Para identificar esses *emoticons*, usamos o mesmo conjunto usado em [Avanço \(2015\)](#). Os *emoticons* positivos e negativos podem ser vistos na [Figura 18](#).

Figura 18 – Emoticons positivos e negativos usados na representação dos documentos.

:-) :) :o) :] :3 :c) :> =] 8) =) :} :^) :-)) ;-) :'-) :') \o/ \0/ :-D :D 8-D 8D x-D xD X-D XD =-D =D =-3 =3 B^D <3 ;-) *-) *) ;-] ;] ;D ;^) :-	>:\ \ >:/ :-/ :- . :/ :\ \ =/ =\ \ :L =L :S >.< :- <:- >:[:- (: (:-c :c :-< :< :-[:[:{ :- :@ :'- (:' D:< D: D8 D; D= DX v.v D-' : (>_<) :
Emoticons Positivos	Emoticons Negativos

Fonte: Elaborada pelo autor.

A representação de *emoticons* consistiu em duas dimensões (uma para positivos e outra para negativos) concatenada ao vetor de representação com a ocorrência de *emoticons* de cada classe no documento.

Essa *feature* pode ser muito importante na classificação de classes polares, porém não fornece tanta informação para a classe neutra.

- **Emojis:** Semelhante ao que foi feito com os *emoticons*, usamos o mesmo argumento para o uso de *emojis*, que diferem dos primeiros por serem caracteres especiais em vez de conjuntos de caracteres.

Um trabalho que ilustra a importância dos *emojis* é o [Junior et al. \(2017\)](#), onde a mesma abordagem proposta por [Go, Bhayani e Huang \(2009\)](#) é usada com a adição de *emojis* como indicadores de polaridade para a anotação de um córpis.

Para identificar os *emojis* utilizamos a base de dados *Emoji Sentiment Ranking* (NOVAK *et al.*, 2015). Essa base foi construída com base em 1.6 milhões de *tweets* etiquetados por 83 anotadores em 13 idiomas europeus (incluindo o Português).

O léxico de sentimentos apresenta a distribuição de *emojis* em documentos Positivos, Neutros e Negativos, o que nos proporcionou uma pontuação para cada *emoji* encontrado.

A Figura 19 mostra uma tabela com exemplos da anotação do *Emoji Sentiment Ranking*: na primeira coluna temos o *emoji*, na segunda o número de *tweets* onde ele foi encontrado; na próxima a posição média do *emoji* em um *tweets* (variando de 0 até 1, respectivamente do início ao final do documento), nas três colunas centrais temos a pontuação do *emoji* nas três classes, em seguida a pontuação final do *emoji* (variando de -1 a 1) e por último uma breve definição do mesmo.

Figura 19 – Exemplo de anotação no *Emoji Sentiment Ranking*.

Emoji	N	Position	p_-	p_0	p_+	$\bar{s}$	Name
😭	14,622	0.80	0.25	0.29	0.47	0.22	face with tears of joy
❤️	8,050	0.74	0.04	0.17	0.79	0.75	heavy black heart
♥️	7,144	0.75	0.04	0.27	0.69	0.66	black heart suit
😊	6,359	0.76	0.05	0.22	0.73	0.68	smiling face with heart-shaped eyes
😭	5,526	0.80	0.44	0.22	0.34	-0.09	loudly crying face
😘	3,648	0.85	0.05	0.19	0.75	0.70	face throwing a kiss
😊	3,186	0.81	0.06	0.24	0.70	0.64	smiling face with smiling eyes
👌	2,925	0.80	0.09	0.25	0.66	0.56	ok hand sign
❤️❤️	2,400	0.76	0.04	0.29	0.67	0.63	two hearts
👏	2,336	0.78	0.10	0.27	0.62	0.52	clapping hands sign

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

Usamos somente as ocorrências, concatenando-as ao vetor de representação do documento.

- **Léxico de sentimentos:** Inspirados em Avanço, Brum e Nunes (2016), utilizamos também informação de léxicos de sentimentos para a representação dos documentos. Usamos o Sentilex (SILVA; CARVALHO; SARMENTO, 2012) na contagem de palavras etiquetadas no léxico como positivas e negativas.

Como nos itens acima, concatenamos essa informação à representação do documento.

- **Etiquetas de *Part-of-Speech*:** Uma informação que pode auxiliar os classificadores de sentimentos são as etiquetas morfossintáticas, também chamadas de *part-of-speech tags* ou *PoS tags*.

Essa informação adiciona sintaxe para a classificação, assim como foi feito em Avanço, Brum e Nunes (2016), porém explorando uma característica mais relevante para o foco do nosso trabalho. Utilizamos as ocorrências de Adjetivos, Substantivos, Advérbios e Verbos nos documentos com o objetivo de facilitar a classificação da classe neutra.

Assumimos que a contagem dessas etiquetas (em especial aos Adjetivos) pode representar as características significativas na identificação de sentenças neutras, pois observa-se empiricamente número baixo de adjetivos em alguns grupos de sentenças dessa classe.

Para extrair as etiquetas utilizamos o *tagger* da NLPnet (FONSECA; ROSA; ALUÍSIO, 2015), ferramenta de PLN que contém diversas implementações para etiquetas morfo-sintáticas e *parsers* ⁵.

- **Vetores de Representação Densos (*word embeddings*):** Nos últimos anos, modelos de espaço vetorial vêm se destacando como boas alternativas de representações de palavras. Os modelos já existiam há muito tempo (BENGIO *et al.*, 2003), mas ganharam notoriedade após o trabalho de Mikolov *et al.* (2013), onde os autores apresentaram indícios de que relações complexas entre palavras podiam ser capturadas usando o word2vec, modelo neural que gera uma representação densa de palavras.

Métodos de *Deep Learning* geralmente fazem uso dessas representações (JUNIOR *et al.*, 2017; TREVISIO; SHULBY; ALUÍSIO, 2017) para inicializar redes neurais multi-camadas e diversas tarefas de PLN como normalização (BERTAGLIA, 2017) e *parsing* (FONSECA; ALUISIO *et al.*, 2015).

Essas representações são geradas a partir da tarefa de previsão de uma palavra em um contexto (chamado modelo *c-bow*) ou na previsão de um contexto, dado uma palavra (modelo chamado *skip-gram*), ambos introduzidos em Mikolov *et al.* (2013).

Usando os pesos aprendidos pelo modelo neural das duas tarefas, é possível gerar uma representação densa que pode ser capaz de aprender relações sintáticas e semânticas entre palavras de um vocabulário. Dessa maneira decidimos usar esse modelo para gerar representações para os nossos documentos. Essas representações são conhecidas como *word embeddings*.

As *word embeddings* possuem uma grande densidade nos dados e suas *features* não são discriminativas, ou seja, não se sabe exatamente o que cada dimensão representa para cada palavra. Por essa razão elas não são utilizadas frequentemente em modelos como SVM ou Naive Bayes, onde se é possível analisar o espaço para melhorar a classificação. Por essa razão reduzimos nossos experimentos a somente dois modelos.

Os dois modelos utilizados foram os mesmos aplicados aos métodos de *Deep Learning* apresentados em Junior *et al.* (2017). Ambos modelos foram treinados usando um conjunto de 14 milhões de *tweets* em Português extraídos sem filtragem de termo.

Um dos modelos foi treinado com o método *skip-gram* e possui 50 dimensões, enquanto o outro foi treinado usando *c-bow* e possui 600 dimensões. Ambos os modelos foram os que obtiveram os melhores resultados em (JUNIOR *et al.*, 2017) e essa foi nossa motivação para experimentá-los em nosso *framework*.

⁵ <<http://nilc.icmc.usp.br/nlpnet/>>. Acessado em 16 de Janeiro de 2018.

A importância das representações está na maneira como o espaço analisado será formado. Podemos observar que o elevado número de trabalhos na literatura compreende um grande número de abordagens para se representar as classes positiva e negativa (*emoticons* e uso do léxico de sentimentos). Entretanto o foco do trabalho é utilizar também a classe neutra, algo que ainda é pouco explorado na literatura e ainda são poucas as representações que dão ênfase nesse desafio.

4.2.3 Métodos de Classificação

A partir das representações dos dados, utilizamos diferentes métodos de AM para classificá-los nas três classes de opiniões. Nessa seção apresentaremos os métodos utilizados e suas características.

Todos os classificadores apresentados são supervisionados e discutiremos mais sobre hiperparâmetros e sobre o processo de classificação no [Capítulo 5](#). A implementação dos métodos utilizados foi a da biblioteca scikit-learn ([PEDREGOSA et al., 2011](#)) disponível gratuitamente na internet ⁶.

- **Support Vector Machines:** O SVM é um algoritmo supervisionado muito utilizado em tarefas de classificação e regressão. O método consiste na utilização de vetores de suporte para calcular um hiperplano definido por uma função que divide o espaço em dois subespaços ([VAPNIK; GOLOWICH; SMOLA, 1997](#)). Na análise de sentimentos o método foi utilizado desde os trabalhos mais clássicos como [Pang, Lee e Vaithyanathan \(2002\)](#), até trabalhos mais recentes para o inglês ([SOCHER et al., 2013](#); [SILVA et al., 2016](#)) e português ([MORAES et al., 2016](#); [AVANÇO; BRUM; NUNES, 2016](#)).

Apesar de ser um classificador binário, para as classificações multiclasse utilizamos a abordagem *one-vs-all* ([ROBERT, 2014](#)). Nesta abordagem nós treinamos um número de classificadores iguais ao número de classes, tratando a classe analisada como *positiva*, enquanto os demais documentos são considerados *negativos*. A classe final obtida para cada documento será definida por $y(x) = \operatorname{argmax}_c f_c(x)$.

Um dos pontos principais do classificador é a maneira como definimos o hiperplano de corte. O SVM aceita corte linear ou polinomial, sendo que o segundo consegue se adequar a espaços mais complexos dependendo do grau utilizado. Optamos por realizar experimentos somente com SVM de *kernel* linear, visto que a utilização de um *kernel* polinomial aumenta consideravelmente o tempo de treinamento dos modelos. Essa preocupação com o tempo de execução é relevante, visto que nossa abordagem prevê iterações que realizam constantes retreinamento dos modelos de classificação.

- **Naive Bayes:** O classificador Naive Bayes utiliza o Teorema de Bayes para prever o acontecimento de uma classe, dado um conjunto de *features* numéricas representadas

⁶ <http://scikit-learn.org/>. Acessado em 15 de Janeiro de 2018.

em um vetor. O método, apesar de simples, é muito utilizado na análise de sentimentos ([AVANÇO; BRUM; NUNES, 2016](#); [BRUM; ARAUJO; KEPLER, 2016](#)).

Apesar da condição de independência entre atributos e classes, [Domingos e Pazzani \(1997\)](#) aponta que, mesmo em espaços onde existam claras dependências, o método funciona de maneira satisfatória. Além disso, o classificador Naive Bayes possui como característica a baixa incidência de *overfitting* ([ROBERT, 2014](#)).

A implementação que utilizamos para a nossa classificação foi a mesma usada por [Avanço \(2015\)](#), chamada de Bernoulli Naive Bayes, onde a distribuição de Bernoulli é utilizada no cálculo das probabilidades das classes.

- **Regressão Logística:** A Regressão Logística pode ser entendida como uma generalização de modelos de regressão linear para classificação binária onde uma função sigmoid extrai as probabilidades de cada classe ([ROBERT, 2014](#)).

Em [Junior et al. \(2017\)](#) os autores utilizam a Regressão Logística na classificação de polaridades e conseguem obter resultados que superam os atingidos com SVM e com arquiteturas de *Deep Learning*.

O classificador é muito simples e possui agilidade no treinamento de novos modelos, o que favorece nosso *framework*.

- **Multilayer Perceptron:** A *Multilayer Perceptron* (MLP) é uma arquitetura de rede neural artificial composta de três ou mais camadas, sendo uma camada de entrada com os dados de entrada, uma camada oculta contendo neurônios com uma função de ativação definida e uma camada de saída ([ROSENBLATT, 1961](#)).

Cada elemento da camada de entrada é multiplicado por um peso e submetido a uma função de ativação em um neurônio da camada escondida. O processo de treinamento do modelo busca otimizar esses pesos para que a saída produzida na camada final seja a mais próxima possível da esperada pelo conjunto de treinamento. O algoritmo realiza diversas iterações, usando *back-propagation* ([RUMELHART; HINTON; WILLIAMS, 1985](#)) para atualizar os pesos da MLP.

Em nossos modelos usamos *relu* como a função de ativação. Discutiremos no [Capítulo 5](#) sobre o número de neurônios e camadas utilizados em nosso *framework*.

- **Árvore de Decisão:** O aprendizado por árvores de decisão se baseia na formação de uma árvore onde as folhas representam as classes, e regras são baseadas nas análises dos dados do conjunto de treinamento.

O modelo é muito intuitivo e é possível seguir os caminhos obtidos na classificação para identificar quais *features* foram mais impactantes para que uma classificação seja feita em determinada classe.

- **Random Forest:** *Random Forest* é uma abordagem de classificação proposta em Breiman (2001) onde diversos estimadores (que são Árvores de Decisão) são utilizados para classificar diferentes subconjuntos dos documentos que estão sendo classificados. O método pode ser visto como um comitê de Árvores de Decisão.

Um dos critérios mais importantes em nossa expansão é a medida de confiança que usaremos para ranquear os documentos classificados.

Para cada classificador, obtemos uma distribuição probabilística para cada classe analisada. Usamos essa distribuição como medida de confiança para cada documento.

4.2.4 Expansão de Córpus

Após a implementação e avaliação dos classificadores de polaridade, damos continuidade à etapa de classificação e expansão classificando documentos não anotados, medindo a confiança da classificação e adicionando uma parcela dos dados mais confiáveis ao conjunto de treinamento.

Acompanhando a Figura 20 conseguimos observar o processo como um ciclo iterativo onde o conjunto de treino aumenta a medida que novos dados são adicionado.

Figura 20 – Segunda etapa do *framework* de expansão de córpus.

Fonte: Elaborada pelo autor.

Implementamos um *framework* de aprendizado semissupervisionado baseado em *self-training*. Indicamos inicialmente os dados que serão usados como treinamento do modelo inicial de classificação e os dados que serão usados como expansão do treinamento inicial. Um *threshold* de adição é especificado, ele definirá o número de iterações que o *framework* realizará e o número de documentos adicionado ao córpus expandido em cada iteração.

Os *thresholds* devem ser porcentagens. Em nossos experimentos variamos os valores de 40%, 30%, 25%, 20%, 10%, 5% e 1%. Podemos observar que cada *threshold* já define o número

de iterações que serão feitas nos dados, por exemplo, usando o *threshold* 25%, serão feitas 4 iterações adicionando 1/4 do corpus não anotado em cada iteração.

Na primeira iteração, os dados são classificados com um dos classificadores apresentados na subseção anterior e medimos a probabilidade de cada classe predita pelo classificador. Ordenamos os documentos por maior valor de confiança e adicionamos a quantidade de documentos definida pelo valor de *treshold* de adição no corpus de treinamento.

Nosso *framework* está disponível livremente na internet ⁷, contendo todas as implementações realizadas em *python*.

No próximo capítulo apresentamos e discutimos os resultados dos experimentos realizados.

⁷ <<https://bitbucket.org/HBrum/tweetsentbr/>>

EXPERIMENTOS

Neste capítulo apresentamos os experimentos e resultados obtidos no trabalho. Dividimos o capítulo em duas seções, avaliando separadamente o TweetSentimentBR (TTsBR) e a expansão semissupervisionada do mesmo usando dados não anotados.

Na [Seção 5.1](#) discutimos as vantagens e desvantagens do TTsBR em relação aos demais corpú da literatura, usando a tarefa de classificação de polaridades como comparativo.

Em seguida exploramos a expansão semissupervisionada, na [Seção 5.2](#), iniciando com experimentos com diferentes configurações de representação (*features*) e, depois, com as avaliações dos métodos de expansão.

5.1 Experimentos com o TTsBR

O TTsBR foi construído com o objetivo de nos proporcionar material de estudo de abordagens semissupervisionadas na expansão de dados e também para servir de recurso científico para a área de AS no Português Brasileiro.

A [Tabela 13](#) apresenta uma visão geral dos corpú disponíveis na literatura com destaque para o TTsBR. Podemos observar o domínio do corpú, a quantidade de documentos por classe e se o corpú é balanceado, seu tamanho total e a maneira como os dados foram etiquetados. Os corpú da Copa das Confederações ([ALVES et al., 2014](#)), 7x1 ([MORAES; MANSSOUR; SILVEIRA, 2015](#)) e Computer-BR ([MORAES et al., 2016](#)) estão indicados com asterisco pois não conseguimos acesso aos mesmos, por essa razão eles não serão utilizados no decorrer dos experimentos.

Recursos para a classificação em três polaridades ainda são escassos no PB, somente quatro corpú possuem anotação para a classe neutra, sendo que todas essas anotações foram manuais (o corpú da Copa das Confederações possui anotação automática somente para as classes positiva e negativa). O ReLi é o maior corpú que abrange essa categoria, porém conta

Tabela 13 – Visão geral dos corpú de SA para Português Brasileiro em comparação ao TTsBR.

Córpus	Domínio	# pos	# neu	# neg	Balanceado	# Documentos	Anotação
Buscapé-1	<i>Reviews</i> de produtos	1.000	-	1.000	sim	2.000	Automático
Buscapé-2	<i>Reviews</i> de produtos	6.873	-	6.812	sim	13.685	Automático
Mercado Livre	<i>Reviews</i> de produtos	21.820	-	21.499	sim	43.818	Automático
Pelesent	Genérico (<i>tweets</i>)	554.623	-	425.444	não	980.067	Automático
Copa das Conf.*	Futebol (<i>tweets</i>)	1.794	463	813	não	3.070	Misto
Córpus 7x1*	Futebol (<i>tweets</i>)	449	1.101	1.178	não	2.728	Manual
Computer-BR*	Produtos (<i>tweets</i>)	197	1.677	443	não	2.317	Manual
Eleições-Dilma	Política (<i>tweets</i>)	46.808	-	19.835	não	66.643	Manual
Eleições-Serra	Política (<i>tweets</i>)	1.371	-	8.347	não	9.718	Manual
ReLi	<i>Reviews</i> de livros	2.883	8.991	596	não	12.470	Manual
TTsBR	Programas de TV (<i>tweets</i>)	6.648	3.926	4.426	não	15.000	Manual

Fonte: Elaborada pelo autor.

com um desbalanceamento considerável nas classes, onde a classe negativa representa somente 4% do tamanho total do córpus.

Para contornar o enviesamento de classes, uma das estratégias é o uso de *under-sampling*, ou seja, a retirada de documentos das classes majoritárias até que a distribuição de classes no córpus seja equivalente (BRUM; ARAUJO; KEPLER, 2016). Fazendo uso dessa abordagem, o córpus da Copa das Confederações seria reduzido a 1.389 documentos (45% do seu tamanho total), o córpus 7x1 teria 1.347 documentos (49% do seu tamanho total), o *Computer-BR* reduziria drasticamente para 591 documentos (25% do seu tamanho total), o ReLi também seria gravemente afetado com 1.788 documentos (14% do seu tamanho total), enquanto o TTsBR se manteria com 11.778 documentos (78% do seu tamanho total). Isso mostra que, apesar de desbalanceado, essa desvantagem pode ser superada por meio de um método simples sem afetar tanto o tamanho total do córpus.

Na classificação binária o Pelesent (JUNIOR *et al.*, 2017) se sobressai no tamanho total de documentos, o que se justifica pelo processo de etiquetamento de dados que usa *emoticons* e *emojis* (fenômenos recorrentes em *tweets*). Uma desvantagem desse córpus é justamente a ausência de *emoticons* e *emojis* no córpus final, o que elimina uma *feature* determinante na classificação de positividade e negatividade. Em seguida temos o Mercado Livre (AVANÇO, 2015), que trabalha com *reviews* de produtos e os corpú de Eleições (SILVA *et al.*, 2011), que se assemelham ao TTsBR pelo suporte linguístico do *Twitter*.

O tamanho e balanceamento dos corpú Buscapé e Mercado Livre (HARTMANN *et al.*, 2014) os tornam excelentes bases de dados para trabalhar com classificação em *reviews*, enquanto a classificação de *tweets* possui diversas alternativas - o Pelesent pelo tamanho e diversidade de domínio, os corpú de Eleições no domínio político, e o Copa das Confederações sobre futebol. Os corpú 7x1 e *Computer-BR* ainda carecem de dados para viabilizar treinamento de bons modelos de classificação. Nesse sentido o TTsBR se destaca por proporcionar um domínio inédito de trabalho no Português, visto que Programas de TV podem ser adaptados para diferentes idiomas (algo que futebol, por exemplo, pode acabar limitando por sua popularidade proeminente

na América do Sul e Europa) e ainda oferece boa distribuição de *tweets* positivos, neutros e negativos (podemos notar nos *corpus* de outros domínios uma classe com baixa representação).

Entre os *corpus* anotados manualmente, o TTsBR é o segundo que possui mais documentos anotados. Apesar da agilidade no processo da anotação automática, é sempre difícil de se garantir confiabilidade nas etiquetas, visto que não existe uma supervisão humana na anotação.

5.1.1 Comparativo na tarefa de classificação de polaridades

Usamos os classificadores apresentados na [Subseção 4.2.3](#) para avaliar o TTsBR em comparação aos demais *corpus* já existentes. Primeiramente avaliamos cada método sobre o TTsBR para avaliar qual a média de *F-Measure* obtida na classificação de polaridades.

Os experimentos mostrados na [Tabela 14](#) apresentam valores de *F1* para cada classe, sendo que os experimentos foram feitos usando os subconjuntos de treino e teste discutidos na [Subseção 4.1.4](#).

Tabela 14 – Avaliação do TTsBR na classificação de polaridades em três classes usando configuração Treino/Teste.

Classificador	F-Pos	F-Neu	F-Neg	F-Measure
SVM	73.3	47.2	62.2	60.89
Naive Bayes	70.4	48.2	58.8	59.12
Regressor Logístico	76.6	51.7	66.3	64.87
Multilayer Perceptron	76.3	51.9	65.6	64.60
Árvore de Decisão	67.8	40.3	55.4	54.50
Random Forest	74.9	43.2	68.4	62.18

Fonte: Elaborada pelo autor.

O classificador que obteve os melhores resultados foi o que usa Regressão Logística, atingindo 64% na classificação com três classes, porém a *Multilayer Perceptron* (MLP) também chegou a resultados semelhantes.

A classe neutra obteve os piores resultados de *F-Measure* com todos os classificadores. Isso pode indicar a complexidade que a presença da classe neutra adiciona à tarefa de classificação de polaridades. Comparando os resultados com três classes aos obtidos somente na classificação binária, apresentados na [Tabela 15](#), observamos uma diferença média de 18% a mais na classificação binária.

Essa diferença vai ao encontro de outros experimentos obtidos na literatura, como os valores de estado-da-arte na classificação de polaridades no inglês, onde o maior valor obtido na classificação binária foi 89,9% enquanto a avaliação com cinco classes obtêm 52,4% ([ZHOU et al., 2016](#)).

Tabela 15 – Avaliação do TTsBR na classificação binária de polaridades usando configuração Treino/Teste.

Classificador	F-Pos	F-Neg	F-Measure
SVM	84,4	74,7	79,57
Naive Bayes	81,4	70,4	75,92
Regressor Logístico	86,4	77,5	81,95
Multilayer Perceptron	87,0	79,2	83,11
Árvore de Decisão	80,9	70,0	75,43
Random Forest	74,9	78,4	82,85

Fonte: Elaborada pelo autor.

Na classificação binária, o método que obteve melhores resultados foi a MLP, sendo que resultados semelhantes foram obtidos com *Random Forest* e com Regressão Logística. Esses resultados ficam próximos dos obtidos na literatura em trabalhos no *Twitter*, como em [Alves et al. \(2014\)](#), que obteve *F-Measure* de 80% no corpus da Copa das Confederações. [Moraes et al. \(2016\)](#) também usou o *Twitter* como base de dados, porém sua avaliação foi feita com medida de acurácia, o que não avaliamos em nossos experimentos, pois essa medida só deve ser utilizada com corpus balanceados.

Para comparar mais precisamente as avaliações de corpus, utilizamos nossos classificadores para avaliar todos os corpus da literatura sob os mesmos ambientes. Para esse comparativo não utilizamos os subconjuntos de treino e teste do TTsBR, mas o *10-fold cross validation*. A utilização de um corpus único de avaliação para todos os modelos afetaria diretamente corpus de domínios e gêneros diferentes do conjunto de avaliação, o que fugiria da proposta desse experimento, que é somente observar a classificação de cada corpus da literatura em seu próprio domínio.

Primeiramente avaliamos o único corpus que possui anotação da classe neutra - o ReLi. A [Tabela 16](#) apresenta os resultados obtidos em ambos os corpus ressaltando os maiores valores obtidos em cada um. A distribuição de *F-Measure* das classes difere entre os corpus: enquanto o TTsBR obtém os maiores valores de *F-Measure* para as duas classes polares, mas os menores para a classe neutra (semelhante aos obtidos na [Tabela 14](#)).

Enquanto a diferença de documentos entre as classes minoritária (negativa) e demais pode justificar os baixos valores obtidos no ReLi, a diferença no TTsBR não é tão significativa - a classe neutra só contém 500 *tweets* a menos que a negativa, enquanto a diferença entre *F-Measure* é de quase 20% em alguns classificadores.

Como já citado anteriormente, uma das possíveis razões para esse comportamento na classificação é a ausência de *features* discriminativas para a classe neutra. A literatura possui uma carência grande no estudo de tal segmento, portanto o uso de algum modelo de representação que consiga discriminar neutralidade para os classificadores pode ser muito relevante na análise de três classes.

Tabela 16 – Classificação de polaridades em três classes com os corpú ReLi e TTsBR.

Classificador	TTsBR				ReLi			
	F-Pos	F-Neu	F-Neg	F-Measure	F-Pos	F-Neu	F-Neg	F-Measure
SVM	70,7	44,5	60,7	59,23	58,4	85,4	26,3	56,75
Naive Bayes	68,7	44,1	56,2	56,36	55,4	84,6	24,6	54,88
Regressor Logístico	73,6	46,3	63,1	61,06	60,0	87,8	22,4	56,78
Multilayer Perceptron	74,1	45,5	63,9	61,23	58,1	87,6	15,2	53,65
Árvore de Decisão	66,7	42,8	54,5	54,73	49,4	83,0	15,5	49,34
Random Forest	73,6	41,5	64,1	59,77	48,2	87,2	5,8	47,10

Fonte: Elaborada pelo autor.

Em seguida apresentamos os resultados obtidos com a classificação binária dos dados. A mesma configuração de *10-fold cross validation* foi utilizada para esses experimentos, sendo que os corpú ReLi e TTsBR foram limitados a suas classes polares.

Além disso, os experimentos usando o corpú Pelesent foram feitos usando como representação *word-embeddings* com 600 dimensões, como descrito na [Subseção 4.2.2](#). Utilizamos essa representação por questões de recursos computacionais disponíveis, visto que a utilização de *bag-of-words* consome oito vezes mais memória que as *word-embeddings* de tamanho fixo, tornando inviável o armazenamento em memória das centenas de milhares de documentos que o corpú Pelesent é constituído.

Tabela 17 – Comparação de classificação de corpú usando SVM com duas classes.

Córpus	F-Pos	F-Neg	F-Measure
Buscapé-1	82,95	82,39	82,67
Buscapé-2	83,89	83,26	83,58
Mercado Livre	93,28	93,13	93,20
Eleições Dilma	94,22	85,97	90,09
Eleições Serra	89,03	98,22	93,63
Pelesent*	78,37	69,59	73,98
Reli	91,35	51,54	71,45
TTsBR	<u>82,37</u>	<u>72,99</u>	<u>77,68</u>

Fonte: Elaborada pelo autor.

A [Tabela 17](#) apresenta os resultados obtidos com o classificador SVM. Os corpú que obtiveram os melhores valores de *F-Measure* foram os das Eleições Presidenciais - isso se repetiu nos demais classificadores, portanto acreditamos ser indício de que as *features* utilizadas são eficientes na divisão do espaço para a classificação de polaridades.

Os resultados obtidos com o SVM no corpú Mercado Livre ficaram 2% abaixo dos relatados em [Avanço, Brum e Nunes \(2016\)](#) usando o mesmo classificador aliado a um classificador baseado em léxico. Optamos por não utilizar a abordagem híbrida por conta da próxima

etapa do *framework*, onde precisamos inserir a classe neutra na classificação. A inserção de uma nova classe demandaria novas regras linguísticas para a identificação da classe neutra, portanto utilizamos exclusivamente classificadores baseados em AM.

É possível também observar a diferença entre o TTsBR e o ReLi quando avaliados em duas e três classes (Tabela 16), como o aumento considerável da *F1* obtida na classe negativa do cópús ReLi.

Usando o classificador Naive Bayes, obtivemos valores inferiores aos do SVM em cerca de 8%, como demonstra a Tabela 18. Essa redução é observada igualmente nos trabalhos da literatura (AVANÇO, 2015; ALVES *et al.*, 2014). A incerteza do Naive Bayes aumentou a lacuna nos cópús balanceados (Buscapé-1, Buscapé-2 e Mercado Livre), algo não tão agravado usando SVM.

Tabela 18 – Comparação de classificação de cópús usando Naive Bayes com duas classes.

Cópús	F-Pos	F-Neg	F-Measure
Buscapé-1	76,59	73,32	74,95
Buscapé-2	77,89	73,95	75,92
Mercado Livre	84,10	82,79	83,44
Eleições Dilma	89,94	75,15	82,54
Eleições Serra	67,42	93,44	80,43
Pelesent	66,14	63,00	64,57
Reli	90,14	42,02	66,08
<u>TTsBR</u>	<u>80,06</u>	<u>69,03</u>	<u>74,54</u>

Fonte: Elaborada pelo autor.

A Tabela 19 apresenta os resultados obtidos com a Regressão Logística. Esse método obteve os melhores resultados em Junior *et al.* (2017) na tarefa de classificação de polaridade em *cross-domain* (onde os cópús de treino e teste são de domínios diferentes).

A Regressão Logística conseguiu resultados melhores que os do SVM em cópús onde os dados se encontravam mais balanceados (Buscapé-1, Buscapé-2, Mercado Livre e TTsBR). O classificador SVM consegue generalizar melhor os dados com menos exemplos, enquanto a Regressão Logística precisa de maior representação dos dados para dividir o espaço corretamente.

O melhor resultado obtido no cópús Buscapé-1 foi no modelo treinado usando Regressão Logística.

Usamos também uma MLP com duas camadas e 200 neurônios para a classificação (Tabela 20). A MLP é um dos métodos mais robustos que utilizamos na classificação, portanto obteve resultados muito satisfatórios, sendo o método que obteve a maior *F-Measure* nos cópús Buscapé-2, Mercado Livre, Pelesent e no TTsBR.

Sua maior vantagem é o número de cortes no espaço que o modelo alcança, sendo a base

Tabela 19 – Comparação de classificação de corpus usando Regressão Logística com duas classes.

Cópus	F-Pos	F-Neg	F-Measure
Buscapé-1	85,51	84,31	84,91
Buscapé-2	86,83	86,11	86,47
Mercado Livre	94,57	94,46	94,52
Eleições Dilma	94,13	85,45	89,78
Eleições Serra	88,71	98,22	93,46
Pelesent	78,35	69,58	73,95
Reli	92,03	44,51	68,26
<u>TTsBR</u>	<u>84,78</u>	<u>75,99</u>	<u>80,38</u>

Fonte: Elaborada pelo autor.

das técnicas mais recentes de *Deep Learning*. Uma das desvantagens desse classificador é o *overfitting*, onde a função se aproxima tanto dos pontos que perde seu potencial de generalização, ou seja, perde sua eficiência para dados não vistos. Algo que podemos observar é a classificação no corpus ReLi, onde existem tão poucos documentos negativos que a maioria dos dados é rotulada como positiva, diminuindo drasticamente a *F1* da classe negativa quando comparado aos outros classificadores.

Tabela 20 – Comparação de classificação de corpus usando *Multilayer Perceptron* com duas classes.

Cópus	F-Pos	F-Neg	F-Measure
Buscapé-1	84,97	83,49	84,23
Buscapé-2	88,06	87,27	87,66
Mercado Livre	94,81	94,72	94,76
Eleições Dilma	95,53	89,34	92,43
Eleições Serra	87,76	98,07	92,91
Pelesent	78,16	69,95	74,05
Reli	91,80	32,83	62,32
<u>TTsBR</u>	<u>85,08</u>	<u>76,85</u>	<u>80,96</u>

Fonte: Elaborada pelo autor.

A [Tabela 21](#) apresenta os resultados obtidos com Árvore de Decisão. Os valores foram semelhantes aos obtidos com Naive Bayes, portanto abaixo dos demais.

Outra abordagem foi o uso de Random Forest na classificação de polaridades. Sendo o Random Forest um comitê de Árvores de Decisão, seus resultados superaram consideravelmente o modelo anterior, atingindo o maior valor de *F-Measure* no corpus Eleições-Dilma. Os resultados podem ser vistos na [Tabela 22](#).

Tabela 21 – Comparação de classificação de córpis usando Árvore de Decisão com duas classes.

Córpis	F-Pos	F-Neg	F-Measure
Buscapé-1	79,59	78,42	79,00
Buscapé-2	79,42	79,11	79,26
Mercado Livre	87,32	87,10	87,20
Eleições Dilma	94,47	86,72	90,59
Eleições Serra	84,48	97,44	90,95
Pelesent	66,40	56,78	61,66
Reli	89,49	38,98	64,23
<u>TTsBR</u>	<u>79,53</u>	<u>67,42</u>	<u>74,05</u>

Fonte: Elaborada pelo autor.

Tabela 22 – Comparação de classificação de córpis usando Random Forest com duas classes.

Córpis	F-Pos	F-Neg	F-Measure
Buscapé-1	83,57	82,43	83,00
Buscapé-2	85,04	85,00	85,02
Mercado Livre	92,93	92,99	92,96
Eleições Dilma	96,11	90,19	93,15
Eleições Serra	88,33	98,26	93,30
Pelesent	74,56	51,90	63,22
Reli	91,79	19,04	55,41
<u>TTsBR</u>	<u>84,73</u>	<u>75,40</u>	<u>80,06</u>

Fonte: Elaborada pelo autor.

5.1.2 Resumo dos experimentos com o TTsBR

Exceto nos córpis Pelesent e ReLi, conseguimos atingir valores acima de 80% de *F-Measure* nos córpis utilizados. Os córpis que obtiveram os maiores valores foram o Mercado Livre e os córpis de Eleições. Pelo comportamento do córpis de Eleições na classificação, podemos dizer que as *features* escolhidas conseguiram discriminar corretamente os dados, visto que, mesmo alterando os classificadores, seus valores continuaram sempre entre os maiores em comparação com os demais córpis.

Ainda existe uma lacuna muito grande nas representações para a classe neutra, uma causa disso pode ser a ausência de uma definição formal para a classe.

Os valores obtidos com os classificadores implementados foram condizentes com os obtidos na literatura, e alguns podem ser utilizados como *baselines* para motivar mais pesquisa na AS - em especial com os córpis das Eleições que não haviam sido avaliados para essa tarefa.

Os valores obtidos com o TTsBR na classificação em três classes são importantes para a avaliação da extensão do córpis, apresentada na próxima seção. A [Tabela 23](#) apresenta o resumo dos melhores valores obtidos em cada córpis.

Tabela 23 – Visão geral da classificação de polaridades nos corpúscos da literatura (corpúscos ReLi e TTsBR possuem duas entradas assinaladas com * para indicar classificação em três classes).

Corpúscos	Classificador	F-Pos	F-Neu	F-Neg	F-Measure
ReLi*	Regressão Logística	60,0	87,8	22,4	56,78
TTsBR*	MLP	74,1	45,5	63,9	61,23
Buscapé-1	Regressão Logística	85,51	-	84,31	84,91
Buscapé-2	MLP	88,06	-	87,27	87,66
Mercado Livre	MLP	94,81	-	94,72	94,76
Eleições Dilma	Random Forest	96,11	-	90,19	93,15
Eleições Serra	SVM	89,03	-	98,22	93,63
Pelesent	MLP	78,16	-	69,95	74,05
Reli	SVM	91,35	-	51,54	71,45
TTsBR	MLP	85,08	-	76,85	80,96

Fonte: Elaborada pelo autor.

5.2 Experimentos com expansão semissupervisionada

Nossa avaliação do *framework* de expansão fez uso do TTsBR como treinamento inicial para a classificação de 117.050 *tweets* extraídos anteriormente junto do material anotado. Esses *tweets* foram pré-processados da mesma forma e compreendem o mesmo domínio, gênero e compartilham o suporte linguístico do ambiente virtual.

Para cada classificador utilizado na pesquisa (SVM, Naive Bayes, Regressão Logística, MLP, Árvore de Decisão e *Random Forest*) geramos um corpúscos, utilizando o mesmo na anotação dos documentos. Inicialmente fizemos uma busca pelos melhores hiperparâmetros e representações para a classificação do TTsBR, essas configurações serão apresentadas na [Subseção 5.2.1](#).

Para avaliar nosso método, utilizamos o conjunto de teste do TTsBR como conjunto de *held-out*, ou seja, avaliamos cada corpúscos gerado no treinamento de um modelo de classificação e utilizamos esse modelo no conjunto de teste do TTsBR. Dessa maneira comparamos os resultados obtidos com cada corpúscos entre si e com a classificação usando somente os dados anotados manualmente.

Esse método de anotação indica se o corpúscos anotado semi-automaticamente consegue resultados equivalentes ou melhores do que os obtidos usando um conjunto anotado manualmente, porém consideravelmente menor. Isso não exclui a possibilidade de se realizar a anotação dos 117.050 documentos, medindo assim exatamente a acurácia da classificação semi-automática, porém isso foi inviável devido aos altos custos de anotação envolvidos na tarefa.

Como descrito na [Subseção 4.2.4](#), utilizamos um *threshold* fixo para definir a quantidade de documentos a serem adicionados ao corpúscos de treinamento em cada iteração do aprendizado. Outros trabalhos da literatura exploram outras maneiras de se aplicar um modelo semissupervi-

onado iterativo, como adicionar um *threshold* de confiança na anotação (SILVA, 2016; SILVA; COLETTA; HRUSCHKA, 2016).

Optamos por não utilizar a abordagem do *threshold* na confiança, pois experimentos iniciais apontaram para a dificuldade de se definir um valor fixo que garanta a confiabilidade na classificação e faça uso de boa parte dos documentos disponibilizados para a anotação.

No primeiro momento, a maioria dos trabalhos da literatura que trabalha com duas classes consegue valores de *F-Measure* e Acurácia altos em comparação com a classificação em três polaridades, portanto *thresholds* altos se mostram eficientes em manter a confiança e garantir em parte o balanceamento do conjunto de treino. Quando um *threshold* muito alto foi definido em nossos experimentos, rapidamente uma classe (positiva) tornou-se desproporcionalmente majoritária, reunindo mais de 70% dos documentos do *corpus*, visto que o próprio TTsBR possui certo enviesamento (como pode ser visto na *F1* obtida na classe positiva nos experimentos da seção anterior). Devido a isso, a classe neutra (minoritária) acabou crescendo menos de 5% e logo nenhum documento classificado era adicionado ao *corpus*, por conta da baixa confiança dos modelos.

Experimentamos adicionar um número fixo de documentos de cada classe respeitando o *threshold*, mas limitando a adição à classe minoritária. Nesses casos, após uma ou duas iterações os modelos sofreram com a vazão limitada da classe minoritária, a classe neutra começou a obter somente 1 ou 2 documentos por iteração, aumentando o número de iterações (visto que as demais classes eram limitadas por essa) e rapidamente não possuindo mais documentos para serem adicionados - encerrando a adição. Nos poucos experimentos que fizemos, pouco mais de cinco mil documentos foram incorporados ao *corpus* final.

Por essas razões, utilizamos um valor fixo de adição, ou *threshold* de adição, onde definimos uma porcentagem e, a cada iteração, adicionamos um número fixo de documentos, ordenados pela confiança de cada modelo. Discutimos os resultados obtidos com essa abordagem na Subseção 5.2.2.

Para tentar reduzir o enviesamento de classes, realizamos experimentos também fazendo *under-sampling* do TTsBR, ou seja, retirando documentos das classes positiva e negativa até que o *corpus* se tornasse balanceado. Essa abordagem surtiu o efeito esperado e em alguns casos conseguiu gerar *corpus* finais que se destacaram na classificação. Os resultados com o TTsBR reduzido são discutidos na Subseção 5.2.3.

Variamos os *threshold* em 1%, 5%, 10%, 20%, 25%, 30% e 40% para experimentar desde *corpus* gerados com muitas iterações (onde o enviesamento foi consideravelmente superior ao longo do aprendizado), até *corpus* gerados com poucas iterações (onde o risco de ruído na classificação é extremamente alto).

Além de avaliar a classificação em três classes, avaliamos também a classificação binária, descartando a classe neutra. A expansão do TTsBR em duas classes não deve ser realizada,

pois provavelmente existem documentos neutros no *córpus* não anotado, portanto utilizamos os mesmos *córpus* avaliados com três classes.

Apesar dos nossos experimentos indicarem a eficiência do *framework* na anotação semissupervisionada, é muito difícil garantir que tenhamos atingido 100% de certeza em qualquer dos *córpus*. Sendo nosso objetivo fornecer uma alternativa na anotação semi-automática de *córpus*, os resultados apresentados indicam somente se os *córpus* anotados semi-automaticamente são tão bons quanto aqueles com o uso de um conjunto de dados consideravelmente menor, porém anotado manualmente. Buscando avaliar um pouco mais nosso *framework*, aplicamos a mesma técnica de *self-training* usada nos dados não anotados em *córpus* da literatura.

Nesses experimentos usamos 10% dos *córpus* para o treinamento inicial do modelo e, usando os mesmos *thresholds* de adição já apresentados, realizamos a classificação dos demais documentos. De posse das etiquetas corretas, podemos avaliar a Acurácia do nosso método. A vantagem desses experimentos é que eles nos ajudam a verificar mais precisamente a eficiência do *framework* e nos permitem observar como ele se comporta em outros domínios. A desvantagem é que existem poucos *córpus* de sentimentos anotados em três classes, portanto os experimentos nesses *córpus* não podem ser comparados diretamente aos realizados na expansão do TTsBR. Além disso, o tamanho dos *córpus* também limita o experimento a usar menos dados do que o TTsBR no treinamento (somente 10% dos *córpus*) e na expansão dos dados. Esses resultados são apresentados e discutidos na [Subseção 5.2.4](#).

5.2.1 Seleção de features e hiperparâmetros

Antes de realizarmos os experimentos com os métodos de classificação e expansão, buscamos a melhor combinação de *features* e hiperparâmetros para potencializar a classificação de polaridades.

Para isso, utilizamos o TTsBR para avaliar diferentes configurações de hiperparâmetros e *features* de representação (apresentadas na [Subseção 4.2.2](#)). Os valores apresentados para cada parâmetro foram estipulados empiricamente.

A avaliação de tais configurações foi feita levando em consideração a *F-Measure* devido ao desbalanceamento de classes do TTsBR. A utilização do subconjunto de teste também não pôde ser feita, pois enviesaria os parâmetros escolhidos para o conjunto avaliado.

Utilizamos uma parcela do *córpus* de treinamento para validar os parâmetros escolhidos com 10% do tamanho total do *córpus*, garantindo assim que os resultados obtidos são completamente independentes do conjunto de teste avaliado nos experimentos posteriores. Usamos a média de 5 execuções para ranquear as combinações., mantendo o mesmo subconjunto de validação.

Por conta do elevado número de execuções relacionadas a cada experimento (cerca de 8 mil tuplas) optamos por disponibilizar as tabelas completas virtualmente em <https://bitbucket>.

[org/HBrum/tweetsentbr/](https://twitter.com/HBrum/tweetsentbr/).

Na [Subsubseção 5.2.1.7](#) fazemos um resumo dos hiperparâmetros utilizados no decorrer dos experimentos e discutimos algumas possibilidades para trabalhos futuros.

5.2.1.1 SVM Linear

A representação que obteve os melhores resultados com SVM foram o *Bag-of-words*, a contagem de *emoticons* e a distribuição de *emojis*. Além disso, a seleção de *features* também obteve os melhores resultados na classificação. Para definir um valor de C, usamos 0,01, 0,01, 1 e 10, sendo que o valor que obteve os melhores resultados foi 1. A [Tabela 24](#) apresenta os 5 melhores resultados obtidos.

O método de seleção de *features* utilizado foi o mesmo apresentado em ([AVANÇO, 2015](#)), ele diminui o número de *features* treinando um classificador para identificar as *features* menos variantes nos dados. O mesmo método foi aplicado em todos os classificadores abaixo.

Tabela 24 – Melhores resultados obtidos na avaliação de hiperparâmetros com classificador SVM.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	C	F-Pos	F-Neu	F-Neg	F-Measure
X		X	X			X	1	73,47	50,29	64,32	62,69
X		X	X		X	X	0.1	74,26	47,85	65,29	62,47
X	X		X	X		X	1	73,07	49,92	64,17	62,38
X	X	X	X	X		X	1	72,99	49,44	64,60	62,34
X		X	X			X	1	72,91	49,75	64,21	62,29

Fonte: Elaborada pelo autor.

5.2.1.2 Naive Bayes

A representação que obteve os melhores resultados com o classificador Naive Bayes foi *bag-of-words*, contagem de *emoticons*, distribuição de *emojis*, léxico de sentimentos e etiquetas morfossintáticas - assim como com o SVM, a seleção de *features* melhorou os resultados. Para os valores do *alpha* do classificador, experimentamos 0, 1, 0,5 e 1, sendo que o melhor valor obtido foi com o *alpha* como 0,1. A [Tabela 25](#) apresenta os resultados para as melhores configurações do classificador bayesiano.

5.2.1.3 Regressão Logística

Realizamos experimentos somente com as representações usando regressão logística (não modificamos nenhum hiperparâmetro), a combinação de *features* que obteve os melhores resultados foi com *bag-of-words*, *emoticons*, *emojis* e etiquetas morfossintáticas. A [Tabela 26](#) apresenta os melhores resultados obtidos.

Tabela 25 – Melhores resultados obtidos na avaliação de hiperparâmetros com classificador Naive Bayes.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	Alpha	F-Pos	F-Neu	F-Neg	F-Measure
X		X	X	X	X	X	0,1	67,67	42,14	54,17	54,66
X	X	X	X		X	X	0,1	67,95	41,71	54,16	54,60
X		X	X	X	X	X	0,5	68,31	41,20	54,21	54,57
X		X	X	X		X	0,1	67,72	41,31	54,22	54,42
X		X	X			X	0,1	67,85	40,75	54,45	54,34

Fonte: Elaborada pelo autor.

Tabela 26 – Melhores resultados obtidos na avaliação de hiperparâmetros com Regressão Logística.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	F-Pos	F-Neu	F-Neg	F-Measure	Acc
X		X	X		X	X	0,7416	0,4781	0,6476	0,6224	0,6509
X		X	X	X		X	0,7386	0,4749	0,6529	0,6221	0,6504
X	X	X	X	X		X	0,7371	0,4768	0,6516	0,6218	0,6498
X	X	X	X			X	0,7390	0,4765	0,6498	0,6217	0,6501
X			X	X	X	X	0,7373	0,4781	0,6478	0,6210	0,6490

Fonte: Elaborada pelo autor.

5.2.1.4 Multilayer Perceptron

Por ser um método que compreende múltiplos cortes no espaço e um número elevado de hiperparâmetros, optamos por utilizar todas as *features* na representação dos dados durante os experimentos com a *Multilayer Perceptron*.

Quanto às variações, fizemos experimentos com uma e duas camadas, variando o número de neurônios (o mesmo para ambas as camadas) em 30, 60, 100 e 200. Variamos também o *alpha* em 0,0001, 0,001 e 0,01 e a taxa de aprendizado (*learning rate*) entre 0,001, 0,1 e 0,1. [Tabela 27](#) mostra os melhores resultados, sendo que a configuração escolhida foi usando 2 camadas com 200 neurônios, *alpha* de 0,0001 e a taxa de aprendizado mais baixa, 0,001.

Tabela 27 – Melhores resultados obtidos na avaliação de hiperparâmetros com *Multilayer Perceptron*.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	Camadas	Neurônios	Alpha	Learning-rate	F-Pos	F-Neu	F-Neg	F-Measure
X	X	X	X	X	X	X	2	200	0,0001	0,001	73,89	48,80	65,20	62,63
X	X	X	X	X	X	X	2	100	0,001	0,001	74,30	48,46	65,06	62,60
X	X	X	X	X	X	X	2	100	0,01	0,001	73,90	48,77	65,07	62,58
X	X	X	X	X	X	X	2	60	0,01	0,001	73,82	48,18	65,22	62,40
X	X	X	X	X	X	X	2	100	0,0001	0,001	73,98	47,75	65,35	62,36

Fonte: Elaborada pelo autor.

5.2.1.5 Árvore de Decisão

Para o classificador baseado em árvore de decisão variamos o critério de divisão usando o Coeficiente de Gini e Entropia. Também realizamos experimentos com a profundidade máxima

da árvore de decisão, usando os valores 4, 5, 8 e deixando sem limite. Optamos por experimentar a limitação da altura da árvore para reduzir o custo computacional, visto que reduziu consideravelmente o tempo de treinamento do modelo, porém os melhores valores foram obtidos sem limite de altura e usando o Coeficiente de Gini como critério de divisão.

A Tabela 28 apresenta os resultados obtidos nas melhores configurações. Os valores com limite de profundidade estiveram abaixo de grande parte das execuções sem limite e nem figuraram entre as 5 melhores configurações. A representação utilizada é composta de *bag-of-words*, presença/ausência de negação, *emoticons*, *emojis* e esse foi o único classificador onde a seleção de *features* diminuiu a *F-Measure*.

Tabela 28 – Melhores resultados obtidos na avaliação de hiperparâmetros com Árvores de Decisão.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	Critério	Prof. Max.	F-Pos	F-Neu	F-Neg	F-Measure
X	X	X	X				gini	-	65,66	41,39	53,80	53,61
X	X	X	X			X	entropy	-	65,88	42,28	52,00	53,39
X	X	X	X	X			entropy	-	65,86	42,04	52,03	53,31
X	X	X	X	X		X	gini	-	65,30	42,68	51,88	53,29
X	X	X	X	X			gini	-	65,79	41,00	52,84	53,21

Fonte: Elaborada pelo autor.

5.2.1.6 Random Forest

Random Forest é basicamente um comitê de árvores de decisão, portanto seus hiperparâmetros são muito semelhantes. Repetimos os experimentos com a profundidade das árvores (4, 5, 8 e sem limite) e usando o Coeficiente de Gini e Entropia como critérios. A principal diferença é o número de estimadores, ou seja, de árvores de decisão utilizadas no comitê. Utilizamos 30, 60, 100 e 200 estimadores, sendo que, como esperado, o maior número de estimadores atingiu os melhores valores de *F-Measure*.

Um grande risco do grande número de estimadores é o *overfitting*, isso justifica novamente o uso de uma parcela aleatória do treinamento para a avaliação, visto que o uso do conjunto de teste enviesaria a classificação somente para esses dados. A representação utilizada foi semelhante à utilizada na classificação com Árvore de Decisão, porém aqui a seleção de *features* obteve os melhores resultados; o mesmo aconteceu com o critério de divisão por Entropia que obteve melhores resultados na Random Forest.

5.2.1.7 Resumo dos hiperparâmetros escolhidos

Os experimentos feitos sobre os hiperparâmetros e representações usados nos classificadores conseguiram aumentar um pouco a *F-Measure* obtida por estes, porém um dos maiores dilemas da nossa avaliação é justamente a avaliação dos modelos durante a etapa de expansão. Os 117.050 *tweets* que usamos para expandir o corpus não possuem qualquer anotação de etiqueta para avaliarmos a classificação, isso nos motivou ainda mais em tentar obter o melhor ambiente

Tabela 29 – Melhores resultados obtidos na avaliação de hiperparâmetros com *Random Forest*.

BoW	Neg	Emot	Emoj	Sent.	PoS	FS	Estimadores	Critério	Prof. Max.	F-Pos	F-Neu	F-Neg	F-Measure
X	X	X	X			X	200	entropy	-	71,94	44,87	60,91	59,23
X	X	X	X	X		X	200	entropy	-	71,80	44,71	61,10	59,20
X		X	X			X	200	gini	-	71,79	44,43	61,10	59,10
X		X	X	X		X	200	entropy	-	71,75	45,22	60,28	59,08
X		X	X	X		X	200	gini	-	71,67	44,56	61,01	59,07

Fonte: Elaborada pelo autor.

de classificação antes da aplicação do *self-training*. A definição dos hiperparâmetros ainda facilita a replicação dos experimentos em trabalhos comparativos futuros.

O *bag-of-words* é a representação que melhor discrimina nossos dados, visto que todos os modelos se comportaram consideravelmente melhor com essa *feature*. Apesar de não constar na tabela reduzida, realizamos experimentos com *word-embeddings* nos dados (nestes casos não agregamos nenhuma outra representação, ou efetuamos seleção de *features*), porém os resultados obtidos não foram satisfatórios.

Acreditamos que a densidade das *word-embeddings* funciona melhor na inicialização de arquiteturas de *deep learning* do que nos classificadores que utilizamos na nossa pesquisa.

A seguir está o resumo das configurações de cada classificador nos próximos experimentos reportados:

- **SVM:** *bag-of-words*, léxico de *emoticons*, distribuição de *emojis* e seleção de *features*. C: 1.
- **Naive Bayes:** *bag-of-words*, léxico de *emoticons*, distribuição de *emojis*, termos positivos e negativos, etiquetas morfossintáticas e seleção de *features*. alpha: 0,1.
- **Regressão Logística:** *bag-of-words*, léxico de *emoticons*, distribuição de *emojis*, etiquetas morfossintáticas e seleção de *features*.
- **MLP:** *bag-of-words*, presença de negação, léxico de *emoticons*, distribuição de *emojis*, termos positivos e negativos, etiquetas morfossintáticas e seleção de *features*. Duas camadas com 200 neurônios em cada uma, alpha: 0,0001, taxa de aprendizado: 0,001.
- **Árvore de Decisão:** *bag-of-words*, presença de negação, léxico de *emoticons* e distribuição de *emojis*, sem seleção de *features*. Critério de divisão: coeficiente de Gini, sem profundidade máxima.
- **Random Forest:** *bag-of-words*, presença de negação, léxico de *emoticons*, distribuição de *emojis* e seleção de *features*. Número de estimadores: 200, critério de divisão: entropia, sem profundidade máxima.

5.2.2 Framework de Expansão com o TTsBR completo

Utilizando nosso *framework*, expandimos o TTsBR usando os 117.050 documentos não anotados que haviam sido extraídos da mesma forma que os dados que formaram o *corpus*. Para cada classificador utilizado, usamos a abordagem iterativa para avaliar todos os *tweets*, ordenar todos por probabilidade e definimos vários *thresholds* para adicionar os dados de diferentes maneiras.

Após todas as iterações, definidas pelo *threshold* de adição, avaliamos o *corpus* na classificação de polaridades usando o conjunto de testes que serviu de *held-out* para a avaliação. Treinamos um modelo com cada *corpus* gerado para cada classificador usado, ou seja, cada *corpus* foi avaliado seis vezes (uma com cada classificador) para que pudéssemos observar se a diferença de valores de cada *corpus* se mantinha com diferentes classificadores.

Por conta do elevado número de execuções e dos diversos resultados gerados, as tabelas relacionadas a essa subseção estão apresentadas no [Apêndice D](#), ao final deste documento. As tabelas estão organizadas por *threshold* de adição, classificador utilizado na expansão e classificador usado na avaliação. É importante ressaltar que os valores obtidos representam a média de cinco execuções de cada avaliação. Cada *corpus* foi gerado somente uma vez.

Para facilitar a comparação das avaliações de cada *corpus*, calculamos a média das *F-Measure* obtidas com cada classificador e apresentamos na [Tabela 30](#). O valor obtido com a mesma avaliação usando somente o conjunto de treinamento do TTsBR foi de 61,03% de *F-Measure*.

O *corpus* expandido que obteve os melhores valores na avaliação foi o originado por meio da MLP. Uma observação interessante é que durante a avaliação do TTsBR ([Tabela 14](#)), o classificador que obteve os melhores valores foi a Regressão Logística, seguido de perto pela MLP (diferença de 0,20%), entretanto, o classificador que obteve a melhor *FI* para a classe neutra foi a MLP. Essa informação é importante, visto que um dos grandes desafios na expansão é evitar o enviesamento da classificação, pois ele ocasiona ruído no *corpus* gerado e esse erro pode se propagar na classificação.

Tabela 30 – Resumo da *F-Measure* média obtida com os *corpus* expandidos usando *self-training* com três classes.

Classificador	40%	30%	25%	20%	10%	5%	1%
SVM Linear	59,58	58,73	59,47	58,73	55,80	54,15	52,12
Naive Bayes	54,97	53,69	52,73	52,41	50,18	49,45	47,09
Regressão Logística	59,91	58,41	58,12	57,04	53,2	50,86	48,76
MLP	62,14	61,65	61,74	61,40	61,19	61,02	61,04
Árvore de Decisão	57,85	57,54	58,39	56,44	58,29	58,45	57,93
Random Forest	57,72	55,99	54,33	53,31	49,61	49,23	49,06

Fonte: Elaborada pelo autor.

A Figura 21 apresenta graficamente os valores da Tabela 30 com a adição de uma linha representando o TTsBR como *córpus de treinamento* (sem *self-training*).

Figura 21 – Gráfico de *F-Measure* média obtida com cada *córpus* expandido usando *self-training* na avaliação em três classes.

Fonte: Elaborada pelo autor.

Em geral os métodos obtiveram os melhores resultados usando *thresholds* maiores, o que diminui o número de iterações, mas aumenta o número de documentos adicionados em cada passo. Ainda que a maior parte dos valores tenha ficado abaixo dos resultados com o TTsBR, ressalta-se que os *córpus* expandidos possuem cerca de oito vezes mais documentos anotados que o TTsBR, sendo que a parcela anotada manualmente corresponde apenas a 10%. No caso do classificador MLP, por exemplo, os resultados ficaram até melhores para *thresholds* altos.

Os resultados que mostraram valores mais baixos de *F-Measure* foram os obtidos no *córpus* gerado pelo classificador Naive Bayes. Esse resultado vai ao encontro da avaliação apresentada na Tabela 14, onde o mesmo classificador obteve os menores resultados na classificação.

Acreditamos que um fator que influencia muito nessa classificação é o desbalanceamento do *córpus*. Na Tabela 35 podemos ver a distribuição de classes geradas para cada *córpus* com os diferentes *thresholds* de adição. O *córpus* final de todos os métodos de classificação e com todas as porcentagens de adição sempre obtém o menor número de documentos neutros, sendo que essa diferença é em média de 10% de documentos a menos do que os da classe negativa e quase 30% a menos do que os da classe positiva (majoritária). Esse desbalanceamento é ainda mais evidente quando aumentamos as iterações, ou seja, com *thresholds* menores como 1% e 5%.

Nos piores casos, somente 3% dos documentos do *córpus* final são classificados como negativos, sendo que os documentos anotados manualmente (que fazem parte do *córpus* final), representam cerca de 1/3 desses documentos. Curiosamente, apesar do baixo valor obtido nas avaliações, o classificador Naive Bayes não sofre tanto com esse desbalanceamento quando

muitas iterações são feitas, ainda que os baixos resultados indiquem que muitos documentos estejam sendo incorretamente etiquetados nesses casos.

Uma explicação para esse fenômeno é que o desbalanceamento do TTsBR influencia muito nas primeiras iterações do *framework*, portanto esse desbalanceamento é propagado até o fim da execução, fazendo com que poucos documentos neutros sejam adicionados e, consequentemente, a distância entre as classes majoritárias e minoritárias aumente ainda mais.

Se observarmos o comportamento da avaliação do *corpus* gerado com SVM, vamos identificar uma queda brusca na avaliação a partir do *threshold* de 10%. Observando a [Tabela 35](#), vemos que com *thresholds* maiores que esse teremos uma distribuição muito próxima de 20%, enquanto que a partir dos experimentos com 10% temos a classe neutra em 13%, depois 10% (com *threshold* de 5%) e finalmente 7% (com *threshold* de 1%).

Como não sabemos a verdadeira distribuição dos dados, é difícil saber qual a distribuição ideal para o *corpus* final, mas nos baseando na anotação do TTsBR, teríamos como distribuição 44% documentos na classe positiva, 26% na classe neutra e 29% na classe negativa. Nos *corpus* onde obtivemos os melhores valores de *F-Measure*, as distribuições de classes são bem semelhantes a essas, como é o caso da Árvore de Decisão, que se manteve com bons resultados inclusive com baixo *threshold*, e com a MLP com poucas iterações.

Para a classificação binária, utilizamos os mesmos *corpus* formados ignorando-se a classe neutra. Realizar uma nova expansão não faria sentido, pois entre os documentos do *corpus* sem anotação teremos a ocorrência de sentenças neutras.

Os resultados obtidos com a classificação binária (seguindo a mesma abordagem de média de *F-Measure* da classificação anterior) estão dispostos na [Tabela 31](#).

Como apresentado anteriormente, a classificação binária costuma obter resultados muito superiores à classificação em três classes. A avaliação no conjunto de teste (*held-out*) obteve 79,8% de *F-Measure* média com os classificadores, porém mesmo nesse caso poucos *corpus* conseguiram superar esses valores.

O *corpus* gerado com a MLP se destacou novamente dos demais obtendo valores superiores aos do *corpus* manual com quase todos os valores de *threshold* usados. Já o *corpus* gerado com Naive Bayes continuou sendo o menos confiável, obtendo valores muito abaixo dos demais.

É interessante observarmos na [Figura 22](#) que o comportamento das avaliações binárias difere um pouco das avaliações com três classes, pois com muitas iterações (*threshold* de 1%), existe um ganho de performance na avaliação com a Árvore de Decisão e a *Random Forest*, assim como a queda de *F-Measure* não é tão drástica quanto na avaliação em três classes.

Uma possível causa é que a classe neutra age como uma zona de incerteza na classificação. Sendo bem definidos os espaços de documentos positivos e negativos (graças a *features* mais discriminativas para essas classes), os documentos classificados como neutros podem ser vistos

Tabela 31 – Resumo da *F-Measure* média obtida com os corpúis expandidos usando *self-training* na classificação binária.

Classificador	40%	30%	25%	20%	10%	5%	1%
SVM Linear	78,4	78,2	78,6	78,6	77,5	76,4	75,5
Naive Bayes	74,4	74,1	73,6	72,9	72,6	67,2	63,3
Regressão Logística	79,2	78,5	78,1	78,2	76,8	75,9	73,1
MLP	80,3	80,8	80,1	80,0	80,5	79,5	80,2
Árvore de Decisão	76,1	76,0	76,2	76,6	75,4	76,9	76,2
Random Forest	74,5	72,68	70,8	70,4	69,4	69,9	70,6

Fonte: Elaborada pelo autor.

como incertos para a classificação binária. Ignorando a classe neutra, restam apenas documentos que foram etiquetados com mais confiança nas classes polares extremas e isso faz com que a propagação de erro nas iterações seja reduzida.

Figura 22 – Gráfico de *F-Measure* média obtida com cada corpúis expandido usando *self-training* na avaliação binária.

Fonte: Elaborada pelo autor.

Os resultados obtidos com a expansão do TTsBR foram promissores, porém o desbalanceamento dos corpúis deixa uma margem grande para a desconfiança na anotação semi-automática. Buscando diminuir um pouco esse enviesamento, realizamos experimentos fazendo *under-sampling* do TTsBR.

5.2.3 Framework de Expansão com o TTsBR reduzido

Para tentar reduzir o enviesamento dos dados e diminuir os ruídos nas iterações do *framework*, realizamos experimentos com uma versão reduzida do TTsBR.

Semelhante ao processo descrito em [Brum, Araujo e Kepler \(2016\)](#), realizamos o *under-sampling* do TTsBR, equilibrando o número de documentos entre as classes. Para tal, retiramos aleatoriamente documentos das classes positiva e negativa até que ambas possuíssem o mesmo total da classe minoritária (2.894 *tweets*), que em nosso caso era a neutra. Os documentos retirados das classes foram adicionados ao cópús para expansão, portanto o valor final dos cópús expandidos nessa abordagem são os mesmos da expansão do TTsBR completo.

Assim como na subseção anterior, disponibilizamos os valores completos de todos os experimentos no [Apêndice E](#).

A [Tabela 32](#) apresenta os resultados obtidos nas médias das *F-Measures* obtidas com cada cópús. O classificador que gerou os cópús que obtiveram os melhores resultados foi o MLP, entretanto, usando dois *thresholds* (30% e 25%), o cópús expandido com Regressão Logística atingiu resultados ligeiramente maiores.

O classificador SVM e a Regressão Logística obtiveram valores maiores que os apresentados na [Tabela 30](#) (onde apresentamos os valores obtidos na expansão do TTsBR completo) em todos os *thresholds* verificados.

Tabela 32 – Resumo da *F-Measure* média obtida com os cópús expandidos usando *self-training* com o TTsBR reduzido na classificação em três classes.

Classificador	40%	30%	25%	20%	10%	5%	1%
SVM Linear	60,57	60,84	60,69	60,81	60,91	59,19	56,60
Naive Bayes	57,16	56,08	55,27	54,36	49,06	46,78	45,01
Regressão Logística	61,45	61,71	61,48	61,55	58,93	53,56	50,30
MLP	62,13	60,64	61,10	61,68	61,60	61,50	61,64
Árvore de Decisão	58,36	57,72	58,00	57,58	57,68	57,96	58,31
Random Forest	59,34	58,04	57,29	55,52	52,01	50,66	48,99

Fonte: Elaborada pelo autor.

O Naive Bayes continuou a obter os resultados menos expressivos, enquanto observamos uma melhora considerável nos cópús gerados pelo SVM e pela Árvore de Decisão. Esse comportamento é mais facilmente observável na [Figura 23](#). O valor comparativo é o mesmo utilizado na subseção anterior já que a tarefa a ser comparada é a mesma. Podemos notar que os resultados obtidos com o cópús originado da MLP se mantêm acima da avaliação feita com o TTsBR em boa parte dos experimentos. Isso é um forte indício de que o cópús estendido e anotado semi-automaticamente consegue resultados superiores aos obtidos com o cópús manual original.

O cópús originado com Regressão Logística também obteve resultados muito satisfatórios. Comparando com os resultados obtidos na expansão do TTsBR completo, podemos dizer que este modelo foi o mais afetado pelo balanceamento das classes. Exceto nos experimentos com *threshold* baixo, a Regressão Logística obteve resultados superiores em comparação à

Figura 23 – Gráfico de *F-Measure* média obtida com cada cópús expandido com o TTsBR reduzido usando *self-training* na avaliação em três classes.

Fonte: Elaborada pelo autor.

mesma avaliação usando o cópús manual.

Uma evidência importante a ser observada é que a diminuição do *threshold* acaba favorecendo a maximização dos documentos da classe majoritária ou de mais fácil classificação. Se observarmos a Tabela 36, onde apresentamos as distribuições de classes dos cópús originados com a expansão do TTsBR reduzido, vamos notar que a distribuição se parece muito nos *thresholds* mais altos, enquanto continua enviesada em *thresholds* menores.

A literatura já havia indicado que a utilização de *thresholds* mais abrangentes (em nosso caso, *thresholds* maiores) favorecem a adição de ruídos nos cópús de treinamento, porém nossos experimentos mostram que, em nossa abordagem, o contrário, ou seja, *thresholds* menores, favorece o enviesamento do treinamento, podendo gerar resultados não tão satisfatórios.

Tabela 33 – Resumo da *F-Measure* média obtida com os cópús expandidos usando *self-training* com o TTsBR reduzido na classificação binária.

Classificador	40%	30%	25%	20%	10%	5%	1%
SVM Linear	79,1	78,8	79,0	79,0	78,9	78,6	77,7
Naive Bayes	75,5	74,6	74,1	74,1	66,1	60,6	58,6
Regressão Logística	79,3	79,0	79,2	78,9	78,5	77,0	74,6
MLP	79,8	78,8	79,1	79,9	79,4	79,4	79,5
Árvore de Decisão	76,3	75,0	75,2	75,0	75,3	76,4	75,3
Random Forest	77,6	76,4	74,8	72,7	66,4	66,7	68,4

Fonte: Elaborada pelo autor.

Avaliamos também a classificação binária nos cópús originados com o TTsBR reduzido.

Os valores das médias de *F-Measure* estão apresentados na [Tabela 33](#).

O mais curioso dessa avaliação é que, apesar da melhora obtida com o balanceamento do corpus na classificação em três classes, a classificação binária obteve resultados inferiores. Usando como comparativo o valor de 79,80% obtido com o modelo treinado só com dados anotados manualmente, somente o corpus originado com a MLP conseguiu resultados maiores e por 0,01%.

Os resultados ainda são bons, considerando que temos um corpus muito maior anotado semi-automaticamente, mas a própria expansão usando o TTsBR completo obteve resultados superiores. Uma explicação para isso talvez seja o que destacamos na subseção anterior: a classe neutra agindo como uma zona de incerteza na classificação.

Figura 24 – Gráfico de *F-Measure* média obtida com cada corpus expandido com o TTsBR reduzido usando *self-training* na avaliação binária.

Fonte: Elaborada pelo autor.

A [Figura 24](#) mostra o gráfico dos resultados obtidos, onde podemos ver que os modelos variam muito menos do que na classificação em três classes. Isso sugere que o balanceamento do corpus inicial gera corpus expandidos mais semelhantes e, portanto, os valores obtidos na avaliação ficam mais estáveis mesmo com diferentes *thresholds*.

Ainda observamos a queda de rendimento dos corpus gerados com o Naive Bayes e com *Random Forest* em *thresholds* muito baixos. De acordo com a [Tabela 36](#), vemos que são os corpus que têm uma queda drástica no número de documentos da classe positiva.

Podemos concluir que a utilização de um balanceamento de corpus forçado não favorece a formação de corpus expandidos melhores para a tarefa de classificação de polaridades binária. Já na classificação em três classes, o balanceamento do corpus consegue reduzir o enviesamento da classificação e obter resultados mais promissores.

O classificador que obteve os melhores resultados na expansão de *corpus* foi a MLP, sendo que em diversos casos a avaliação realizada superou os valores de *F-Measure* obtidos na classificação somente com os dados anotados manualmente. Concluímos dessa forma que são fortes as evidências de que a utilização do aprendizado semissupervisionado para a expansão de um *corpus* anotado é capaz de gerar grandes *corpus* para os quais os resultados da classificação de polaridade podem superar aqueles do *corpus* original.

5.2.4 Framework de expansão com *corpus* da literatura

Buscamos os *corpus* apresentados na literatura para avaliar nosso *framework* usando dados completamente anotados. A vantagem de utilizar *corpus* já anotados é que podemos avaliar mais precisamente o quão bem anotado o *corpus* final estará. A principal desvantagem é que nosso *framework* foi construído para a expansão de grandes *corpus* de sentimentos, o que faz com que poucos *corpus* da literatura sejam bons casos de estudo para ele.

Para a expansão, escolhemos aleatoriamente um subconjunto estratificado de cada *corpus* com 10% do tamanho do *corpus* total o qual também será usado como conjunto de treinamento para o *framework*. Em seguida expandimos os dados usando o conjunto restante (sem olhar as etiquetas) e avaliamos o *corpus* final como se fosse a predição de um classificador.

Utilizamos os *corpus* Buscapé-2, Mercado Livre, Eleições-Dilma e Eleições-Serra para a expansão binária. O *corpus* Pelesent foi descartado, pois foi construído usando supervisão distante, ou seja, não existe qualquer garantia de que os dados do *corpus* estão corretamente anotados. Pode-se advogar que os *corpus* Buscapé-2 e Mercado Livre também foram anotados automaticamente, porém ambos têm suas etiquetas baseadas em pontuações fornecidas pelos usuários, ou seja, existiu um tipo de anotação humana por trás da etiqueta de cada documento.

Também optamos por descartar o *corpus* Buscapé-1 por conta de seu tamanho reduzido. Com 2.000 documentos não seria recomendado realizar qualquer tipo de expansão com esse *corpus*.

A seguir apresentamos os melhores resultados obtidos com cada *corpus* e comparamos os resultados com os obtidos através da classificação em 10 *fold cross-validation*, apresentados anteriormente na [Tabela 23](#). As tabelas com os valores completos da classificação estão apresentadas no [Apêndice F](#).

5.2.4.1 *Corpus* Buscapé-2

O *corpus* Buscapé-2 ([HARTMANN et al., 2014](#)) se assemelha com o TTsBR pelo seu tamanho, porém se diferencia no domínio de *reviews* de produtos. Sendo as *features* usadas em nossa implementação desenvolvidas para o domínio de programas televisivos no ambiente virtual, espera-se que o resultado da expansão ainda consiga ser melhorado com algumas pequenas alterações no modelo.

Em nossos experimentos obtivemos um *corpus* anotado semi-automaticamente com 84,74% de *F-Measure*. Em nossa classificação usando *10 fold cross-validation*, o maior valor obtido foi 87,66% de *F-Measure* usando Regressão Logística na classificação. Curiosamente, o método de expansão que conseguiu a melhor avaliação foi a MLP, porém a Regressão Logística ficou menos de 1% abaixo.

O *threshold* que obteve o *corpus* mais bem anotado foi com 1%, ou seja, o maior número de iterações e o menor número de documentos sendo adicionados por iteração. Isso contradiz nossos experimentos anteriores com o TTsBR, porém vale lembrar que o *corpus* Buscapé-2 é consideravelmente menor do que os *corpus* expandidos que utilizamos nas etapas anteriores. Além disso, o domínio de *reviews* de produtos geralmente atinge resultados acima de 90% de acurácia sem o uso de arquiteturas mais complexas ou sistemas mais robustos, isso pode indicar que o domínio não ofereça tantos desafios e isso justifica os bons resultados, mesmo adicionando dados em muitas iterações.

5.2.4.2 *Corpus Mercado Livre*

O *corpus* Mercado Livre (AVANÇO, 2015), também no domínio de *reviews* de produtos, possui como vantagem ter um bom número de documentos em sua composição. A mesma característica do *corpus* Buscapé-2 pode ser observada aqui, resultados de estado-da-arte para esse *corpus* atingem cerca de 96% de *F-Measure* (AVANÇO; BRUM; NUNES, 2016), ou seja, as *features* clássicas de AS conseguem discriminar muito bem os dados e dividir com mais facilidade o espaço.

Usando Regressão Logística com *threshold* de 10%, conseguimos anotar semi-automaticamente um *corpus* com 93,17% de *F-Measure*. Em geral as expansões com MLP e Regressão Logística obtiveram resultados muito semelhantes com diferentes *thresholds* de adição.

Em relação a todos os *corpus* anotados em nosso experimento, o Mercado Livre foi o que obteve melhores resultados. A *F-Measure* mais alta obtida com esse *corpus* em nossos experimentos foi de 94,76% usando MLP.

5.2.4.3 *Corpus Eleições-Dilma*

O *corpus* de eleições traz um desafio a mais para a classificação, pois trata de um domínio completamente diferente dos programas televisivos. Além de se tratar de um domínio específico (política), o *corpus* havia sido proposto originalmente para outra tarefa (SILVA *et al.*, 2011). Entretanto, na classificação de polaridades obtivemos *F-Measure* de 93,15% usando *Random Forest* e na expansão do *corpus*, conseguimos uma *F-Measure* de 83,69% usando MLP. O melhor *corpus* foi obtido com *threshold* de 1%.

A *FI* da classe positiva atingida foi de 90,7%, o que destaca a característica do *corpus*, que possui um desbalanceamento muito relevante para a classe positiva. Outro comportamento

interessante foram os baixos valores obtidos com a expansão usando SVM Linear: com *thresholds* mais baixos a *F1* da classe negativa chegou a 2%, enquanto no melhor caso se manteve com 41,9%.

5.2.4.4 *Cópus Eleições-Serra*

O *cópus* eleições-Serra é muito semelhante ao conjunto anterior, porém consideravelmente menor. Sua classificação obteve 93,63% de *F-Measure* com o classificador SVM. Usando nosso *framework* de expansão conseguimos uma *F-Measure* de 88,23% na anotação do *cópus* final.

O melhor valor obtido foi com *Random Forest*. Diferente dos demais *cópus*, os maiores valores obtidos no Eleições-Serra foram com valores altos de *threshold*.

5.3 Comparação do TTsBR expandido com *cópus* anotado por supervisão distante

O trabalho de Junior *et al.* (2017), apresentado no Capítulo 3, diz respeito a um *cópus* de sentimentos anotado usando supervisão distante. Os autores usaram *emojis* e *emoticons* para anotar uma grande parcela de documentos automaticamente.

Pela similaridade com nossa proposta, decidimos comparar o *cópus* daquele trabalho, Pelesent, com o nosso *cópus* que obteve os melhores resultados na classificação binária.

Utilizamos nossos melhores classificadores na avaliação dos *cópus* e *word-embeddings* para representação dos dados, visto que o *bag-of-words* é muito custoso quando aplicado ao *cópus* Pelesent. As avaliações foram feitas em *cross-domain*, ou seja, os classificadores foram treinados sobre o TTsBR e o Pelesent para a avaliação em outros *cópus* da literatura.

Usamos os mesmos *cópus* usados na avaliação do Pelesent em Junior *et al.* (2017): Buscapé-1, Buscapé-2, Mercado Livre, Eleições-Dilma e Eleições-Serra. A Tabela 34 apresenta os resultados obtidos.

Nosso *framework* de expansão se mostrou uma alternativa mais eficiente que a supervisão distante em quase todos os experimentos, com destaque para a avaliação do *cópus* Eleições-Serra, onde nosso *cópus* expandido conseguiu generalizar melhor os documentos de um *cópus* de domínio político e obteve uma grande diferença entre a avaliação com o Pelesent.

O *cópus* Pelesent possui 980.067 documentos, sete vezes maior que o *cópus* expandido que utilizamos (com 128.030 documentos). Isso significa que nossa abordagem consegue resultados tão bons quantos o da supervisão distante, apesar de seu tamanho consideravelmente menor. Além disso, a formação de um *cópus* por meio de aprendizagem distante demanda tempo de extração de dados e sofre com a limitação das duas classes polares. Nossa abordagem tem a

Tabela 34 – Comparação com o TTsBR expandido usando MLP com *threshold* de 30% com o córpus Pelesent na avaliação de *cross-domain* com os córpus da literatura.

Córpus Avaliado	Classificador	TTsBR expandido			Córpus Pelesent		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
Eleições-Dilma	SVM Linear	52,97	73,90	63,45	80,8	42,6	61,69
	Regressão Logística	56,03	75,27	65,66	79,8	40,3	60,06
	MLP	51,90	72,90	62,39	79,2	46,3	62,78
Eleições-Serra	SVM Linear	79,33	28,80	54,07	20,4	36,5	28,45
	Regressão Logística	79,70	28,27	53,99	20,7	35,0	27,81
	MLP	78,47	27,00	52,72	20,3	39,6	29,90
Mercado Livre	SVM Linear	84,90	82,97	83,93	77,5	62,6	70,01
	Regressão Logística	85,07	82,90	83,97	77,6	62,5	70,04
	MLP	85,07	83,60	84,34	79,3	69,8	74,54
Buscapé-1	SVM Linear	66,53	73,47	69,99	70,1	56,8	63,46
	Regressão Logística	69,87	73,87	71,86	70,3	57,3	63,80
	MLP	65,90	73,10	69,49	70,6	62,3	66,41
Buscapé-2	SVM Linear	77,40	79,73	78,55	72,9	53,9	63,39
	Regressão Logística	78,63	79,77	79,18	73,1	54,2	63,63
	MLP	77,47	79,50	78,45	73,8	57,0	65,38

Fonte: Elaborada pelo autor.

vantagem de conseguir ser replicada em outras tarefas, assim como para diferentes números de classes e domínios.

Tabela 35 – Distribuição das classes nos corpus expandidos usando *self-training* a partir do TTsBR completo.

<i>Threshold</i> de Adição	Classificador	# pos	# neu	# neg
40%	SVM Linear	62.065 (48%)	29.084 (22%)	36.881 (28%)
	Naive Bayes	64.917 (50%)	22.705 (17%)	40.408 (31%)
	Regressão Logística	67.093 (52%)	24.364 (19%)	36.573 (28%)
	MLP	58.611 (45%)	34.965 (27%)	34.454 (26%)
	Árvore de Decisão	57.100 (44%)	34.896 (27%)	36.034 (28%)
	<i>Random Forest</i>	71.661 (55%)	24.986 (19%)	31.383 (24%)
30%	SVM Linear	63.617 (49%)	27.111 (21%)	37.302 (29%)
	Naive Bayes	65.262 (50%)	19.297 (15%)	43.471 (33%)
	Regressão Logística	70.163 (54%)	20.727 (16%)	37.140 (29%)
	MLP	63.286 (49%)	23.910 (18%)	40.834 (31%)
	Árvore de Decisão	58.139 (45%)	33.443 (26%)	36.448 (28%)
	<i>Random Forest</i>	75.892 (59%)	22.890 (17%)	29.248 (22%)
25%	SVM Linear	64.594 (50%)	25.864 (20%)	37.572 (29%)
	Naive Bayes	65.293 (51%)	17.005 (13%)	45.732 (35%)
	Regressão Logística	72.412 (56%)	18.142 (14%)	37.476 (29%)
	MLP	62.627 (48%)	31.005 (24%)	34.398 (26%)
	Árvore de Decisão	57.484 (44%)	34.784 (27%)	35.762 (27%)
	<i>Random Forest</i>	79.554 (62%)	21.107 (16%)	27.369 (21%)
20%	SVM Linear	66.100 (51%)	24.098 (18%)	37.832 (29%)
	Naive Bayes	64.475 (50%)	15.475 (12%)	48.080 (37%)
	Regressão Logística	75.729 (59%)	14.922 (11%)	37.379 (29%)
	MLP	62.014 (48%)	31.568 (24%)	34.448 (26%)
	Árvore de Decisão	58.216 (45%)	33.781 (26%)	36.033 (28%)
	<i>Random Forest</i>	82.439 (64%)	18.478 (14%)	27.113 (21%)
10%	SVM Linear	71.773 (56%)	17.722 (13%)	38.535 (30%)
	Naive Bayes	61.698 (48%)	11.070 (8%)	55.262 (43%)
	Regressão Logística	84.043 (65%)	7.633 (5%)	3.6354 (28%)
	MLP	60.872 (47%)	28.815 (22%)	38.343 (29%)
	Árvore de Decisão	57.931 (45%)	34.097 (26%)	36.002 (28%)
	<i>Random Forest</i>	89.296 (69%)	12.479 (9%)	26.255 (20%)
5%	SVM Linear	76.334 (59%)	13.003 (10%)	38.693 (30%)
	Naive Bayes	34.434 (26%)	32.476 (25%)	61.120 (47%)
	Regressão Logística	88.704 (69%)	5.520 (4%)	33.806 (26%)
	MLP	63.645 (49%)	27.415 (21%)	36.970 (28%)
	Árvore de Decisão	57.311 (44%)	35.678 (27%)	35.041 (27%)
	<i>Random Forest</i>	91.143 (71%)	10.537 (8%)	263.50 (20%)
1%	SVM Linear	81.178 (63%)	9.360 (7%)	37.492 (29%)
	Naive Bayes	29.500 (23%)	39.521 (30%)	59.009 (46%)
	Regressão Logística	94.937 (74%)	4.819 (3%)	28.274 (22%)
	MLP	63.238 (49%)	25.308 (19%)	39.484 (30%)
	Árvore de Decisão	57.899 (45%)	33.679 (26%)	36.452 (28%)
	<i>Random Forest</i>	90.515 (70%)	9.481 (7%)	2803.4 (21%)

Fonte: Elaborada pelo autor.

Tabela 36 – Distribuição das classes nos corpúis expandidos usando *self-training* a partir do TTsBR reduzido.

<i>Threshold</i> de Adição	Classificador	# pos	# neu	# neg
40 %	SVM Linear	49.783 (38%)	40.042 (31%)	38.205 (29%)
	Naive Bayes	50.172 (39%)	35.753 (27%)	42.105 (32%)
	Regressão Logística	51.665 (40%)	37.610 (29%)	38.755 (30%)
	MLP	50.485 (39%)	38.908 (30%)	38.637 (30%)
	Árvore de Decisão	47.110 (36%)	41.391 (32%)	39.529 (30%)
	<i>Random Forest</i>	50.133 (39%)	39.352 (30%)	38.545 (30%)
30 %	SVM Linear	50.158 (39%)	39.677 (30%)	38.195 (29%)
	Naive Bayes	49.416 (38%)	33.853 (26%)	44.761 (34%)
	Regressão Logística	52.808 (41%)	36.124 (28%)	39.098 (30%)
	MLP	49.900 (38%)	40.219 (31%)	37.911 (29%)
	Árvore de Decisão	46.648 (36%)	41.613 (32%)	39.769 (31%)
	<i>Random Forest</i>	53.937 (42%)	39.064 (30%)	35.029 (27%)
25 %	SVM Linear	50.563 (39%)	39.300 (30%)	38.167 (29%)
	Naive Bayes	46.671 (36%)	34.836 (27%)	46.523 (36%)
	Regressão Logística	53.684 (41%)	34.914 (27%)	39.432 (30%)
	MLP	47.649 (37%)	43.384 (33%)	36.997 (28%)
	Árvore de Decisão	47.533 (37%)	41.106 (32%)	39.391 (30%)
	<i>Random Forest</i>	55.476 (43%)	37.996 (29%)	34.558 (26%)
20 %	SVM Linear	51.260 (40%)	38.661 (30%)	38.109 (29%)
	Naive Bayes	43.875 (34%)	35.416 (27%)	48.739 (38%)
	Regressão Logística	55.340 (43%)	32.539 (25%)	40.151 (31%)
	MLP	56.122 (43%)	36.052 (28%)	35.856 (28%)
	Árvore de Decisão	46.690 (36%)	41.521 (32%)	39.819 (31%)
	<i>Random Forest</i>	59.612 (46%)	38.609 (30%)	29.809 (23%)
10 %	SVM Linear	53.196 (41%)	35.438 (27%)	39.396 (30%)
	Naive Bayes	31.567 (24%)	38.655 (30%)	57.808 (45%)
	Regressão Logística	67.974 (53%)	17.531 (13%)	42.525 (33%)
	MLP	51.438 (40%)	38.317 (29%)	38.275 (29%)
	Árvore de Decisão	46.826 (36%)	41.424 (32%)	39.780 (31%)
	<i>Random Forest</i>	72.325 (56%)	30.247 (23%)	25.458 (19%)
5 %	SVM Linear	58.376 (45%)	29.490 (23%)	40.164 (31%)
	Naive Bayes	27.523 (21%)	41.758 (32%)	58.749 (45%)
	Regressão Logística	79.245 (61%)	8.486 (6%)	40.299 (31%)
	MLP	49.976 (39%)	40.609 (31%)	37.445 (29%)
	Árvore de Decisão	46.623 (36%)	42.132 (32%)	39.275 (30%)
	<i>Random Forest</i>	79.604 (62%)	21.773 (17%)	26.653 (20%)
1 %	SVM Linear	68.710 (53%)	19.232 (15%)	40.088 (31%)
	Naive Bayes	25.678 (20%)	43.636 (34%)	58.716 (45%)
	Regressão Logística	90.145 (70%)	5.698 (4%)	32.187 (25%)
	MLP	53.621 (41%)	34.241 (26%)	40.168 (31%)
	Árvore de Decisão	48.681 (38%)	39.701 (31%)	39.648 (30%)
	<i>Random Forest</i>	83.197 (64%)	13.594 (10%)	31.239 (24%)

Fonte: Elaborada pelo autor.

CONCLUSÕES E TRABALHOS FUTUROS

Em nosso trabalho tivemos o objetivo de avaliar a anotação semi-automática de um corpus de sentimentos, observando se essa anotação consegue igualar ou superar resultados obtidos com corpus anotados manualmente.

É indiscutível a importância da anotação manual de corpus, especialmente na área de PLN, porém os altos custos muitas vezes dificultam e entram pesquisas nas mais diversas áreas. Análise de Sentimentos em português é um exemplo, visto que a grande maioria dos trabalhos da área explora os recursos limitados disponíveis.

Neste trabalho, coletamos fortes evidências de que, sob certas condições (escolha de classificadores, parâmetros e *features*, entre outras), o processo de expansão aqui proposto é capaz de preservar a qualidade de um corpus inicial e manualmente anotado, na tarefa de classificação de polaridade.

Com nossos experimentos, mostramos que, usando um corpus inicial de 15.000 documentos anotados manualmente, foi possível expandi-lo para mais de 100.000 documentos anotados, capazes de gerar classificadores com desempenhos semelhantes - e por vezes melhores - do que os obtidos com o corpus inicial.

Mesmo que existam ruídos e incertezas na anotação semissupervisionada, a possibilidade de se obter um corpus anotado com centenas de milhares de documentos dispendendo pouco esforço pode favorecer pesquisas na área, ainda mais com o grande fluxo de dados que os ambientes virtuais nos proporcionam.

Além da proposta do *framework* de expansão, nossa pesquisa originou o TweetSentBR, um corpus inédito para a língua portuguesa que, contemplando a anotação da classe neutra, deve favorecer as pesquisas nessa linha, ainda incipientes para o português. O corpus possui tamanho relevante e enriquecerá o conjunto de recursos anotados manualmente que estão disponíveis para nosso idioma.

Investigamos diferentes *features* e hiperparâmetros, experimentando diversas alternativas e abordagens para resolver nosso problema, disponibilizando assim uma extensa análise de recursos linguísticos, fenômenos opinativos e métodos de classificação de sentimentos.

Comparando nosso *framework* com abordagem de supervisão à distância, que é recorrentemente utilizada na literatura, mostramos que nosso método consegue resultados superiores em classificação *cross-domain*, assim como pode ser adaptado para diferentes domínios e até mesmo outras tarefas.

É possível estender nossa pesquisa usando outros métodos iterativos ¹ como o *co-training* (WAN, 2009; YU, 2014) ou os métodos baseados em grafos (GOLDBERG; ZHU, 2006).

O uso dos corpús expandidos em arquiteturas de *Deep Learning* também pode ser explorado, visto que essas arquiteturas costumam obter melhores resultados quanto mais dados estão disponíveis para treinamento.

A variação de abordagens também é recomendável, visto que outros trabalhos da literatura (DASGUPTA; NG, 2009) fazem uso, por exemplo, de *Active Learning*. Essa abordagem pode enriquecer ainda mais a classificação durante as iterações do *framework*, assim como reduzir o enviesamento e a propagação de erro através do processo. Inserindo um passo de anotação manual guiada pelo classificador (enviando ao usuário, por exemplo, documentos menos confiáveis), as garantias de anotação poderiam aumentar consideravelmente.

Por fim, o TTsBR - bem como suas expansões - ainda pode servir de ferramenta linguística para o estudo de fenômenos na linguagem virtual ou das características mais determinantes de tarefas de PLN relacionadas. A Análise de Sentimentos, por exemplo, pode ser beneficiada com identificação de novas formações semânticas, definições mais precisas das classes e até a descoberta de *features* mais discriminativas para a classe neutra podem advir desse tipo de pesquisa.

¹ A implementação do *co-training* está disponível junto do nosso *framework* em <<https://bitbucket.org/HBrum/tweetsentbr/>>

REFERÊNCIAS

- ACHARYA, A.; HRUSCHKA, E. R.; GHOSH, J.; ACHARYYA, S. C³e: A framework for combining ensembles of classifiers and clusterers. **International Workshop on Multiple Classifier Systems**, p. 269–278, 2011. Citado na página 50.
- ALMEIDA, G. A. de M.; AVANÇO, L.; DURAN, M. S.; FONSECA, E. R.; NUNES, M. d. G. V.; ALUÍSIO, S. M. Evaluating phonetic spellers for user-generated content in brazilian portuguese. **International Conference on Computational Processing of the Portuguese Language**, p. 361–373, 2016. Citado na página 28.
- ALUÍSIO, S. M.; OLIVEIRA, L.; PINHEIRO, G. M. Os tipos de anotações, a codificação, e as interfaces do projeto lácio-web: Quão longe estamos dos padrões internacionais para córpus? **Anais do II TIL - Workshop em Tecnologia da Informação e da Linguagem Humana, Salvador, Centro de Convenções**, v. 5, 2004. Citado na página 23.
- ALVES, A. L. F.; BAPTISTA, C. d. S.; FIRMINO, A. A.; OLIVEIRA, M. G. d.; PAIVA, A. C. d. A comparison of svm versus naive-bayes techniques for sentiment analysis in tweets: a case study with the 2013 fifa confederations cup. **Proceedings of the 20th Brazilian Symposium on Multimedia and the Web**, p. 123–130, 2014. Citado nas páginas 43, 54, 55, 57, 58, 61, 81, 84 e 86.
- ANCHIÊTA, R. T.; NETO, F. A. R.; SOUSA, R. F. de; MOURA, R. S. Using stylometric features for sentiment classification. **International Conference on Intelligent Text Processing and Computational Linguistics**, p. 189–200, 2015. Citado na página 28.
- ARRUDA, G. D. de; ROMAN, N. T.; MONTEIRO, A. M. An annotated corpus for sentiment analysis in political news. In: **Proceedings of the 10th Brazilian Symposium in Information and Human Language Technology**. [S.l.: s.n.], 2015. p. 101–110. Citado na página 27.
- AVANÇO, L. V. **Sobre normalização e classificação de polaridade de textos opinativos na web**. 2015. Citado nas páginas 4, 10, 15, 20, 24, 41, 44, 51, 52, 53, 54, 56, 57, 58, 71, 73, 74, 78, 82, 86, 92 e 104.
- AVANÇO, L. V.; BRUM, H. B.; NUNES, M. d. G. V. Improving opinion classifiers by combining different methods and resources. **XIII Encontro Nacional de Inteligência Artificial e Computacional**, 2016. Citado nas páginas 4, 14, 20, 28, 53, 56, 57, 58, 73, 75, 77, 78, 85 e 104.
- BALAGE, P. P.; PARDO, T. A. S.; ALUISIO, S. M. An evaluation of the brazilian portuguese liwc dictionary for sentiment analysis. **Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology (STIL)**, p. 215–219, 2013. Citado nas páginas 16, 28, 41, 51, 53 e 57.
- BAUGH, W. bwbaugh: Hierarchical sentiment analysis with partial self-training. **Atlanta, Georgia, USA**, p. 539, 2013. Citado nas páginas 7 e 32.

- BECKER, L.; ERHART, G.; SKIBA, D.; MATULA, V. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. **Second Joint Conference on Lexical and Computational Semantics (* SEM)**, v. 2, p. 333–340, 2013. Citado nas páginas 7 e 32.
- BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003. Citado na página 76.
- BERGER, A. L.; PIETRA, V. J. D.; PIETRA, S. A. D. A maximum entropy approach to natural language processing. **Computational linguistics**, MIT Press, v. 22, n. 1, p. 39–71, 1996. Citado na página 17.
- BERMINGHAM, A.; SMEATON, A. On using twitter to monitor political sentiment and predict election results. In: **Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)**. [S.l.: s.n.], 2011. p. 2–10. Citado na página 27.
- BERTAGLIA, T. F. C. **Normalização textual de conteúdo gerado por usuário**. 2017. Citado nas páginas 62, 71 e 76.
- BERTAGLIA, T. F. C.; NUNES, M. d. G. V. Exploring word embeddings for unsupervised textual user-generated content normalization. **WNUT 2016**, p. 112, 2016. Citado nas páginas 28 e 43.
- BLITZER, J.; DREDZE, M.; PEREIRA, F. *et al.* Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. **ACL**, v. 7, p. 440–447, 2007. Citado nas páginas 36, 40 e 50.
- BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. **Proceedings of the eleventh annual conference on Computational learning theory**, p. 92–100, 1998. Citado na página 32.
- BOLLEGALA, D.; WEIR, D.; CARROLL, J. Cross-domain sentiment classification using a sentiment sensitive thesaurus. **IEEE transactions on knowledge and data engineering**, IEEE, v. 25, n. 8, p. 1719–1731, 2013. Citado na página 10.
- BONTCHEVA, K.; CUNNINGHAM, H.; ROBERTS, I.; TABLAN, V. Web-based collaborative corpus annotation: Requirements and a framework implementation. **New Challenges for NLP Frameworks**, p. 20–27, 2010. Citado na página 25.
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 79.
- BRUM, H.; ARAUJO, F.; KEPLER, F. Sentiment analysis for brazilian portuguese over a skewed class corpora. **International Conference on Computational Processing of the Portuguese Language**, p. 134–138, 2016. Citado nas páginas 4, 15, 19, 28, 41, 54, 57, 68, 78, 82 e 100.
- CHA, M.; HADDADI, H.; BENEVENUTO, F.; GUMMADI, P. K. Measuring user influence in twitter: The million follower fallacy. **Icwsm**, v. 10, n. 10-17, p. 30, 2010. Citado na página 12.
- CHAUDHURI, A. **Emotion and reason in consumer behavior**. [S.l.]: Routledge, 2006. ISBN 978-0-7506-7976-3. Citado na página 13.
- CHURCH, K. W.; HANKS, P. Word association norms, mutual information, and lexicography. **Computational linguistics**, MIT Press, v. 16, n. 1, p. 22–29, 1990. Citado na página 47.

COHEN, J. A coefficient of agreement for nominal scales. **Educational and psychological measurement**, Sage Publications, v. 20, n. 1, p. 37–46, 1960. Citado na página 27.

COHN, D.; ATLAS, L.; LADNER, R. Improving generalization with active learning. **Machine learning**, Springer, v. 15, n. 2, p. 201–221, 1994. Citado nas páginas 33 e 49.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 17.

DASGUPTA, S.; NG, V. Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In: **Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (ACL '09), p. 701–709. Citado nas páginas 30, 33, 49, 50, 58 e 110.

DAVE, K.; LAWRENCE, S.; PENNOCK, D. M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: **Proceedings of the 12th International Conference on World Wide Web**. New York, NY, USA: ACM, 2003. (WWW '03), p. 519–528. Citado nas páginas 9 e 18.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **Journal of the royal statistical society. Series B (methodological)**, JSTOR, p. 1–38, 1977. Citado na página 30.

DING, X.; LIU, B.; YU, P. S. A holistic lexicon-based approach to opinion mining. In: **Proceedings of the 2008 International Conference on Web Search and Data Mining**. New York, NY, USA: ACM, 2008. (WSDM '08), p. 231–240. Citado na página 36.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. **Machine learning**, Springer, v. 29, n. 2-3, p. 103–130, 1997. Citado na página 78.

DOSCIATTI, M. M.; PATERNO, L.; FERREIRA, C.; CABRERA, E. Anotando um Corpus de Notícias para a Análise de Sentimento: um Relato de Experiência. p. 121–130, 2015. Citado nas páginas 27 e 62.

DURAN, M. S.; AVANÇO, L. V.; NUNES, M. d. G. V. *et al.* A normalizer for ugc in brazilian portuguese. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS-ACL. **Workshop on Noisy User-generated Text**. [S.l.], 2015. Citado nas páginas 43 e 71.

ESULI, A.; SEBASTIANI, F. Determining the semantic orientation of terms through gloss classification. In: **Proceedings of the 14th ACM International Conference on Information and Knowledge Management**. New York, NY, USA: ACM, 2005. (CIKM '05), p. 617–624. Citado na página 16.

FLEISS, J. L.; COHEN, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. **Educational and psychological measurement**, Sage Publications Sage CA: Thousand Oaks, CA, v. 33, n. 3, p. 613–619, 1973. Citado nas páginas 27 e 65.

FONSECA, E. R.; ALUISIO, S. M. *et al.* A deep architecture for non-projective dependency parsing. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS-ACL. **Conference of the North American Chapter of the Association for Computational Linguistics-Human**

Language Technologies; Workshop on Vector Space Modeling for Natural Language Processing, I. [S.l.], 2015. Citado na página 76.

FONSECA, E. R.; ROSA, J. L. G.; ALUÍSIO, S. M. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. **Journal of the Brazilian Computer Society**, Springer London, v. 21, n. 1, p. 2, 2015. Citado na página 76.

FREITAS, C. Sobre a construção de um léxico da afetividade para o processamento computacional do português. **Revista Brasileira de Linguística Aplicada**, SciELO Brasil, v. 13, n. 4, p. 1013–1059, 2013. Citado nas páginas 16 e 42.

FREITAS, C.; MOTTA, E.; MILIDIÚ, R.; CÉSAR, J. Vampiro que brilha... rá! desafios na anotação de opiniao em um corpus de resenhas de livros. **Encontro de Lingística de corpus**, v. 11, p. 3, 2012. Citado nas páginas 4, 23, 27, 28, 40 e 61.

GAMON, M. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In: **Proceedings of the 20th International Conference on Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. (COLING '04). Citado na página 18.

GAMON, M.; AUE, A.; CORSTON-OLIVER, S.; RINGGER, E. Pulse: Mining customer opinions from free text. **International Symposium on Intelligent Data Analysis**, p. 121–132, 2005. Citado na página 23.

GO, A.; BHAYANI, R.; HUANG, L. Twitter sentiment classification using distant supervision. v. 1, n. 2009, p. 12, 2009. Citado nas páginas 6, 15, 42, 71 e 74.

GOLDBERG, A. B.; ZHU, X. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In: **ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing**. [S.l.], 2006. p. 45–52. Citado na página 110.

HARTMANN, N. S.; AVANÇO, L. V.; FILHO, P. P. B.; DURAN, M. S.; NUNES, M. d. G. V.; PARDO, T. A. S.; ALUISIO, S. M. A large corpus of product reviews in portuguese: tackling out-of-vocabulary words. **9th International Conference on Language Resources and Evaluation**, 2014. Citado nas páginas 4, 23, 24, 25, 27, 28, 43, 56, 57, 62, 82 e 103.

HATZIVASSILOGLLOU, V.; MCKEOWN, K. R. Predicting the semantic orientation of adjectives. In: **Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1997. (ACL '98), p. 174–181. Citado na página 16.

HOVY, E.; LAVID, J. Towards a 'science' of corpus annotation: a new methodological challenge for corpus linguistics. **International journal of translation**, v. 22, n. 1, p. 13–36, 2010. Citado nas páginas 3 e 22.

HU, M.; LIU, B. Mining and summarizing customer reviews. In: **Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2004. (KDD '04), p. 168–177. Citado nas páginas 2, 3, 11, 36, 40 e 57.

- JUNIOR, E. A. C.; MARINHO, V. Q.; SANTOS, L. B. d.; BERTAGLIA, T. F.; TREVISO, M. V.; BRUM, H. B. Pelesent: Cross-domain polarity classification using distant supervision. **arXiv preprint arXiv:1707.02657**, 2017. Citado nas páginas 3, 6, 42, 43, 56, 57, 58, 71, 73, 74, 76, 78, 82, 86 e 105.
- JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. ISBN 0130950696. Citado nas páginas 9 e 23.
- KAMVAR, K.; SEPANDAR, S.; KLEIN, K.; DAN, D.; MANNING, M.; CHRISTOPHER, C. Spectral learning. **International Joint Conference of Artificial Intelligence**, 2003. Citado na página 50.
- KILGARRIFF, A. 95% replicability for manual word sense tagging. In: **Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1999. (EACL '99), p. 277–278. Citado na página 26.
- KILGARRIFF, A.; GREFFENSTETTE, G. Introduction to the special issue on the web as corpus. **Computational linguistics**, MIT Press, v. 29, n. 3, p. 333–347, 2003. Citado na página 22.
- KIM, Y. Convolutional neural networks for sentence classification. **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014**, p. 1746–1751, 2014. Citado nas páginas 19, 40, 49 e 56.
- KLEIN, D.; MANNING, C. D. Accurate unlexicalized parsing. In: **Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003. (ACL '03), p. 423–430. Citado na página 36.
- KÖNIG, A. C.; BRILL, E. Reducing the human overhead in text categorization. In: **Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2006. (KDD '06), p. 598–603. Citado na página 18.
- KRIPENDORFF, K. Reliability in content analysis: Some common misconceptions. **Human Communications Research**, v. 30, p. 411–433, 2004. Citado nas páginas 27 e 65.
- KUMAR, A.; IRSOY, O.; ONDRUSKA, P.; IYYER, M.; BRADBURY, J.; GULRAJANI, I.; ZHONG, V.; PAULUS, R.; SOCHER, R. Ask me anything: Dynamic memory networks for natural language processing. **Proceedings of the 33rd International Conference on Machine Learning, ICML**, p. 1378–1387, 2016. Citado nas páginas 19, 40 e 49.
- LAVER, M.; BENOIT, K.; GARRY, J. Extracting policy positions from political texts using words as data. **American Political Science Review**, Cambridge Univ Press, v. 97, n. 02, p. 311–331, 2003. Citado na página 2.
- LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. **ICML**, v. 14, p. 1188–1196, 2014. Citado nas páginas 19, 49 e 54.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Research, v. 521, n. 7553, p. 436–444, 2015. Citado na página 18.

- LEECH, G. 100 million words of english: the british national corpus (bnc). **Language Research**, v. 28, n. 1, p. 1–13, 1992. Citado na página 23.
- LIU, B. Sentiment analysis and opinion mining. **Synthesis lectures on human language technologies**, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012. Citado nas páginas 15, 2, 9, 12, 13, 14, 18 e 24.
- LIU, Q.; GAO, Z.; LIU, B.; ZHANG, Y. Automated rule selection for aspect extraction in opinion mining. **IJCAI**, p. 1291–1297, 2015. Citado na página 36.
- MAEDA, K.; STRASSEL, S. Annotation tools for large-scale corpus development: Using agtk at the linguistic data consortium. **LREC**, 2004. Citado na página 25.
- MCCALLUM, A.; NIGAM, K. *et al.* A comparison of event models for naive bayes text classification. **AAAI-98 workshop on learning for text categorization**, v. 752, p. 41–48, 1998. Citado nas páginas 17 e 31.
- MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citado nas páginas 49, 52 e 76.
- MONARD, M. C.; BATISTA, G. E. Learning with skewed class distributions. **Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC**, v. 85, n. 2002, p. 173, 2002. Citado nas páginas 54 e 68.
- MORAES, S. M.; SANTOS, A. L.; REDECKER, M. S.; MACHADO, R. M.; MENEGUZZI, F. R. Classificação de sentimentos em nível de sentença: uma abordagem de múltiplas camadas para em língua portuguesa. **XIII Encontro Nacional de Inteligência Artificial e Computacional**, 2016. Citado nas páginas 4, 14, 25, 27, 45, 55, 57, 58, 61, 73, 77, 81 e 84.
- MORAES, S. M. W.; MANSSOUR, I. H.; SILVEIRA, M. S. 7x1pt: um corpus extraído do twitter para análise de sentimentos em língua portuguesa. **Proceedings of Symposium in Information and Human Language Technology**, 2015. Citado nas páginas 4, 25, 44, 61, 65 e 81.
- NAAMAN, M.; BOASE, J.; LAI, C.-H. Is it really about me?: Message content in social awareness streams. In: **Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work**. New York, NY, USA: ACM, 2010. (CSCW '10), p. 189–192. Citado na página 5.
- NAKOV, P.; RITTER, A.; ROSENTHAL, S.; SEBASTIANI, F.; STOYANOV, V. Semeval-2016 task 4: Sentiment analysis in twitter. **Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)**, 2016. Citado nas páginas 4, 6, 25, 26, 38, 39, 40, 50 e 56.
- NAKOV, P.; ROSENTHAL, S.; RITTER, A.; WILSON, T. SemEval-2013 Task 2: Sentiment Analysis in Twitter. **Proceedings of the International Workshop on Semantic Evaluation (SemEval-2013)**, v. 2, n. SemEval, p. 312–320, 2013. Citado nas páginas 37, 40 e 57.
- NASUKAWA, T.; YI, J. Sentiment analysis: Capturing favorability using natural language processing. In: **Proceedings of the 2Nd International Conference on Knowledge Capture**. New York, NY, USA: ACM, 2003. (K-CAP '03), p. 70–77. Citado na página 9.
- NG, A. Y.; JORDAN, M. I.; WEISS, Y. *et al.* On spectral clustering: Analysis and an algorithm. **NIPS**, v. 14, n. 2, p. 849–856, 2001. Citado na página 49.

- NOVAK, P. K.; SMAILOVIĆ, J.; SLUBAN, B.; MOZETIČ, I. Sentiment of emojis. **PloS one**, Public Library of Science, v. 10, n. 12, p. e0144296, 2015. Citado na página 75.
- O'CONNOR, B.; BALASUBRAMANYAN, R.; ROUTLEDGE, B. R.; SMITH, N. A. From tweets to polls: Linking text sentiment to public opinion time series. **ICWSM**, v. 11, n. 122-129, p. 1-2, 2010. Citado na página 15.
- OLIVEIRA, H. G.; SANTOS, A. P.; GOMES, P. Assigning polarity automatically to the synsets of a wordnet-like resource. **OASICS-OpenAccess Series in Informatics**, v. 38, 2014. Citado nas páginas 16 e 51.
- PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. **LREc**, v. 10, p. 1320-1326, 2010. Citado nas páginas 3, 42 e 74.
- PAN, S. J.; NI, X.; SUN, J.-T.; YANG, Q.; CHEN, Z. Cross-domain sentiment classification via spectral feature alignment. **Proceedings of the 19th International Conference on World Wide Web**, ACM, New York, NY, USA, p. 751-760, 2010. Citado na página 10.
- PANG, B.; LEE, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: **Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005. (ACL '05), p. 115-124. Citado nas páginas 2, 3, 14, 36 e 40.
- _____. Opinion mining and sentiment analysis. **Foundations and trends in information retrieval**, Now Publishers Inc., v. 2, n. 1-2, p. 1-135, 2008. Citado na página 3.
- PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: Sentiment classification using machine learning techniques. In: **Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (EMNLP '02), p. 79-86. Citado nas páginas 2, 11, 13, 14, 18, 23, 24, 36, 40, 47, 50, 73 e 77.
- PASQUALOTTI, P. R.; VIEIRA, R. Wordnet affect br: uma base lexical de palavras de emoções para a língua portuguesa. **RENOTE**, v. 6, n. 1, 2008. Citado na página 55.
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825-2830, 2011. Citado na página 77.
- PICARD, R. W. **Affective Computing**. Cambridge, MA, USA: MIT Press, 1997. ISBN 0-262-16170-2. Citado na página 2.
- QI, Y.; KUKSA, P.; COLLOBERT, R.; SADAMASA, K.; KAVUKCUOGLU, K.; WESTON, J. Semi-supervised sequence labeling with self-learned features. In: IEEE. **Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on**. [S.l.], 2009. p. 428-437. Citado na página 30.
- RAINA, R.; BATTLE, A.; LEE, H.; PACKER, B.; NG, A. Y. Self-taught learning: Transfer learning from unlabeled data. In: **Proceedings of the 24th International Conference on Machine Learning**. New York, NY, USA: ACM, 2007. (ICML '07), p. 759-766. ISBN 978-1-59593-793-3. Disponível em: <<http://doi.acm.org/10.1145/1273496.1273592>>. Citado na página 30.

- READ, J. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the ACL student research workshop**. [S.l.], 2005. p. 43–48. Citado na página 42.
- ROBERT, C. **Machine learning, a probabilistic perspective**. [S.l.]: Taylor & Francis, 2014. Citado nas páginas 77 e 78.
- ROSENBLATT, F. **Principles of neurodynamics. perceptrons and the theory of brain mechanisms**. [S.l.], 1961. Citado na página 78.
- ROSENTHAL, S.; NAKOV, P.; KIRITCHENKO, S.; MOHAMMAD, S. M.; RITTER, A.; STOYANOV, V. Semeval-2015 task 10: Sentiment analysis in twitter. **Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)**, p. 451–463, 2015. Citado nas páginas 38, 39 e 40.
- ROSENTHAL, S.; RITTER, A.; NAKOV, P.; STOYANOV, V. Semeval-2014 task 9: Sentiment analysis in twitter. In: **Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)**. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, 2014. p. 73–80. Citado nas páginas 38, 40 e 57.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning internal representations by error propagation**. [S.l.], 1985. Citado na página 78.
- SAGAE, K. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing**. [S.l.], 2010. p. 37–44. Citado na página 31.
- SAMMUT, C.; WEBB, G. I. **Encyclopedia of Machine Learning**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2011. ISBN 0387307680, 9780387307688. Citado na página 21.
- SILVA, I. S.; GOMIDE, J.; VELOSO, A.; JR, W. M.; FERREIRA, R. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In: ACM. **Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval**. [S.l.], 2011. p. 475–484. Citado nas páginas 4, 45, 56, 82 e 104.
- SILVA, M. J.; CARVALHO, P.; SARMENTO, L. Building a sentiment lexicon for social judgement mining. **International Conference on Computational Processing of the Portuguese Language**, p. 218–228, 2012. Citado nas páginas 16, 51, 55 e 75.
- SILVA, N. F. F. d. **Análise de sentimentos em textos curtos provenientes de redes sociais**. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado nas páginas 20, 58 e 90.
- SILVA, N. F. F. D.; COLETTA, L. F. S.; HRUSCHKA, E. R. A survey and comparative study of tweet sentiment analysis via semi-supervised learning. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 49, n. 1, p. 15:1–15:26, jun. 2016. Citado nas páginas 7 e 90.
- SILVA, N. F. F. da; COLETTA, L. F.; HRUSCHKA, E. R.; JR, E. R. H. Using unsupervised information to improve semi-supervised tweet sentiment classification. **Information Sciences**, Elsevier, v. 355, p. 348–365, 2016. Citado nas páginas 32, 50 e 77.

SOCHER, R.; PERELYGIN, A.; WU, J. Y.; CHUANG, J.; MANNING, C. D.; NG, A. Y.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. **Proceedings of the conference on empirical methods in natural language processing (EMNLP)**, v. 1631, p. 1642, 2013. Citado nas páginas 3, 4, 11, 14, 18, 19, 23, 25, 26, 36, 37, 40, 48, 49, 54, 56, 57 e 77.

SOUZA, M.; VIEIRA, R.; Buseti, D.; CHISHMAN, R.; ALVES, I. M. *et al.* Construction of a portuguese opinion lexicon from multiple resources. **8th Brazilian Symposium in Information and Human Language Technology**, p. 59–66, 2011. Citado nas páginas 16 e 51.

TABOADA, M.; BROOKE, J.; TOFILOSKI, M.; VOLL, K.; STEDE, M. Lexicon-based methods for sentiment analysis. **Computational linguistics**, MIT Press, v. 37, n. 2, p. 267–307, 2011. Citado nas páginas 15, 51 e 54.

TAI, K. S.; SOCHER, R.; MANNING, C. D. Improved semantic representations from tree-structured long short-term memory networks. **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers**, p. 1556–1566, 2015. Citado nas páginas 14, 19, 40 e 49.

TANHA, J.; SOMEREN, M. van; AFSARMANESH, H. Semi-supervised self-training for decision tree classifiers. **International Journal of Machine Learning and Cybernetics**, Springer, v. 8, n. 1, p. 355–370, 2017. Citado na página 30.

TONG, S.; KOLLER, D. Support vector machine active learning with applications to text classification. **Journal of machine learning research**, v. 2, n. Nov, p. 45–66, 2001. Citado nas páginas 49 e 50.

TREVISIO, M. V.; SHULBY, C.; ALUÍSIO, S. M. Sentence segmentation in narrative transcripts from neuropsychological tests using recurrent convolutional neural networks. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers**. [S.l.: s.n.], 2017. p. 315–325. Citado nas páginas 56 e 76.

TUMASJAN, A.; SPRENGER, T. O.; SANDNER, P. G.; WELPE, I. M. Election forecasts with twitter: How 140 characters reflect the political landscape. **Social science computer review**, Sage Publications Sage CA: Los Angeles, CA, v. 29, n. 4, p. 402–418, 2011. Citado na página 62.

TUMITAN, D.; BECKER, K. Tracking sentiment evolution on user-generated content: A case study on the brazilian political scene. **SBB (Short Papers)**, p. 24–1, 2013. Citado nas páginas 4 e 15.

_____. Sentiment-based features for predicting election polls: a case study on the brazilian scenario. In: IEEE. **Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on**. [S.l.], 2014. v. 2, p. 126–133. Citado na página 4.

TURNER, P. D. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: **Proceedings of the 40th Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ACL '02), p. 417–424. Citado nas páginas 2, 11, 13, 14, 15, 23, 24, 35, 40 e 46.

- VAPNIK, V.; GOLOWICH, S. E.; SMOLA, A. J. Support vector method for function approximation, regression estimation and signal processing. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1997. p. 281–287. Citado na página 77.
- WAN, X. Co-training for cross-lingual sentiment classification. In: **Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (ACL '09), p. 235–243. Citado nas páginas 30, 32 e 110.
- WANG, J.; SHEN, X.; PAN, W. On transductive support vector machines. **Contemporary Mathematics**, Providence, RI: American Mathematical Society, v. 443, p. 7–20, 2007. Citado na página 50.
- WARNER, W.; HIRSCHBERG, J. Detecting hate speech on the world wide web. In: **Proceedings of the Second Workshop on Language in Social Media**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. (LSM '12), p. 19–26. Citado na página 2.
- WIEBE, J.; WILSON, T.; CARDIE, C. Annotating expressions of opinions and emotions in language. **Language resources and evaluation**, Springer, v. 39, n. 2, p. 165–210, 2005. Citado na página 42.
- WILSON, T.; WIEBE, J.; HWA, R. Just how mad are you? finding strong and weak opinion clauses. **aaai**, v. 4, p. 761–769, 2004. Citado na página 11.
- XIANG, B.; ZHOU, L. Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**. [S.l.: s.n.], 2014. v. 2, p. 434–439. Citado na página 32.
- YANG, J.; QU, Z.; LIU, Z. Improved feature-selection method considering the imbalance problem in text categorization. **The Scientific World Journal**, Hindawi Publishing Corporation, v. 2014, 2014. Citado na página 55.
- YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: **Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995. (ACL '95), p. 189–196. Citado na página 31.
- YU, N. Exploring co-training strategies for opinion detection. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 65, n. 10, p. 2098–2110, 2014. Citado nas páginas 32 e 110.
- ZHAO, J.; LAN, M.; ZHU, T. T. Ecnu: Expression-and message-level sentiment orientation classification in twitter using multiple effective features. **SemEval 2014**, p. 259, 2014. Citado nas páginas 7 e 32.
- ZHOU, P.; QI, Z.; ZHENG, S.; XU, J.; BAO, H.; XU, B. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In: **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers**. Osaka, Japan: The COLING 2016 Organizing Committee, 2016. p. 3485–3495. Citado nas páginas 4, 19, 49 e 83.

ZHU, X. Semi-supervised learning literature survey. **Technical report. University of Wisconsin-Madison**, 2005. Citado nas páginas 7, 29, 32 e 33.

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. **Synthesis lectures on artificial intelligence and machine learning**, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–130, 2009. Citado na página 31.

MANUAL DE ANOTAÇÃO

A.1 Resumo

Este trabalho tem o objetivo de formar um *cópus* de *tweets* sobre comentários de programas televisivos, anotados quanto a sua polaridade, ou seja, se transmitem opinião negativa, positiva ou neutra. A metodologia de anotação desse *cópus* prevê uma parcela anotada manualmente e uma parcela anotada automaticamente por meio de classificadores baseados em Aprendizado de Máquina.

Este documento descreve as regras de anotação manual de um trecho do *cópus*, bem como ilustra, com exemplos do *corpus*, alguns conceitos fundamentais para a anotação e alguns fenômenos observados empiricamente nos documentos coletados.

A.2 Origem dos dados

Os dados do *corpus* foram extraídos da rede social Twitter usando como palavras-chaves algumas *hashtags* de programas televisivos em exibição. *Tweets* de usuários “verificados”, ou seja, que representam uma entidade (celebridades, programas, emissoras) foram removidos, assim como RTs (*retweets*), que são compartilhamentos de *tweets* já publicados, para evitar enviesamento e replicação de documentos.

Nesta etapa as entidades foram mantidas, assim como as *hashtags* e os usuários das postagens. O anotador se compromete a não divulgar esses dados, mantendo o sigilo e a privacidade dos usuários da rede social.

A.3 Tarefa

A tarefa consiste na anotação de blocos de *tweets*, separados em "Fases". As fases são sequenciais, de modo que apenas quando o anotador tiver anotado todos os *tweets* de uma fase é que ele terá acesso ao bloco da fase subsequente (no entanto, é permitido voltar a uma fase anterior para eventuais correções).

A Fase 1 compreende um bloco comum a todos os anotadores, que servirá para o cálculo de concordância. As demais são distintas para os diferentes anotadores.

Em cada etapa, e para cada *tweet*, o anotador deve escolher necessariamente entre 3 opções - **Negativo**, **Neutro** ou **Positivo** - dependendo de sua percepção quanto à polaridade do sentimento do autor, subjacente ao texto apresentado. Apesar de envolver subjetividade, ambiguidade, dúvidas decorrente do vocabulário, da escassez de contexto, etc., o anotador deve **necessariamente** escolher uma das opções.

Em caso de se sentir completamente dividido entre mais de uma opção, deve escolher uma delas e também assinalar o quadro "**Não tenho certeza**", que fica abaixo das 3 opções principais. Apenas nesse caso é que esse quadro deve ser assinalado, ou seja, será natural ter dúvidas entre mais de uma opção, mas na maioria dos casos, haverá uma tendência para uma delas. Nesses casos, não é preciso informar a falta de certeza. Por exemplo, ao anotar "Pra mim #HellsKitchenBR é pau a pau com #MasterChefBr bora fazer um disputa conjunta!", perceba-se que há uma opinião implícita sobre os programas, que, para um anotador, poderia tanto ser positiva quanto negativa. Nesse caso, ele deve escolher uma delas e assinalar "**Não tenho certeza**". O conjunto de casos assinalados dessa forma poderão ser úteis para nos esclarecer sobre eventuais erros dos sistemas automáticos de anotação.

A **classe Positiva** é caracterizada por textos que descrevem sentimentos de prazer, satisfação, elogio, recomendação. Nesse domínio, tweets positivos podem se referir ao **programa** ou a **qualquer outra entidade do domínio do programa** (apresentador, atração, quadro, música, audiência, etc.). Comparações positivas com outros programas ("esse show é muito melhor que aquele") ou com edições anteriores ("hoje o programa está muito melhor que ontem") também refletem positividade. Expressões factuais eventualmente têm polaridade, como em ("o programa começou mais cedo, vou poder assistir inteiro"), que indica uma satisfação do autor. O uso de emoticons também ajudam a atribuir polaridade positiva, como em "Essas músicas".

Uma pergunta que talvez ajude o anotador quanto a esta classe é:

“O tweet reflete uma intenção do autor em mostrar um sentimento positivo?”

A **classe Negativa** é o oposto da anterior, ou seja, deve ser atribuída aos tweets cujos textos descrevem sentimentos negativos, de insatisfação, discordância ou desgosto pelo **programa** ou por **qualquer outra entidade do domínio do programa** (apresentador, atração, quadro, música, audiência, etc.). Negatividade pode ser explícita ("#altashoras de hoje ta bem ruim hein..

?"), ou pode estar implícita na forma de uma sugestão (“poderiam melhorar esse apresentador, né?”), na insatisfação com alguma atração (“esse quadro é horroroso”) ou na discordância de algo referente ao programa (“por que continuam falando disso? Já deu desse assunto!”). Informações factuais também podem carregar polaridade negativa, como em (“entro no twitter e recebo um spoiler de #HellsKitchenBR no meio da fuça.... –”).

Uma pergunta que talvez ajude o anotador quanto a esta classe é:

“O tweet reflete uma intenção do autor em mostrar um sentimento negativo?”

A classe **Neutra** deve ser escolhida sempre que o anotador não conseguir identificar no texto uma opinião (positiva ou negativa) explícita ou implícita. Incluem as expressões factuais típicas, que não carregam polaridade, como em (“Começou o programa.”), opiniões imprecisas, como em (“não sei o que pensar desse cantor”), comentários aos quais não é possível atribuir polaridade, com em (“além do que foi dito no programa, gostaria de adicionar Y e Z”) e (“cheguei em casa e está passando o programa, vou assistir”).

Uma sentença que talvez ajude o anotador quanto a esta classe pode ser:

“O tweet não reflete nem sentimento positivo, nem sentimento negativo do autor.”

A.4 Exemplos

• Classe positiva

- *Gostei muito desse óculos do @lopesjoca tá lindo com cara de intelectual ? #VideoShowAoVivo*
- *Essa mulher que faz a voz da siri e do google tradutor é mó linda #TheNoite*
- *Eu amo as músicas do Luan, na moral #altashoras*
- *um dos melhores comediantes do brasil, tirullipa!! #TheNoite*
- *Luan e Justin, 2 cantores pra quem já torci o nariz e hj canto em voz alta: faxinando, no banheiro, no carro... #LuanNoAltasHoras #altashoras*
- *Cara, só tou dando risada boa hoje no programa #Altashoras xD*
- *#HellsKitchenBR Amando o programa de hoje , só comida boa.*
- *#AltasHoras está maravilhoso com @danielamercury*

• Classe negativa

- *Espero que o #DomingoLegal se reinvente e volte melhor depois dessa chuva de reprise. Mistura jornalismo e entretenimento que dá uma virada.*
- *Todo mundo começou a rir quando Luan disse que tinha ido pro Paraguai, incrível como brasileiro é baba ovo da america do norte #altashoras*

- *Depois que a @flora_reginatto saiu esse programa perdeu toda a graça que tinha, não assisto mais!! #HellsKitchenBR*
- *Nunca fiquei tão bravo numa eliminação quanto hoje. Mas fazer o que, né? #HellsKitchenBR*
- *Falando sério, essa chef me irrita as vezes haha #HellsKitchenBR*
- *Porque o @oserginho grita tanto? #altashoras*
- *#MasterChefBR CADE O SOMMMMM????*
- *Mano Isso ai tá muito errado #HellsKitchenBR*
- *O programa tá religioso Jesus é Abraão . Socorro #AltasHoras*

● **Classe neutra**

- *Rio de Janeiro || 14h08 #VídeoShowAoVivo 16.5, #BalançoGeralRJ 10.1, #ClubeDoChaves 9.1*
- *Daqui a pouco, logo após #TheNoite fique bem informado durante toda a madrugada com o #SBTNoticias, continue no SBT*
- *#HellsKitchenBR vi essa tag achei que era de demolidor*
- *eu e minha mãe ficamos imitando o jacquin fazendo o barulhinho com a língua quando ele tá provando a comida #MasterChefBR*
- *ela ta muito diferente..fico impressionada #TheNoite*
- *Vamos acorda esse prédio!!!! #AltasHoras @luansantana*
- *vamo preparando o core pra ver quem vai sentar na graxa #MasterChefBR*
- *#Thenoite Do crente ao ateu ninguém explica Deus*
- *Hoje a comida deve estar boa... Até agora ninguém reclamou de nada #HellsKitchenBR*
- *Gabriel Jesus no #AltasHoras*

TELAS DA INTERFACE DE ANOTAÇÃO

B.1 Tela de *login*

Cada usuário recebeu um *login* e uma senha única para a utilização do sistema. Isso garantiu a individualidade de cada anotação, tornando possível a avaliação de concordância. A [Figura 25](#) mostra o *design* da tela de *login* da interface.

Figura 25 – Tela de *login* da interface de anotação.

A imagem mostra a interface de login de um sistema web. No topo, há uma barra de navegação escura com os links "Anotação de Sentimentos", "Home" e "Manual de anotação". O corpo principal da página tem um fundo cinza claro e contém o seguinte formulário:

- O título "Please sign in" está centralizado.
- Abaixo dele, há dois campos de entrada brancos: o primeiro contém o texto "guest" e o segundo contém pontos para mascarar a senha.
- Logo abaixo dos campos, há uma opção "Remember me" com uma caixa de seleção desmarcada.
- No final, há um botão azul com o texto "Sign in" em branco.

Fonte: Elaborada pelo autor.

B.2 Painel principal

O painel principal (Figura 26) apresenta as fases de anotação. Inicialmente, cada anotador só pôde visualizar a primeira fase de anotação e à medida que fosse terminando cada fase, a próxima se tornava visível. Utilizamos a barra de progresso para tornar o processo o mais transparente possível para o anotador. Cada fase corresponde a 500 *tweets*, com as exceções da primeira fase, composta somente de 300, e da última, com 328.

Figura 26 – Painel principal da interface de anotação.

Fonte: Elaborada pelo autor.

Os *tweets* contidos na primeira fase do processo foram os mesmos para todos os anotadores. Essa fase foi usada para o cálculo da concordância entre anotadores. As demais fases eram individuais para cada um, sendo recomendado que cada anotador encerrasse cada fase no período de uma semana. A média diária sugerida foi de 100 *tweets*, resultando em uma fase por semana. O objetivo foi que cada anotador anotasse 2.300 *tweets* durante as 5 semanas de anotação, mas decidimos incluir mais alguns na última fase para eventuais problemas com balanceamento de classes.

B.3 Manual de anotação

Através do menu superior da interface, cada anotador podia conferir o manual de anotação (Figura 27). Essa opção é importante para que os anotadores possam sanar dúvidas durante a anotação, revisar regras antes de iniciar ou continuar uma fase ou para revisar as diretrizes de anotação.

O [Apêndice A](#) contém o manual de anotação. Ele possui as diretrizes de etiquetamento das três classes, assim como exemplos e o contato do supervisor da anotação para o caso de dúvidas ou *bugs* na interface.

Figura 27 – Manual de anotação da interface de anotação.

Fonte: Elaborada pelo autor.

B.4 Ambiente de anotação

O ambiente de anotação ([Figura 28](#)) é acessado sempre que um anotador inicia ou continua uma fase por meio do painel principal. Nele estão apresentados todos os *tweets* referentes à fase e as três opções de etiquetas.

Cada *tweet* deve ser anotado com uma das três opções, podendo ou não ser marcada em conjunto com a opção “Não tenho certeza”. Apesar da opção de indecisão, cada anotador foi instruído a marcar a opção que mais lhe parecesse adequada. O ambiente foi dividido em páginas com 50 documentos cada, a fim de não causar cansaço nos anotadores, porém o sistema só avançaria uma página caso o anotador anotasse todos os documentos da página atual.

Após anotado, o *tweet* muda de cor, dependendo da classe escolhida: verde, se positiva; vermelha, se negativa; cinza, se neutra [Figura 29](#). Dessa maneira o anotador pode conferir sua anotação, além de conseguir localizar documentos não anotados na página.

Na lateral do ambiente, adicionamos um resumo do manual com dicas rápidas de anotação: uma definição sucinta de cada classe, um lembrete sobre a anotação de todos os documentos da página e da marcação da opção “Não tenho certeza”, bem como o conjunto de *hashtags* sendo avaliadas no projeto. Um link para o manual também estava presente para rápido acesso dos anotadores.

Figura 28 – Primeiro acesso ao ambiente de anotação.

Fonte: Elaborada pelo autor.

Figura 29 – Documentos anotados na interface de anotação.

Fonte: Elaborada pelo autor.

B.4.1 Painel de estatísticas

O painel de estatísticas (Figura 30) foi adicionado à interface para que o anotador tivesse a opção de acompanhar seu desempenho. Nele cada anotador pode conferir o número de *tweets* anotados e restantes, assim como as porcentagens de positivos, neutros e negativos que foram por ele anotados.

Figura 30 – Painel de estatísticas da interface de anotação.

Fonte: Elaborada pelo autor.

TWEETS COM EMPATE NA ANOTAÇÃO

Abaixo estão listados os *tweets* que obtiveram empate na anotação manual. Os primeiros 17 itens da lista ocorreram durante a primeira etapa da anotação (onde os 7 anotadores etiquetaram) e os demais são oriundos da revisão de anotação (com três anotadores).

Os 47 *tweets* dessa lista foram descartados dos conjuntos de treino e teste nos experimentos, porém estão disponíveis no TTsBR devidamente identificados. Os documentos estão pré-processados, portanto as *hashtags* de programas foram removidas e os identificadores dos usuários foram substituídos por *USERNAME*, assim como números por *NUMBER*.

C.1 Durante a fase de concordância (7 anotadores)

Empatados entre positivo e negativo

"como é que dorme agora depois destas fortes emoções do"
"eu acho que sou um cara meio frustrado por não ter conhecido
a hebe camargo sempre fui fã tô emocionado"

Empatados entre positivo e neutro

"esse vestido de casamento da cibebe deu o que falar kkk"
"essa músicakkk"
"aiii meu coração"
"em casa era NUMBER filhos mais meu pai e minha mãe é maravilhoso
imagina quando cada um" "trazia seus amigos para casa kkk"
"tensão e emoção essa é a cozinha do #OutraAmazônia"
"USERNAME no a gente só espera que cante boa sorte"
"kkk ator da globo"

Empatados entre negativo e neutro

"O xande de pilares tinha q começar ou encerrar o com malandro é malandro e mané é mané"

"essa ana maria braga tá chapadona hj kkk"

"perdi #DiegoDarksonMc no vejo depois no USERNAME"

"manda o vídeo pro céu quem sabe lá tem wi-fi"

"LUCERO EU TÔ TREMENDO LUCERO"

"SAL EM VEZ DE AÇÚCAR NAMARIA SOU EUUU"

"E essa camisa ninguém vai comentar"

"estou aqui só ver se consigo acompanhar o hoje mas estou tão cansada vamos ver"

C.2 Durante a fase de revisão (3 anotadores)

"pois ééé acabei de ver um vídeo de um parto e cara é um acontecimento único e especial é mas q é bizarro é simmm"

"poderia lembrar o joão da silva com nelson xavier marcou minha infância sdd"

"se o episódio de hj for igual ao da semana passada já tô tenso aqui (e com fome)"

"higiene mental é a palavra-chave"

"mentira que eles vão passar isso A bruna tava demais"

"to caindo de sono vai até que horas hoje USERNAME "

"quero saber do vitor b só isso"

"EMPODERAMENTO KKK"

"aí meo deus !"

"USERNAME USERNAME só digo isso agora quero repescagem para ontem !"

"poderiam tirar o temer"

"vanessa da matta me representa jamais faria um parto desses da bela gil horripilante"

"QUERO ME PERDER NO MUNDO"

"minha mãe definitivamente é a rainha do spoiler do"

"eu não consigo falar um eu te amo"

"MEU DEUS DO CÉU PAOLA CAROSELLA"

"USERNAME ensinandu o cuiabanes é a coisa mais linda que oces vão ver hodje seus critinu"

"USERNAME tão lindo mas tem língua plesa bora consultar um fono

seu lindo"

"vermelho tá melhor mais organizado e não desesperados"

"USERNAME não queime esses dedos de fada pelo amor de deus"

"sempre q vejo o pedro bial lembro dele no bbb q saudade"

"aguentando a terça-feira só por causa do ."

"cecília de hábito ? ai papai ! USERNAME"

"fé no pai que hj a miriam sai do"

"USERNAME / a rainha das indiretas soltou uma agora p USERNAME

q ish quem vai no programa dela tem q comer o q n gosta"

"mds tenho escola de manhã e ainda to aki tbm quem manda ser tão emocionante"

"mulheres que choram vendo masterchef"

"nelson xavier um ícone na televisão brasileira descanse em paz"

"encontraram uma namorada pro danilo palmitão & vera verão

#AmorEterno USERNAME USERNAME USERNAME"

"chocado com elas cantando elis regina"

RESULTADOS DO *FRAMEWORK* DE EXPANSÃO COM TTSBR COMPLETO

Tabela 37 – *Self-training* adicionando 40% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	73,2	48,3	61,2	60,90
	Naive Bayes	70,4	48,2	58,8	59,12	71,2	44,9	59,5	58,52
	Regressão Log.	76,6	51,7	66,3	64,87	73,9	48,4	61,4	61,21
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,3	47,9	60,6	60,61
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,2	41,3	54,1	55,19
	Random Forest	74,9	43,2	68,4	62,18	75,0	41,1	67,0	61,05
Média:						72,8	45,3	60,6	59,58
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	71,1	39,2	60,1	56,79
	Naive Bayes	70,4	48,2	58,8	59,12	68,2	38,6	56,7	54,50
	Regressão Log.	76,6	51,7	66,3	64,87	71,1	38,8	61,0	56,97
	Multilayer Perc.	76,3	51,9	65,6	64,60	70,6	40,7	60,1	57,15
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,6	33,9	55,1	52,16
	Random Forest	74,9	43,2	68,4	62,18	70,4	30,1	56,2	52,22
Média:						69,8	36,9	58,2	54,97
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	75,5	47,4	63,5	62,12
	Naive Bayes	70,4	48,2	58,8	59,12	70,9	41	57,7	56,51
	Regressão Log.	76,6	51,7	66,3	64,87	75,7	48	64,2	62,62
	Multilayer Perc.	76,3	51,9	65,6	64,60	75,5	48,3	64,2	62,67
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,6	37,5	58,0	55,38
	Random Forest	74,9	43,2	68,4	62,18	75,3	38,1	67,1	60,14
Média:						73,9	43,4	62,5	59,91
MLP	SVM Linear	73,3	47,2	62,2	60,89	75,2	52,9	64,6	64,25
	Naive Bayes	70,4	48,2	58,8	59,12	70,9	46,4	58,3	58,50
	Regressão Log.	76,6	51,7	66,3	64,87	75,4	53,7	64,9	64,65
	Multilayer Perc.	76,3	51,9	65,6	64,60	75,5	52,8	65,0	64,43
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,8	45,5	56,6	57,29
	Random Forest	74,9	43,2	68,4	62,18	74,1	51,4	65,7	63,74
Média:						73,5	50,5	62,5	62,14
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	70,9	45,9	61,8	59,52
	Naive Bayes	70,4	48,2	58,8	59,12	69,1	46,2	55,9	57,08
	Regressão Log.	76,6	51,7	66,3	64,87	71,0	47,5	62,4	60,32
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,0	43,4	58,8	56,73
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,5	44,6	51,4	53,82
	Random Forest	74,9	43,2	68,4	62,18	70,4	45,8	62,6	59,60
Média:						69,2	45,6	58,8	57,85
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	71,3	47,4	59,6	59,41
	Naive Bayes	70,4	48,2	58,8	59,12	67,6	44,7	52,0	54,76
	Regressão Log.	76,6	51,7	66,3	64,87	71,2	46,6	60,2	59,33
	Multilayer Perc.	76,3	51,9	65,6	64,60	71,4	47,1	61,7	60,06
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,1	43,7	57,9	56,90
	Random Forest	74,9	43,2	68,4	62,18	68,8	41,5	57,4	55,88
Média:						69,9	45,2	58,1	57,72

Fonte: Elaborada pelo autor.

Tabela 38 – *Self-training* adicionando 30% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	73,1	45,5	60,9	59,81
	Naive Bayes	70,4	48,2	58,8	59,12	70,0	41,2	59,3	56,83
	Regressão Log.	76,6	51,7	66,3	64,87	74,1	46,2	61,5	60,58
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,1	45,1	60,8	59,66
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,0	39,6	57,8	55,78
	Random Forest	74,9	43,2	68,4	62,18	75,3	37,7	66,2	59,71
Média:						72,6	42,6	61,1	58,73
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	70,6	35,5	59,2	55,11
	Naive Bayes	70,4	48,2	58,8	59,12	68,1	37,0	56,7	53,92
	Regressão Log.	76,6	51,7	66,3	64,87	70,7	35,1	59,1	54,98
	Multilayer Perc.	76,3	51,9	65,6	64,60	70,6	38,7	59,1	56,13
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,9	32	53,4	51,08
	Random Forest	74,9	43,2	68,4	62,18	69,5	26,4	56,8	50,92
Média:						69,6	34,1	57,4	53,69
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,9	44,8	62,9	60,84
	Naive Bayes	70,4	48,2	58,8	59,12	70,8	38,5	57,7	55,65
	Regressão Log.	76,6	51,7	66,3	64,87	74,8	45,3	62,9	61,01
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,4	45	63,4	60,93
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,9	35,8	60,1	55,22
	Random Forest	74,9	43,2	68,4	62,18	74,4	29,8	66,4	56,83
Média:						73,2	39,9	62,2	58,41
MLP	SVM Linear	73,3	47,2	62,2	60,89	77,2	49,2	66,2	64,20
	Naive Bayes	70,4	48,2	58,8	59,12	70,9	46,3	60,0	59,07
	Regressão Log.	76,6	51,7	66,3	64,87	77,1	48,9	66,7	64,22
	Multilayer Perc.	76,3	51,9	65,6	64,60	76,8	48,9	66,9	64,17
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,5	40,5	60,7	57,22
	Random Forest	74,9	43,2	68,4	62,18	75,0	40,1	68,1	61,03
Média:						74,6	45,7	64,8	61,65
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	72,6	44,4	62,1	59,70
	Naive Bayes	70,4	48,2	58,8	59,12	68,2	47	56,3	57,14
	Regressão Log.	76,6	51,7	66,3	64,87	72,7	45,6	62,3	60,19
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,9	44,5	57,1	56,83
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,9	44,6	50,7	53,72
	Random Forest	74,9	43,2	68,4	62,18	69,8	43,9	59,2	57,63
Média:						69,7	45,0	58,0	57,54
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	70,8	44,3	57,0	57,37
	Naive Bayes	70,4	48,2	58,8	59,12	66,9	43	50,2	53,36
	Regressão Log.	76,6	51,7	66,3	64,87	70,6	44,2	57,2	57,32
	Multilayer Perc.	76,3	51,9	65,6	64,60	70,8	45,3	61,0	59,03
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,3	39,9	54,2	53,80
	Random Forest	74,9	43,2	68,4	62,18	68,7	40,7	55,8	55,06
Média:						69,2	42,9	55,9	55,99

Fonte: Elaborada pelo autor.

Tabela 39 – *Self-training* adicionando 25% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	74,3	46,5	62,2	61,01
	Naive Bayes	70,4	48,2	58,8	59,12	70,6	40,5	60,3	57,12
	Regressão Log.	76,6	51,7	66,3	64,87	74,8	46,9	62,8	61,48
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,5	47,7	61,4	61,18
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,3	40,9	57,2	56,15
	Random Forest	74,9	43,2	68,4	62,18	75,2	36,6	67,8	59,86
Média:						73,3	43,2	62,0	59,47
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	70,3	34,3	59,4	54,66
	Naive Bayes	70,4	48,2	58,8	59,12	68,1	34,5	56,4	52,99
	Regressão Log.	76,6	51,7	66,3	64,87	70,5	33,8	59,6	54,61
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,7	36,9	58,5	55,04
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,7	28,8	52,7	49,05
	Random Forest	74,9	43,2	68,4	62,18	68,7	24,5	56,8	50,00
Média:						68,8	32,1	57,2	52,73
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,8	43,7	62,9	60,48
	Naive Bayes	70,4	48,2	58,8	59,12	70,5	34,8	57,1	54,14
	Regressão Log.	76,6	51,7	66,3	64,87	74,9	44,1	63,4	60,78
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,6	44,4	63,0	60,68
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,5	38,8	60,2	56,48
	Random Forest	74,9	43,2	68,4	62,18	74,4	28,8	65,2	56,14
Média:						73,3	39,1	62,0	58,12
MLP	SVM Linear	73,3	47,2	62,2	60,89	74,8	50,9	64,1	63,25
	Naive Bayes	70,4	48,2	58,8	59,12	70,2	46,2	57,3	57,88
	Regressão Log.	76,6	51,7	66,3	64,87	75,6	51,8	65,0	64,11
	Multilayer Perc.	76,3	51,9	65,6	64,60	75,9	51,4	65,2	64,15
	Árvore de Decisão	67,8	40,3	55,4	54,50	71,7	45,3	57,4	58,15
	Random Forest	74,9	43,2	68,4	62,18	75,1	47,7	65,8	62,87
Média:						73,9	48,9	62,5	61,74
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	72,6	48,3	63,1	61,32
	Naive Bayes	70,4	48,2	58,8	59,12	69,1	49,4	56,5	58,32
	Regressão Log.	76,6	51,7	66,3	64,87	72,6	49,5	63,1	61,73
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,8	43,1	55,9	56,29
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,4	43,4	52,0	53,60
	Random Forest	74,9	43,2	68,4	62,18	71,1	44,9	61,2	59,09
Média:						70,1	46,4	58,6	58,39
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	70,1	42,5	54,6	55,75
	Naive Bayes	70,4	48,2	58,8	59,12	66,5	41,6	49,2	52,40
	Regressão Log.	76,6	51,7	66,3	64,87	70,3	42,6	55,3	56,05
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,8	41,6	57,4	55,92
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,3	37,6	53,2	52,72
	Random Forest	74,9	43,2	68,4	62,18	68,1	37,3	54,0	53,15
Média:						68,5	40,5	54,0	54,33

Fonte: Elaborada pelo autor.

Tabela 40 – *Self-training* adicionando 20% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	74,2	43,9	62,5	60,23
	Naive Bayes	70,4	48,2	58,8	59,12	70,5	37,7	58,7	55,63
	Regressão Log.	76,6	51,7	66,3	64,87	74,5	44,4	62,5	60,47
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,7	46,5	63,4	61,56
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,8	38,8	56,7	55,42
	Random Forest	74,9	43,2	68,4	62,18	75,1	35,1	67,0	59,07
Média:						73,3	41,1	61,8	58,73
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	69,4	33,3	59,4	54,03
	Naive Bayes	70,4	48,2	58,8	59,12	68,1	33,2	56,4	52,53
	Regressão Log.	76,6	51,7	66,3	64,87	69,6	32,9	58,9	53,76
	Multilayer Perc.	76,3	51,9	65,6	64,60	70,1	38,3	58,6	55,64
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,0	27,7	52,9	49,20
	Random Forest	74,9	43,2	68,4	62,18	68,0	23,5	56,4	49,29
Média:						68,7	31,5	57,1	52,41
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,7	41,3	62,3	59,43
	Naive Bayes	70,4	48,2	58,8	59,12	70,5	31	57,8	53,12
	Regressão Log.	76,6	51,7	66,3	64,87	74,9	40,3	63,4	59,49
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,6	41,4	63,8	59,95
	Árvore de Decisão	67,8	40,3	55,4	54,50	71,2	35,7	60,4	55,75
	Random Forest	74,9	43,2	68,4	62,18	73,4	26,0	64,2	54,52
Média:						73,2	36,0	62,0	57,04
MLP	SVM Linear	73,3	47,2	62,2	60,89	75,3	50,8	63,8	63,32
	Naive Bayes	70,4	48,2	58,8	59,12	70,8	46,7	57,6	58,37
	Regressão Log.	76,6	51,7	66,3	64,87	75,5	51,2	64,6	63,75
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,7	49,7	64,5	62,96
	Árvore de Decisão	67,8	40,3	55,4	54,50	71,4	43,8	57,8	57,69
	Random Forest	74,9	43,2	68,4	62,18	74,5	48,3	64,7	62,50
Média:						73,7	48,4	62,2	61,40
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	71,1	42,9	60,7	58,22
	Naive Bayes	70,4	48,2	58,8	59,12	68,3	44,3	54,9	55,82
	Regressão Log.	76,6	51,7	66,3	64,87	71,0	44,8	60,9	58,89
	Multilayer Perc.	76,3	51,9	65,6	64,60	67,1	41,7	54,5	54,42
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,3	41,4	52,3	53,02
	Random Forest	74,9	43,2	68,4	62,18	69,9	43,4	61,5	58,26
Média:						68,8	43,1	57,5	56,44
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	69,6	38,5	53,6	53,89
	Naive Bayes	70,4	48,2	58,8	59,12	67,2	39,8	48,6	51,87
	Regressão Log.	76,6	51,7	66,3	64,87	69,6	38,8	55,3	54,56
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,0	39,2	58,2	55,46
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,3	38,6	52,2	52,70
	Random Forest	74,9	43,2	68,4	62,18	67,8	33,5	52,8	51,36
Média:						68,4	38,1	53,5	53,31

Fonte: Elaborada pelo autor.

Tabela 41 – *Self-training* adicionando 10% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	73,8	38,5	60,6	57,64
	Naive Bayes	70,4	48,2	58,8	59,12	70,3	30,1	58,9	53,12
	Regressão Log.	76,6	51,7	66,3	64,87	73,7	38,9	61,4	57,99
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,3	40,7	61,0	58,32
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,3	33,6	56,1	52,99
	Random Forest	74,9	43,2	68,4	62,18	73,6	24,8	65,8	54,72
Média:						72,3	34,4	60,6	55,80
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	68,8	25,0	58,9	50,90
	Naive Bayes	70,4	48,2	58,8	59,12	66,8	28,2	55,8	50,25
	Regressão Log.	76,6	51,7	66,3	64,87	68,8	26,3	58,8	51,30
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,7	28,7	58,4	52,27
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,8	25,0	54,3	48,34
	Random Forest	74,9	43,2	68,4	62,18	67,4	20,4	56,2	48,01
Média:						67,9	25,6	57,1	50,18
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,3	29,0	63,7	55,68
	Naive Bayes	70,4	48,2	58,8	59,12	70,0	22,2	56,2	49,46
	Regressão Log.	76,6	51,7	66,3	64,87	74,0	28,1	63,0	55,04
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,9	32,8	63,0	56,56
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,8	26,5	59,0	51,76
	Random Forest	74,9	43,2	68,4	62,18	71,7	18,2	62,2	50,72
Média:						72,3	26,1	61,2	53,20
MLP	SVM Linear	73,3	47,2	62,2	60,89	75,1	49,1	64,1	62,77
	Naive Bayes	70,4	48,2	58,8	59,12	70,7	46,2	58,6	58,49
	Regressão Log.	76,6	51,7	66,3	64,87	75,7	50,2	64,5	63,43
	Multilayer Perc.	76,3	51,9	65,6	64,60	75,4	48,9	63,1	62,44
	Árvore de Decisão	67,8	40,3	55,4	54,50	71,7	44,9	59,5	58,68
	Random Forest	74,9	43,2	68,4	62,18	74,4	44,0	65,6	61,31
Média:						73,8	47,2	62,6	61,19
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	71,4	46,9	61,6	59,96
	Naive Bayes	70,4	48,2	58,8	59,12	68,9	46,0	55,0	56,64
	Regressão Log.	76,6	51,7	66,3	64,87	71,6	48,5	61,9	60,66
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,9	47,0	58,6	58,47
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,0	46,3	51,4	54,56
	Random Forest	74,9	43,2	68,4	62,18	70,2	45	63,1	59,44
Média:						69,7	46,6	58,6	58,29
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	68,8	26,1	52,3	49,06
	Naive Bayes	70,4	48,2	58,8	59,12	66,4	33,9	47,0	49,08
	Regressão Log.	76,6	51,7	66,3	64,87	68,9	29,0	53,5	50,44
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,6	32,0	55,1	51,92
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,5	29,2	52,6	49,41
	Random Forest	74,9	43,2	68,4	62,18	67,7	23,5	52,1	47,75
Média:						67,8	29,0	52,1	49,61

Fonte: Elaborada pelo autor.

Tabela 42 – *Self-training* adicionando 5% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	72,9	34,5	60,2	55,87
	Naive Bayes	70,4	48,2	58,8	59,12	70,2	27,0	58,2	51,82
	Regressão Log.	76,6	51,7	66,3	64,87	73,0	32,6	60,8	55,44
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,0	35,2	61,7	56,63
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,0	29,2	59,5	52,90
	Random Forest	74,9	43,2	68,4	62,18	72,5	20,8	63,5	52,26
	Média:					71,9	29,9	60,7	54,15
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	54,5	42,0	55,9	50,79
	Naive Bayes	70,4	48,2	58,8	59,12	51,9	41,0	54,2	49,03
	Regressão Log.	76,6	51,7	66,3	64,87	55,4	41,8	56,4	51,21
	Multilayer Perc.	76,3	51,9	65,6	64,60	55,6	41,2	54,3	50,35
	Árvore de Decisão	67,8	40,3	55,4	54,50	53,5	40,3	51,8	48,52
	Random Forest	74,9	43,2	68,4	62,18	48,6	38,9	52,8	46,77
	Média:					53,3	40,9	54,2	49,45
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	73,4	20,4	62,1	51,96
	Naive Bayes	70,4	48,2	58,8	59,12	69,4	19,8	55,3	48,12
	Regressão Log.	76,6	51,7	66,3	64,87	73,0	21,2	62,2	52,13
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,0	26,9	63,1	54,31
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,6	23,3	57,0	49,96
	Random Forest	74,9	43,2	68,4	62,18	70,4	17,3	58,3	48,65
	Média:					71,5	21,5	59,7	50,86
MLP	SVM Linear	73,3	47,2	62,2	60,89	74,3	49,8	65,1	63,06
	Naive Bayes	70,4	48,2	58,8	59,12	70,0	43,8	58,4	57,37
	Regressão Log.	76,6	51,7	66,3	64,87	74,8	50,8	65,6	63,74
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,0	50,8	64,9	63,24
	Árvore de Decisão	67,8	40,3	55,4	54,50	70,4	44,1	57,5	57,34
	Random Forest	74,9	43,2	68,4	62,18	74,6	43,6	65,9	61,38
	Média:					73,0	47,2	62,9	61,02
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	73,2	49,6	62,0	61,61
	Naive Bayes	70,4	48,2	58,8	59,12	68,3	47,4	54,9	56,87
	Regressão Log.	76,6	51,7	66,3	64,87	73,6	50,5	63,1	62,38
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,3	47,9	58,4	58,53
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,3	42,4	52,7	53,46
	Random Forest	74,9	43,2	68,4	62,18	69,1	44,7	59,8	57,87
	Média:					69,8	47,1	58,5	58,45
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	69,7	23,8	53,4	48,94
	Naive Bayes	70,4	48,2	58,8	59,12	66,9	27,9	47,4	47,37
	Regressão Log.	76,6	51,7	66,3	64,87	69,4	24,0	53,6	49,01
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,1	30,6	55,1	51,59
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,7	30,2	53,3	50,40
	Random Forest	74,9	43,2	68,4	62,18	67,9	23,6	52,7	48,07
	Média:					68,5	26,7	52,6	49,23

Fonte: Elaborada pelo autor.

Tabela 43 – *Self-training* adicionando 1% dos documentos por iteração e avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	72,5	26,3	61,0	53,27
	Naive Bayes	70,4	48,2	58,8	59,12	70,0	23,6	57,8	50,48
	Regressão Log.	76,6	51,7	66,3	64,87	72,2	25,7	61,1	53,02
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,3	29,9	60,8	54,34
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,7	26,5	57,4	51,19
	Random Forest	74,9	43,2	68,4	62,18	71,0	18,7	61,6	50,43
Média:						71,2	25,1	59,9	52,12
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	48,2	42,1	55,4	48,55
	Naive Bayes	70,4	48,2	58,8	59,12	44,5	40,1	53,2	45,92
	Regressão Log.	76,6	51,7	66,3	64,87	49,2	41,7	55,2	48,70
	Multilayer Perc.	76,3	51,9	65,6	64,60	52,6	39,6	53,9	48,69
	Árvore de Decisão	67,8	40,3	55,4	54,50	48,2	38,6	50,3	45,70
	Random Forest	74,9	43,2	68,4	62,18	43,2	39,7	52,1	44,99
Média:						47,6	40,3	53,3	47,09
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	72,0	19,4	59,9	50,41
	Naive Bayes	70,4	48,2	58,8	59,12	68,7	17,3	53,9	46,60
	Regressão Log.	76,6	51,7	66,3	64,87	71,4	18,5	59,8	49,90
	Multilayer Perc.	76,3	51,9	65,6	64,60	71,7	21,7	59,9	51,09
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,3	22,4	56,5	49,03
	Random Forest	74,9	43,2	68,4	62,18	68,6	16,0	52,1	45,56
Média:						70,1	19,2	57,0	48,76
MLP	SVM Linear	73,3	47,2	62,2	60,89	75,0	48,4	65,0	62,79
	Naive Bayes	70,4	48,2	58,8	59,12	70,4	45,2	59,6	58,40
	Regressão Log.	76,6	51,7	66,3	64,87	75,1	48,5	65,7	63,09
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,7	48,1	64,2	62,36
	Árvore de Decisão	67,8	40,3	55,4	54,50	71,5	41,7	61,3	58,18
	Random Forest	74,9	43,2	68,4	62,18	75,3	41,9	67,1	61,42
Média:						73,6	45,6	63,8	61,04
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	72,3	46,5	62,2	60,33
	Naive Bayes	70,4	48,2	58,8	59,12	68,9	45,1	56,6	56,86
	Regressão Log.	76,6	51,7	66,3	64,87	72,8	48,3	63,1	61,40
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,7	43,8	60,5	57,65
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,2	43,0	52,0	53,42
	Random Forest	74,9	43,2	68,4	62,18	69,6	42,7	61,6	57,97
Média:						69,5	44,9	59,3	57,93
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	69,2	21,2	54,0	48,12
	Naive Bayes	70,4	48,2	58,8	59,12	67,0	26,8	48,1	47,30
	Regressão Log.	76,6	51,7	66,3	64,87	69,3	23,7	54,9	49,29
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,1	28,2	57,4	51,55
	Árvore de Decisão	67,8	40,3	55,4	54,50	67,6	27,7	54,3	49,85
	Random Forest	74,9	43,2	68,4	62,18	68,3	22,3	54,2	48,28
Média:						68,4	24,9	53,8	49,06

Fonte: Elaborada pelo autor.

Tabela 44 – *Self-training* adicionando 40% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,6	73,1	78,36
	Naive Bayes	81,4	70,4	75,92	82,0	70,2	76,10
	Regressão Log.	86,4	77,5	81,95	84,0	73,5	78,75
	Multilayer Perc.	87,0	79,2	83,11	84,2	73,7	78,95
	Árvore de Decisão	80,9	70,0	75,43	81,8	70,4	76,09
	Random Forest	74,9	78,4	82,85	86,7	77,2	81,95
Média:					83,72	73,0	78,4
Naive Bayes	SVM Linear	84,4	74,7	79,57	81,4	69,7	75,57
	Naive Bayes	81,4	70,4	75,92	79,2	67,0	73,11
	Regressão Log.	86,4	77,5	81,95	81,2	69,3	75,20
	Multilayer Perc.	87,0	79,2	83,11	81,1	69,8	75,46
	Árvore de Decisão	80,9	70,0	75,43	79,2	66,1	72,65
	Random Forest	74,9	78,4	82,85	80,9	68,2	74,57
Média:					80,50	68,4	74,4
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,5	75,5	80,50
	Naive Bayes	81,4	70,4	75,92	81,5	68,4	74,94
	Regressão Log.	86,4	77,5	81,95	86,0	76,1	81,05
	Multilayer Perc.	87,0	79,2	83,11	85,8	76,3	81,05
	Árvore de Decisão	80,9	70,0	75,43	82,2	70,8	76,53
	Random Forest	74,9	78,4	82,85	86,3	75,9	81,09
Média:					84,55	73,8	79,2
MLP	SVM Linear	84,4	74,7	79,57	86,0	77,3	81,65
	Naive Bayes	81,4	70,4	75,92	82,1	71,3	76,72
	Regressão Log.	86,4	77,5	81,95	86,2	77,5	81,83
	Multilayer Perc.	87,0	79,2	83,11	86,2	77,8	82,01
	Árvore de Decisão	80,9	70,0	75,43	82,6	71,7	77,14
	Random Forest	74,9	78,4	82,85	86,8	78,1	82,45
Média:					84,98	75,6	80,3
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	83,6	72,0	77,79
	Naive Bayes	81,4	70,4	75,92	81,2	69,0	75,13
	Regressão Log.	86,4	77,5	81,95	83,5	72,0	77,76
	Multilayer Perc.	87,0	79,2	83,11	81,1	69,6	75,35
	Árvore de Decisão	80,9	70,0	75,43	79,0	65,9	72,46
	Random Forest	74,9	78,4	82,85	83,5	73,0	78,24
Média:					81,98	70,3	76,1
Random Forest	SVM Linear	84,4	74,7	79,57	84,5	68,5	76,51
	Naive Bayes	81,4	70,4	75,92	81,0	59,6	70,31
	Regressão Log.	86,4	77,5	81,95	84,4	68,7	76,56
	Multilayer Perc.	87,0	79,2	83,11	84,0	69,0	76,45
	Árvore de Decisão	80,9	70,0	75,43	81,9	64,8	73,34
	Random Forest	74,9	78,4	82,85	83,1	64,9	73,98
Média:					83,15	65,9	74,5

Fonte: Elaborada pelo autor.

Tabela 45 – Self-training adicionando 30% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,6	74,2	79,40
	Naive Bayes	81,4	70,4	75,92	81,6	69,8	75,68
	Regressão Log.	86,4	77,5	81,95	84,5	73,7	79,07
	Multilayer Perc.	87,0	79,2	83,11	84,2	74,0	79,14
	Árvore de Decisão	80,9	70,0	75,43	80,7	68,5	74,64
	Random Forest	74,9	78,4	82,85	86,6	76,5	81,56
		Média:			83,70	72,8	78,2
Naive Bayes	SVM Linear	84,4	74,7	79,57	81,2	69,3	75,20
	Naive Bayes	81,4	70,4	75,92	78,9	67,2	73,09
	Regressão Log.	86,4	77,5	81,95	81,6	70,3	75,96
	Multilayer Perc.	87,0	79,2	83,11	81,1	70,0	75,51
	Árvore de Decisão	80,9	70,0	75,43	78,1	65,0	71,52
	Random Forest	74,9	78,4	82,85	79,7	66,8	73,26
		Média:			80,10	68,1	74,1
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,4	75,0	80,17
	Naive Bayes	81,4	70,4	75,92	81,3	67,4	74,35
	Regressão Log.	86,4	77,5	81,95	85,4	75,1	80,25
	Multilayer Perc.	87,0	79,2	83,11	85,0	74,2	79,61
	Árvore de Decisão	80,9	70,0	75,43	82,1	70,5	76,27
	Random Forest	74,9	78,4	82,85	86,1	74,9	80,50
		Média:			84,22	72,9	78,5
MLP	SVM Linear	84,4	74,7	79,57	86,1	78,3	82,21
	Naive Bayes	81,4	70,4	75,92	81,5	71,5	76,50
	Regressão Log.	86,4	77,5	81,95	86,0	78,3	82,17
	Multilayer Perc.	87,0	79,2	83,11	85,2	77,5	81,32
	Árvore de Decisão	80,9	70,0	75,43	83,7	74,7	79,19
	Random Forest	74,9	78,4	82,85	86,8	79,5	83,13
		Média:			84,88	76,6	80,8
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	83,9	73,0	78,46
	Naive Bayes	81,4	70,4	75,92	81,0	67,9	74,44
	Regressão Log.	86,4	77,5	81,95	84,4	73,8	79,10
	Multilayer Perc.	87,0	79,2	83,11	81,5	69,8	75,63
	Árvore de Decisão	80,9	70,0	75,43	77,6	63,6	70,60
	Random Forest	74,9	78,4	82,85	83,1	72,1	77,62
		Média:			81,92	70,0	76,0
Random Forest	SVM Linear	84,4	74,7	79,57	82,9	63,4	73,17
	Naive Bayes	81,4	70,4	75,92	80,5	56,1	68,33
	Regressão Log.	86,4	77,5	81,95	83,3	64,4	73,85
	Multilayer Perc.	87,0	79,2	83,11	83,4	66,1	74,75
	Árvore de Decisão	80,9	70,0	75,43	81,4	61,7	71,54
	Random Forest	74,9	78,4	82,85	82,9	62,5	72,68
		Média:			82,40	62,4	72,4

Fonte: Elaborada pelo autor.

Tabela 46 – *Self-training* adicionando 25% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,6	74,3	79,43
	Naive Bayes	81,4	70,4	75,92	82,1	70,5	76,27
	Regressão Log.	86,4	77,5	81,95	85,1	75,1	80,13
	Multilayer Perc.	87,0	79,2	83,11	85,0	75,6	80,28
	Árvore de Decisão	80,9	70,0	75,43	80,3	67,5	73,91
	Random Forest	74,9	78,4	82,85	86,6	76,3	81,43
Média:					83,95	73,2	78,6
Naive Bayes	SVM Linear	84,4	74,7	79,57	80,5	69,1	74,82
	Naive Bayes	81,4	70,4	75,92	78,4	67,0	72,70
	Regressão Log.	86,4	77,5	81,95	80,5	69,2	74,83
	Multilayer Perc.	87,0	79,2	83,11	80,4	69,5	74,91
	Árvore de Decisão	80,9	70,0	75,43	77,8	65,1	71,41
	Random Forest	74,9	78,4	82,85	79,3	66,9	73,12
Média:					79,48	67,8	73,6
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,1	74,2	79,65
	Naive Bayes	81,4	70,4	75,92	81,7	67,7	74,70
	Regressão Log.	86,4	77,5	81,95	85,4	74,7	80,08
	Multilayer Perc.	87,0	79,2	83,11	85,1	74,3	79,68
	Árvore de Decisão	80,9	70,0	75,43	80,9	68,0	74,46
	Random Forest	74,9	78,4	82,85	85,9	73,9	79,91
Média:					84,02	72,1	78,1
MLP	SVM Linear	84,4	74,7	79,57	85,6	76,1	80,82
	Naive Bayes	81,4	70,4	75,92	81,8	69,9	75,85
	Regressão Log.	86,4	77,5	81,95	85,6	76,0	80,81
	Multilayer Perc.	87,0	79,2	83,11	86,4	77,4	81,88
	Árvore de Decisão	80,9	70,0	75,43	84,5	73,9	79,16
	Random Forest	74,9	78,4	82,85	86,4	77,2	81,82
Média:					85,05	75,1	80,1
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	84,6	73,6	79,07
	Naive Bayes	81,4	70,4	75,92	81,2	68,3	74,72
	Regressão Log.	86,4	77,5	81,95	84,9	74,2	79,53
	Multilayer Perc.	87,0	79,2	83,11	81,3	68,3	74,81
	Árvore de Decisão	80,9	70,0	75,43	77,6	63,7	70,66
	Random Forest	74,9	78,4	82,85	83,9	72,7	78,29
Média:					82,25	70,1	76,2
Random Forest	SVM Linear	84,4	74,7	79,57	82,2	59,5	70,83
	Naive Bayes	81,4	70,4	75,92	80,4	54,4	67,42
	Regressão Log.	86,4	77,5	81,95	82,4	60,9	71,67
	Multilayer Perc.	87,0	79,2	83,11	82,7	63,6	73,17
	Árvore de Decisão	80,9	70,0	75,43	81,3	60,3	70,80
	Random Forest	74,9	78,4	82,85	82,1	59,9	71,04
Média:					81,85	59,8	70,8

Fonte: Elaborada pelo autor.

Tabela 47 – Self-training adicionando 20% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,5	74,4	79,46
	Naive Bayes	81,4	70,4	75,92	82,0	70,0	76,00
	Regressão Log.	86,4	77,5	81,95	84,7	74,3	79,48
	Multilayer Perc.	87,0	79,2	83,11	84,4	74,5	79,47
	Árvore de Decisão	80,9	70,0	75,43	82,5	70,2	76,31
	Random Forest	74,9	78,4	82,85	86,5	75,8	81,15
		Média:			84,10	73,2	78,6
Naive Bayes	SVM Linear	84,4	74,7	79,57	80,0	69,4	74,66
	Naive Bayes	81,4	70,4	75,92	78,0	66,9	72,45
	Regressão Log.	86,4	77,5	81,95	79,6	68,8	74,22
	Multilayer Perc.	87,0	79,2	83,11	79,8	69,0	74,41
	Árvore de Decisão	80,9	70,0	75,43	76,2	63,4	69,79
	Random Forest	74,9	78,4	82,85	78,0	65,4	71,69
		Média:			78,60	67,2	72,9
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,4	74,3	79,86
	Naive Bayes	81,4	70,4	75,92	81,9	67,8	74,87
	Regressão Log.	86,4	77,5	81,95	85,7	74,7	80,19
	Multilayer Perc.	87,0	79,2	83,11	85,6	75,1	80,38
	Árvore de Decisão	80,9	70,0	75,43	82,2	68,3	75,28
	Random Forest	74,9	78,4	82,85	85,2	72,0	78,60
		Média:			84,33	72,0	78,2
MLP	SVM Linear	84,4	74,7	79,57	85,2	75,9	80,54
	Naive Bayes	81,4	70,4	75,92	82,3	70,7	76,50
	Regressão Log.	86,4	77,5	81,95	85,6	76,5	81,03
	Multilayer Perc.	87,0	79,2	83,11	85,5	76,6	81,07
	Árvore de Decisão	80,9	70,0	75,43	84,0	73,7	78,85
	Random Forest	74,9	78,4	82,85	86,8	77,5	82,12
		Média:			84,90	75,2	80,0
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	83,3	71,5	77,38
	Naive Bayes	81,4	70,4	75,92	81,1	68,5	74,80
	Regressão Log.	86,4	77,5	81,95	83,4	72,0	77,70
	Multilayer Perc.	87,0	79,2	83,11	84,0	73,9	78,91
	Árvore de Decisão	80,9	70,0	75,43	78,2	66,4	72,31
	Random Forest	74,9	78,4	82,85	83,5	73,2	78,34
		Média:			82,25	70,9	76,6
Random Forest	SVM Linear	84,4	74,7	79,57	81,9	58,5	70,21
	Naive Bayes	81,4	70,4	75,92	80,2	53,7	66,96
	Regressão Log.	86,4	77,5	81,95	82,4	59,9	71,11
	Multilayer Perc.	87,0	79,2	83,11	82,6	61,7	72,18
	Árvore de Decisão	80,9	70,0	75,43	81,7	61,0	71,35
	Random Forest	74,9	78,4	82,85	82,3	59,5	70,88
		Média:			81,85	59,1	70,4

Fonte: Elaborada pelo autor.

Tabela 48 – *Self-training* adicionando 10% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,1	72,8	78,43
	Naive Bayes	81,4	70,4	75,92	81,9	68,7	75,29
	Regressão Log.	86,4	77,5	81,95	84,6	73,1	78,84
	Multilayer Perc.	87,0	79,2	83,11	84,0	73,1	78,53
	Árvore de Decisão	80,9	70,0	75,43	81,0	68,8	74,88
	Random Forest	74,9	78,4	82,85	85,5	73,0	79,24
Média:					83,52	71,6	77,5
Naive Bayes	SVM Linear	84,4	74,7	79,57	78,2	69,5	73,83
	Naive Bayes	81,4	70,4	75,92	76,5	66,6	71,54
	Regressão Log.	86,4	77,5	81,95	78,4	69,5	73,94
	Multilayer Perc.	87,0	79,2	83,11	78,5	69,7	74,08
	Árvore de Decisão	80,9	70,0	75,43	75,2	65,6	70,38
	Random Forest	74,9	78,4	82,85	76,9	66,7	71,76
Média:					77,28	67,9	72,6
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,3	72,7	78,98
	Naive Bayes	81,4	70,4	75,92	81,5	65,0	73,23
	Regressão Log.	86,4	77,5	81,95	85,6	73,3	79,44
	Multilayer Perc.	87,0	79,2	83,11	85,1	72,3	78,71
	Árvore de Decisão	80,9	70,0	75,43	81,5	65,0	73,23
	Random Forest	74,9	78,4	82,85	85,0	69,2	77,11
Média:					84,00	69,6	76,8
MLP	SVM Linear	84,4	74,7	79,57	85,7	77,0	81,32
	Naive Bayes	81,4	70,4	75,92	81,6	71,2	76,43
	Regressão Log.	86,4	77,5	81,95	86,1	78,0	82,08
	Multilayer Perc.	87,0	79,2	83,11	86,1	77,5	81,79
	Árvore de Decisão	80,9	70,0	75,43	83,1	73,7	78,36
	Random Forest	74,9	78,4	82,85	86,7	78,8	82,76
Média:					84,88	76,0	80,5
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	83,0	69,8	76,40
	Naive Bayes	81,4	70,4	75,92	80,9	67,5	74,16
	Regressão Log.	86,4	77,5	81,95	83,8	70,8	77,26
	Multilayer Perc.	87,0	79,2	83,11	81,0	66,8	73,89
	Árvore de Decisão	80,9	70,0	75,43	79,7	66,1	72,86
	Random Forest	74,9	78,4	82,85	83,7	72,3	77,97
Média:					82,02	68,9	75,4
Random Forest	SVM Linear	84,4	74,7	79,57	81,9	57,0	69,45
	Naive Bayes	81,4	70,4	75,92	79,8	52,1	65,98
	Regressão Log.	86,4	77,5	81,95	82,1	58,1	70,09
	Multilayer Perc.	87,0	79,2	83,11	82,3	60,1	71,22
	Árvore de Decisão	80,9	70,0	75,43	81,5	58,8	70,17
	Random Forest	74,9	78,4	82,85	81,8	57,6	69,71
Média:					81,57	57,3	69,4

Fonte: Elaborada pelo autor.

Tabela 49 – *Self-training* adicionando 5% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,7	71,1	77,36
	Naive Bayes	81,4	70,4	75,92	81,8	67,8	74,83
	Regressão Log.	86,4	77,5	81,95	84,0	71,7	77,86
	Multilayer Perc.	87,0	79,2	83,11	83,6	71,7	77,64
	Árvore de Decisão	80,9	70,0	75,43	80,0	64,6	72,28
	Random Forest	74,9	78,4	82,85	85,5	71,9	78,67
		Média:			83,10	69,8	76,4
Naive Bayes	SVM Linear	84,4	74,7	79,57	67,1	67,4	67,26
	Naive Bayes	81,4	70,4	75,92	67,1	65,9	66,52
	Regressão Log.	86,4	77,5	81,95	68,4	68,3	68,33
	Multilayer Perc.	87,0	79,2	83,11	70,6	67,8	69,21
	Árvore de Decisão	80,9	70,0	75,43	67,1	65,9	66,52
	Random Forest	74,9	78,4	82,85	65,0	65,8	65,39
		Média:			67,55	66,9	67,2
Regressão Logística	SVM Linear	84,4	74,7	79,57	85,6	72,7	79,12
	Naive Bayes	81,4	70,4	75,92	81,2	62,7	71,98
	Regressão Log.	86,4	77,5	81,95	85,0	71,0	78,01
	Multilayer Perc.	87,0	79,2	83,11	85,7	72,2	78,97
	Árvore de Decisão	80,9	70,0	75,43	81,9	63,8	72,82
	Random Forest	74,9	78,4	82,85	84,2	65,0	74,61
		Média:			83,93	67,9	75,9
MLP	SVM Linear	84,4	74,7	79,57	85,5	76,6	81,07
	Naive Bayes	81,4	70,4	75,92	81,4	69,9	75,62
	Regressão Log.	86,4	77,5	81,95	85,6	76,9	81,25
	Multilayer Perc.	87,0	79,2	83,11	85,2	76,5	80,81
	Árvore de Decisão	80,9	70,0	75,43	82,0	70,5	76,24
	Random Forest	74,9	78,4	82,85	86,4	77,2	81,83
		Média:			84,35	74,6	79,5
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	84,6	73,6	79,13
	Naive Bayes	81,4	70,4	75,92	80,9	67,6	74,26
	Regressão Log.	86,4	77,5	81,95	85,1	73,8	79,45
	Multilayer Perc.	87,0	79,2	83,11	83,8	72,6	78,19
	Árvore de Decisão	80,9	70,0	75,43	78,8	65,9	72,31
	Random Forest	74,9	78,4	82,85	84,0	72,2	78,11
		Média:			82,87	71,0	76,9
Random Forest	SVM Linear	84,4	74,7	79,57	82,3	58,5	70,41
	Naive Bayes	81,4	70,4	75,92	79,9	52,3	66,08
	Regressão Log.	86,4	77,5	81,95	82,3	58,6	70,44
	Multilayer Perc.	87,0	79,2	83,11	82,3	60,4	71,32
	Árvore de Decisão	80,9	70,0	75,43	81,7	59,9	70,78
	Random Forest	74,9	78,4	82,85	82,1	58,7	70,4
		Média:			81,77	58,1	69,9

Fonte: Elaborada pelo autor.

Tabela 50 – *Self-training* adicionando 1% dos documentos por iteração na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,0	70,8	77,38
	Naive Bayes	81,4	70,4	75,92	81,9	66,2	74,00
	Regressão Log.	86,4	77,5	81,95	84,3	71,2	77,71
	Multilayer Perc.	87,0	79,2	83,11	83,9	70,4	77,13
	Árvore de Decisão	80,9	70,0	75,43	80,0	63,1	71,57
	Random Forest	74,9	78,4	82,85	84,1	66,7	75,37
Média:					83,03	68,1	75,5
Naive Bayes	SVM Linear	84,4	74,7	79,57	59,2	64,4	61,80
	Naive Bayes	81,4	70,4	75,92	59,8	63,7	61,74
	Regressão Log.	86,4	77,5	81,95	60,7	65,5	63,09
	Multilayer Perc.	87,0	79,2	83,11	64,5	65,8	65,18
	Árvore de Decisão	80,9	70,0	75,43	64,9	65,9	65,38
	Random Forest	74,9	78,4	82,85	60,7	64,9	62,80
Média:					61,63	65,0	63,3
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,2	67,6	75,92
	Naive Bayes	81,4	70,4	75,92	81,2	60,2	70,71
	Regressão Log.	86,4	77,5	81,95	84,1	67,2	75,65
	Multilayer Perc.	87,0	79,2	83,11	84,1	67,7	75,90
	Árvore de Decisão	80,9	70,0	75,43	81,7	60,3	71,00
	Random Forest	74,9	78,4	82,85	82,2	56,1	69,12
Média:					82,92	63,2	73,1
MLP	SVM Linear	84,4	74,7	79,57	85,8	77,7	81,76
	Naive Bayes	81,4	70,4	75,92	81,8	70,9	76,32
	Regressão Log.	86,4	77,5	81,95	85,7	77,6	81,62
	Multilayer Perc.	87,0	79,2	83,11	85,6	77,7	81,68
	Árvore de Decisão	80,9	70,0	75,43	82,4	72,4	77,41
	Random Forest	74,9	78,4	82,85	86,5	78,3	82,39
Média:					84,63	75,8	80,2
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	83,6	72,5	78,03
	Naive Bayes	81,4	70,4	75,92	80,5	67,7	74,10
	Regressão Log.	86,4	77,5	81,95	83,9	72,8	78,37
	Multilayer Perc.	87,0	79,2	83,11	80,6	69,2	74,88
	Árvore de Decisão	80,9	70,0	75,43	79,8	67,4	73,57
	Random Forest	74,9	78,4	82,85	83,5	72,8	78,14
Média:					81,98	70,4	76,2
Random Forest	SVM Linear	84,4	74,7	79,57	82,3	59,1	70,71
	Naive Bayes	81,4	70,4	75,92	80,0	53,5	66,72
	Regressão Log.	86,4	77,5	81,95	82,6	60,5	71,53
	Multilayer Perc.	87,0	79,2	83,11	82,7	62,0	72,33
	Árvore de Decisão	80,9	70,0	75,43	81,8	61,1	71,48
	Random Forest	74,9	78,4	82,85	82,0	59,6	70,78
Média:					81,90	59,3	70,6

Fonte: Elaborada pelo autor.

RESULTADOS DO *FRAMEWORK* DE EXPANSÃO COM TTSBR REDUZIDO

Tabela 51 – *Self-training* adicionando 40% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	71,7	51,6	62,1	61,82
	Naive Bayes	70,4	48,2	58,8	59,12	67,7	49,4	57,9	58,32
	Regressão Log.	76,6	51,7	66,3	64,87	72,1	51,7	61,8	61,87
	Multilayer Perc.	76,3	51,9	65,6	64,60	70,9	50,9	62,3	61,35
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,0	47,2	55,3	57,15
	Random Forest	74,9	43,2	68,4	62,18	72,7	50,6	65,4	62,90
Média:						70,7	50,2	60,8	60,57
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	68,3	47,6	58,8	58,23
	Naive Bayes	70,4	48,2	58,8	59,12	64,8	46,8	55,9	55,83
	Regressão Log.	76,6	51,7	66,3	64,87	68,8	48,4	59,9	59,03
	Multilayer Perc.	76,3	51,9	65,6	64,60	67,7	47,7	58,6	58,00
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,1	44,4	53,9	54,76
	Random Forest	74,9	43,2	68,4	62,18	67,9	45,9	57,6	57,13
Média:						67,3	46,8	57,5	57,16
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,1	53	63,6	63,58
	Naive Bayes	70,4	48,2	58,8	59,12	68,0	49,2	57,7	58,31
	Regressão Log.	76,6	51,7	66,3	64,87	73,7	52,3	63,4	63,11
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,0	52,7	63,3	63,33
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,3	45,5	57,2	57,32
	Random Forest	74,9	43,2	68,4	62,18	72,6	50,8	65,8	63,05
Média:						72,0	50,6	61,8	61,45
MLP	SVM Linear	73,3	47,2	62,2	60,89	72,5	51,6	65,7	63,24
	Naive Bayes	70,4	48,2	58,8	59,12	67,9	50,0	59,2	59,01
	Regressão Log.	76,6	51,7	66,3	64,87	72,9	51,9	66,0	63,60
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,7	52,3	65,9	63,64
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,9	50,9	60,1	59,95
	Random Forest	74,9	43,2	68,4	62,18	72,1	51,9	66,0	63,32
Média:						71,2	51,4	63,8	62,13
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	67,8	49,1	62,0	59,64
	Naive Bayes	70,4	48,2	58,8	59,12	65,8	49,2	55,2	56,73
	Regressão Log.	76,6	51,7	66,3	64,87	68,2	50,6	63,1	60,64
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,4	48,8	58,2	57,78
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,6	48,1	53,2	55,63
	Random Forest	74,9	43,2	68,4	62,18	67,4	49,6	62,3	59,76
Média:						66,9	49,2	59,0	58,36
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	68,3	49,6	63,2	60,35
	Naive Bayes	70,4	48,2	58,8	59,12	64,2	47,3	55,2	55,56
	Regressão Log.	76,6	51,7	66,3	64,87	68,6	50	62,8	60,46
	Multilayer Perc.	76,3	51,9	65,6	64,60	69,0	51,1	64,3	61,47
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,4	47,6	58,5	57,47
	Random Forest	74,9	43,2	68,4	62,18	68,5	50,5	63,3	60,75
Média:						67,5	49,4	61,2	59,34

Fonte: Elaborada pelo autor.

Tabela 52 – *Self-training* adicionando 30% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	71,6	51,3	61,2	61,38
	Naive Bayes	70,4	48,2	58,8	59,12	67,7	50,1	56,9	58,26
	Regressão Log.	76,6	51,7	66,3	64,87	72,2	52,3	62,1	62,19
	Multilayer Perc.	76,3	51,9	65,6	64,60	71,6	51,5	61,8	61,61
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,1	48,1	57,3	57,82
	Random Forest	74,9	43,2	68,4	62,18	73,3	51,9	66,1	63,77
Média:						70,8	50,9	60,9	60,84
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	67,3	46,2	58,0	57,16
	Naive Bayes	70,4	48,2	58,8	59,12	64,0	45,9	55,8	55,23
	Regressão Log.	76,6	51,7	66,3	64,87	67,5	47,2	58,5	57,74
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,7	47,6	58,1	57,46
	Árvore de Decisão	67,8	40,3	55,4	54,50	63,9	42,4	53,7	53,30
	Random Forest	74,9	43,2	68,4	62,18	66,0	44,8	56,0	55,60
Média:						65,9	45,7	56,7	56,08
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	73,5	52,5	64,0	63,35
	Naive Bayes	70,4	48,2	58,8	59,12	67,9	49,7	57,4	58,35
	Regressão Log.	76,6	51,7	66,3	64,87	74,2	52,5	64,4	63,70
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,0	53,5	64,6	64,00
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,8	45,7	58,4	56,97
	Random Forest	74,9	43,2	68,4	62,18	72,9	51,6	67,2	63,89
Média:						71,6	50,9	62,7	61,71
MLP	SVM Linear	73,3	47,2	62,2	60,89	71,0	50,9	62,5	61,44
	Naive Bayes	70,4	48,2	58,8	59,12	67,8	48,4	59,1	58,44
	Regressão Log.	76,6	51,7	66,3	64,87	72,0	50,4	62,8	61,75
	Multilayer Perc.	76,3	51,9	65,6	64,60	71,6	51,5	62,9	61,99
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,2	47,4	56,0	57,18
	Random Forest	74,9	43,2	68,4	62,18	72,0	52	65,2	63,05
Média:						70,4	50,1	61,4	60,64
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	66,7	48,7	61,7	59,06
	Naive Bayes	70,4	48,2	58,8	59,12	66,4	50,0	56,0	57,45
	Regressão Log.	76,6	51,7	66,3	64,87	67,0	49,3	62,3	59,52
	Multilayer Perc.	76,3	51,9	65,6	64,60	65,4	48,4	56,6	56,78
	Árvore de Decisão	67,8	40,3	55,4	54,50	64,0	46,1	52,4	54,14
	Random Forest	74,9	43,2	68,4	62,18	67,2	49,3	61,6	59,37
Média:						66,1	48,6	58,4	57,72
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	67,7	49,0	59,5	58,73
	Naive Bayes	70,4	48,2	58,8	59,12	64,3	47,8	52,2	54,73
	Regressão Log.	76,6	51,7	66,3	64,87	67,9	49,5	60,3	59,26
	Multilayer Perc.	76,3	51,9	65,6	64,60	68,6	48,6	61,4	59,53
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,0	47,8	57,8	57,18
	Random Forest	74,9	43,2	68,4	62,18	68,3	49,1	59,0	58,79
Média:						67,1	48,6	58,4	58,04

Fonte: Elaborada pelo autor.

Tabela 53 – *Self-training* adicionando 25% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	71,8	51,4	62,8	61,99
	Naive Bayes	70,4	48,2	58,8	59,12	67,4	49,5	56,8	57,90
	Regressão Log.	76,6	51,7	66,3	64,87	72,5	51,5	62,7	62,22
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,2	50,6	62,1	61,64
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,4	44,1	57,6	56,66
	Random Forest	74,9	43,2	68,4	62,18	73,2	50,9	67,0	63,71
Média:						70,9	49,7	61,5	60,69
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	65,2	46,4	56,9	56,16
	Naive Bayes	70,4	48,2	58,8	59,12	62,8	44,9	55,3	54,29
	Regressão Log.	76,6	51,7	66,3	64,87	65,6	46,9	57,7	56,74
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,0	46,7	58,1	56,90
	Árvore de Decisão	67,8	40,3	55,4	54,50	62,0	42,5	53,1	52,54
	Random Forest	74,9	43,2	68,4	62,18	64,0	44,5	56,3	54,97
Média:						64,3	45,3	56,2	55,27
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	73,5	53,1	63,6	63,42
	Naive Bayes	70,4	48,2	58,8	59,12	67,8	49,1	57,3	58,09
	Regressão Log.	76,6	51,7	66,3	64,87	73,8	53	63,8	63,54
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,4	51,7	63,3	62,77
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,5	46,1	58,6	57,71
	Random Forest	74,9	43,2	68,4	62,18	73,7	49,5	66,8	63,34
Média:						71,8	50,4	62,2	61,48
MLP	SVM Linear	73,3	47,2	62,2	60,89	71,3	52,0	62,8	62,02
	Naive Bayes	70,4	48,2	58,8	59,12	67,5	50,1	57,8	58,48
	Regressão Log.	76,6	51,7	66,3	64,87	72,3	52,5	62,3	62,40
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,1	52,8	62,2	62,35
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,5	48,0	57,0	57,84
	Random Forest	74,9	43,2	68,4	62,18	72,0	53,6	64,9	63,49
Média:						70,6	51,5	61,2	61,10
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	67,9	49,5	61,0	59,48
	Naive Bayes	70,4	48,2	58,8	59,12	66,3	49,4	56,1	57,27
	Regressão Log.	76,6	51,7	66,3	64,87	67,7	50	61,6	59,76
	Multilayer Perc.	76,3	51,9	65,6	64,60	67,1	49,4	58,4	58,29
	Árvore de Decisão	67,8	40,3	55,4	54,50	63,0	44,3	51,9	53,05
	Random Forest	74,9	43,2	68,4	62,18	67,9	49,8	62,7	60,15
Média:						66,7	48,7	58,6	58,00
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	67,2	48,8	58,4	58,14
	Naive Bayes	70,4	48,2	58,8	59,12	63,6	47,8	52,9	54,77
	Regressão Log.	76,6	51,7	66,3	64,87	67,4	48,9	58,7	58,31
	Multilayer Perc.	76,3	51,9	65,6	64,60	67,8	48	59,0	58,30
	Árvore de Decisão	67,8	40,3	55,4	54,50	66,6	47,8	55,9	56,77
	Random Forest	74,9	43,2	68,4	62,18	66,9	48	57,5	57,44
Média:						66,6	48,2	57,1	57,29

Fonte: Elaborada pelo autor.

Tabela 54 – *Self-training* adicionando 20% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	71,6	51,5	62,3	61,80
	Naive Bayes	70,4	48,2	58,8	59,12	68,0	49,1	56,3	57,81
	Regressão Log.	76,6	51,7	66,3	64,87	72,3	52,6	62,4	62,42
	Multilayer Perc.	76,3	51,9	65,6	64,60	71,7	51	62,9	61,83
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,9	45,7	57,2	57,28
	Random Forest	74,9	43,2	68,4	62,18	73,5	51,1	66,5	63,69
Média:						71,0	50,2	61,3	60,81
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	63,8	45,2	56,4	55,12
	Naive Bayes	70,4	48,2	58,8	59,12	61,4	43,5	55,2	53,38
	Regressão Log.	76,6	51,7	66,3	64,87	64,7	45,8	57,7	56,03
	Multilayer Perc.	76,3	51,9	65,6	64,60	65,3	44,8	57,6	55,89
	Árvore de Decisão	67,8	40,3	55,4	54,50	61,2	41	52,8	51,64
	Random Forest	74,9	43,2	68,4	62,18	62,6	44	55,6	54,08
Média:						63,2	44,1	55,9	54,36
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,1	52,6	63,2	63,29
	Naive Bayes	70,4	48,2	58,8	59,12	69,2	49,5	57,9	58,86
	Regressão Log.	76,6	51,7	66,3	64,87	74,6	51,8	63,9	63,41
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,5	52	63,2	63,21
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,7	46,4	59,1	58,38
	Random Forest	74,9	43,2	68,4	62,18	73,5	47,9	65,2	62,20
Média:						72,6	50,0	62,1	61,55
MLP	SVM Linear	73,3	47,2	62,2	60,89	74,0	52,2	63,7	63,29
	Naive Bayes	70,4	48,2	58,8	59,12	69,0	49,3	58,3	58,86
	Regressão Log.	76,6	51,7	66,3	64,87	74,4	52,9	64,3	63,87
	Multilayer Perc.	76,3	51,9	65,6	64,60	74,3	53	62,2	63,17
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,8	46,5	56,6	57,66
	Random Forest	74,9	43,2	68,4	62,18	73,5	51	65,2	63,23
Média:						72,5	50,8	61,7	61,68
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	67,5	49,7	60,2	59,12
	Naive Bayes	70,4	48,2	58,8	59,12	65,6	49,8	56,6	57,31
	Regressão Log.	76,6	51,7	66,3	64,87	67,6	50,1	60,8	59,48
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,0	48,4	56,5	56,94
	Árvore de Decisão	67,8	40,3	55,4	54,50	63,2	46,2	51,8	53,72
	Random Forest	74,9	43,2	68,4	62,18	66,9	49,3	60,5	58,89
Média:						66,1	48,9	57,7	57,58
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	66,4	48,9	52,2	55,85
	Naive Bayes	70,4	48,2	58,8	59,12	63,7	48	49,2	53,60
	Regressão Log.	76,6	51,7	66,3	64,87	66,6	48,5	52,5	55,85
	Multilayer Perc.	76,3	51,9	65,6	64,60	67,6	48,4	56,7	57,56
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,7	46	52,9	54,85
	Random Forest	74,9	43,2	68,4	62,18	66,0	47,9	52,4	55,42
Média:						66,0	48,0	52,7	55,52

Fonte: Elaborada pelo autor.

Tabela 55 – *Self-training* adicionando 10% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	72,7	51,3	62,6	62,19
	Naive Bayes	70,4	48,2	58,8	59,12	68,0	48,0	56,6	57,56
	Regressão Logística	76,6	51,7	66,3	64,87	73,4	51,8	63,0	62,74
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,6	51,0	62,3	61,97
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,3	46,2	57,1	57,52
	Random Forest	74,9	43,2	68,4	62,18	73,3	50,1	67,0	63,47
Média:						71,6	49,7	61,4	60,91
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	53,1	43,5	55,7	50,76
	Naive Bayes	70,4	48,2	58,8	59,12	49,7	41,5	53,2	48,11
	Regressão Log.	76,6	51,7	66,3	64,87	53,0	43,3	55,5	50,59
	Multilayer Perc.	76,3	51,9	65,6	64,60	54,4	41,7	54,4	50,14
	Árvore de Decisão	67,8	40,3	55,4	54,50	50,9	41,1	51,5	47,83
	Random Forest	74,9	43,2	68,4	62,18	47,7	40,9	52,2	46,90
Média:						51,5	42,0	53,8	49,06
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	74,3	45,6	64,2	61,38
	Naive Bayes	70,4	48,2	58,8	59,12	70,1	37,0	58,3	55,13
	Regressão Log.	76,6	51,7	66,3	64,87	74,4	45,7	64,9	61,63
	Multilayer Perc.	76,3	51,9	65,6	64,60	75,1	47,3	64,9	62,41
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,1	37,1	60,0	55,38
	Random Forest	74,9	43,2	68,4	62,18	73,8	32,5	66,7	57,64
Média:						72,8	40,9	63,2	58,93
MLP	SVM Linear	73,3	47,2	62,2	60,89	73,2	52,8	64,3	63,41
	Naive Bayes	70,4	48,2	58,8	59,12	68,2	49,3	58,3	58,62
	Regressão Log.	76,6	51,7	66,3	64,87	73,3	52,6	63,6	63,18
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,5	53,0	63,9	63,47
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,2	46,3	58,2	57,92
	Random Forest	74,9	43,2	68,4	62,18	72,6	51	65,3	62,97
Média:						71,7	50,8	62,3	61,60
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	67,5	50,2	62,5	60,05
	Naive Bayes	70,4	48,2	58,8	59,12	65,8	49,0	55,8	56,87
	Regressão Log.	76,6	51,7	66,3	64,87	67,7	50,1	63,3	60,36
	Multilayer Perc.	76,3	51,9	65,6	64,60	64,3	46,5	55,3	55,39
	Árvore de Decisão	67,8	40,3	55,4	54,50	63,0	46,8	52,9	54,25
	Random Forest	74,9	43,2	68,4	62,18	68,1	49,2	60,1	59,13
Média:						66,1	48,6	58,3	57,68
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	66,0	46,4	46,0	52,79
	Naive Bayes	70,4	48,2	58,8	59,12	63,2	44,2	44,7	50,68
	Regressão Log.	76,6	51,7	66,3	64,87	65,9	45,9	46,6	52,77
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,0	43,3	49,0	52,78
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,7	44,3	47,2	52,39
	Random Forest	74,9	43,2	68,4	62,18	64,5	42,4	45,0	50,63
Média:						65,2	44,4	46,4	52,01

Fonte: Elaborada pelo autor.

Tabela 56 – *Self-training* adicionando 5% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	71,7	46,5	61,2	59,82
	Naive Bayes	70,4	48,2	58,8	59,12	70,0	43,3	58,3	57,21
	Regressão Log.	76,6	51,7	66,3	64,87	72,8	48,1	61,8	60,91
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,1	48,4	62,4	60,98
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,3	39,0	56,0	54,41
	Random Forest	74,9	43,2	68,4	62,18	75,1	43,5	66,8	61,80
Média:						71,7	44,8	61,1	59,19
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	46,7	42,4	54,6	47,89
	Naive Bayes	70,4	48,2	58,8	59,12	43,2	41,0	53,4	45,85
	Regressão Log.	76,6	51,7	66,3	64,87	47,2	42,4	54,6	48,07
	Multilayer Perc.	76,3	51,9	65,6	64,60	50,3	41,0	54,2	48,49
	Árvore de Decisão	67,8	40,3	55,4	54,50	45,8	39,8	51,1	45,57
	Random Forest	74,9	43,2	68,4	62,18	42,8	40,2	51,4	44,80
Média:						46,0	41,1	53,2	46,78
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	73,6	29,2	62,5	55,06
	Naive Bayes	70,4	48,2	58,8	59,12	69,3	23,6	56,7	49,85
	Regressão Log.	76,6	51,7	66,3	64,87	73,9	28,7	62,8	55,12
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,7	33,1	63,7	56,83
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,4	26,7	58,8	51,63
	Random Forest	74,9	43,2	68,4	62,18	72,4	18,5	64,1	51,66
Média:						72,1	26,6	61,4	53,36
MLP	SVM Linear	73,3	47,2	62,2	60,89	73,8	53,3	63,4	63,48
	Naive Bayes	70,4	48,2	58,8	59,12	68,0	49,8	58,0	58,62
	Regressão Log.	76,6	51,7	66,3	64,87	73,9	53,2	62,7	63,28
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,1	52,6	62,7	62,80
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,7	47,8	57,5	57,98
	Random Forest	74,9	43,2	68,4	62,18	72,5	51,8	64,3	62,85
Média:						71,7	51,4	61,4	61,50
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	68,5	49,7	61,4	59,87
	Naive Bayes	70,4	48,2	58,8	59,12	66,2	48,8	56,8	57,24
	Regressão Log.	76,6	51,7	66,3	64,87	68,8	50,4	61,0	60,03
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,1	47,0	59,3	57,45
	Árvore de Decisão	67,8	40,3	55,4	54,50	62,6	45,5	51,5	53,18
	Random Forest	74,9	43,2	68,4	62,18	68,5	49,4	62,1	60,00
Média:						66,8	48,5	58,7	57,96
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	66,5	41,1	47,0	51,52
	Naive Bayes	70,4	48,2	58,8	59,12	63,5	37,6	43,9	48,29
	Regressão Log.	76,6	51,7	66,3	64,87	66,5	40,2	48,8	51,81
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,5	39,3	52,4	52,74
	Árvore de Decisão	67,8	40,3	55,4	54,50	64,5	38,0	48,0	50,18
	Random Forest	74,9	43,2	68,4	62,18	65,0	37,1	46,1	49,39
Média:						65,4	38,9	47,7	50,66

Fonte: Elaborada pelo autor.

Tabela 57 – *Self-training* adicionando 1% dos documentos por iteração com TTsBR reduzido avaliando com três classes.

Método de Expansão	Classificador	TTsBR				TTsBR estendido			
		F-pos	F-neu	F-neg	F-Measure	F-pos	F-neu	F-neg	F-Measure
SVM	SVM Linear	73,3	47,2	62,2	60,89	73,0	43,0	63,2	59,72
	Naive Bayes	70,4	48,2	58,8	59,12	69,9	32,9	57,3	53,35
	Regressão Log.	76,6	51,7	66,3	64,87	73,0	41,9	63,1	59,32
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,5	42,9	63,3	59,58
	Árvore de Decisão	67,8	40,3	55,4	54,50	68,5	31,6	56,7	52,27
	Random Forest	74,9	43,2	68,4	62,18	74,1	25,9	65,9	55,33
Média:						71,8	36,4	61,6	56,60
Naive Bayes	SVM Linear	73,3	47,2	62,2	60,89	42,8	40,8	53,7	45,80
	Naive Bayes	70,4	48,2	58,8	59,12	40,4	39,7	52,7	44,26
	Regressão Log.	76,6	51,7	66,3	64,87	42,9	40,4	53,5	45,60
	Multilayer Perc.	76,3	51,9	65,6	64,60	45,4	39,2	52,6	45,73
	Árvore de Decisão	67,8	40,3	55,4	54,50	43,0	39,5	50,6	44,34
	Random Forest	74,9	43,2	68,4	62,18	40,7	40,3	51,9	44,31
Média:						42,5	40,0	52,5	45,01
Regressão Logística	SVM Linear	73,3	47,2	62,2	60,89	72,4	21,5	60,9	51,58
	Naive Bayes	70,4	48,2	58,8	59,12	68,6	18,7	54,4	47,22
	Regressão Log.	76,6	51,7	66,3	64,87	72,6	23,8	61,3	52,56
	Multilayer Perc.	76,3	51,9	65,6	64,60	72,3	26,9	62,3	53,81
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,3	21,8	55,1	48,72
	Random Forest	74,9	43,2	68,4	62,18	69,9	17,3	56,6	47,90
Média:						70,9	21,7	58,4	50,30
MLP	SVM Linear	73,3	47,2	62,2	60,89	73,4	51,4	64,5	63,09
	Naive Bayes	70,4	48,2	58,8	59,12	68,1	48,8	58,4	58,43
	Regressão Log.	76,6	51,7	66,3	64,87	73,6	52,0	64,5	63,35
	Multilayer Perc.	76,3	51,9	65,6	64,60	73,3	52,4	62,3	62,67
	Árvore de Decisão	67,8	40,3	55,4	54,50	69,8	48,1	58,5	58,79
	Random Forest	74,9	43,2	68,4	62,18	73,5	51,3	65,8	63,53
Média:						72,0	50,7	62,3	61,64
Árvore de Decisão	SVM Linear	73,3	47,2	62,2	60,89	69,1	48,5	62,3	59,95
	Naive Bayes	70,4	48,2	58,8	59,12	67,6	48,8	57,3	57,89
	Regressão Log.	76,6	51,7	66,3	64,87	69,6	48,7	62,7	60,36
	Multilayer Perc.	76,3	51,9	65,6	64,60	65,6	48,0	56,7	56,76
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,4	46,4	52,8	54,87
	Random Forest	74,9	43,2	68,4	62,18	68,4	49,4	62,3	60,02
Média:						67,6	48,3	59,0	58,31
Random Forest	SVM Linear	73,3	47,2	62,2	60,89	67,2	26,9	51,0	48,37
	Naive Bayes	70,4	48,2	58,8	59,12	66,2	33,4	47,5	49,02
	Regressão Log.	76,6	51,7	66,3	64,87	67,5	29,2	51,5	49,39
	Multilayer Perc.	76,3	51,9	65,6	64,60	66,2	32,4	52,3	50,32
	Árvore de Decisão	67,8	40,3	55,4	54,50	65,6	29,4	51,9	48,95
	Random Forest	74,9	43,2	68,4	62,18	66,5	25,5	51,7	47,90
Média:						66,5	29,5	51,0	48,99

Fonte: Elaborada pelo autor.

Tabela 58 – *Self-training* adicionando 40% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,7	75,5	79,56
	Naive Bayes	81,4	70,4	75,92	80,2	71,5	75,85
	Regressão Log.	86,4	77,5	81,95	84,2	76,8	80,49
	Multilayer Perc.	87,0	79,2	83,11	83,4	75,3	79,33
	Árvore de Decisão	80,9	70,0	75,43	81	73,3	77,12
	Random Forest	74,9	78,4	82,85	85,1	78,9	81,98
Média:					82,93	75,2	79,1
Naive Bayes	SVM Linear	84,4	74,7	79,57	80,4	72,3	76,35
	Naive Bayes	81,4	70,4	75,92	78,5	69,2	73,8
	Regressão Log.	86,4	77,5	81,95	80,9	73,1	76,98
	Multilayer Perc.	87,0	79,2	83,11	80,3	73,0	76,65
	Árvore de Decisão	80,9	70,0	75,43	79,2	69,4	74,29
	Random Forest	74,9	78,4	82,85	79,2	70,8	74,99
Média:					79,75	71,3	75,5
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,3	76,4	80,34
	Naive Bayes	81,4	70,4	75,92	80,3	71,2	75,78
	Regressão Log.	86,4	77,5	81,95	84,2	77,1	80,67
	Multilayer Perc.	87,0	79,2	83,11	83,9	76,1	80,03
	Árvore de Decisão	80,9	70,0	75,43	80,8	73,8	77,3
	Random Forest	74,9	78,4	82,85	84,8	78,7	81,74
Média:					83,05	75,6	79,3
MLP	SVM Linear	84,4	74,7	79,57	84,3	77,2	80,73
	Naive Bayes	81,4	70,4	75,92	80,9	72,5	76,68
	Regressão Log.	86,4	77,5	81,95	84,8	77,9	81,36
	Multilayer Perc.	87,0	79,2	83,11	84,9	77,9	81,4
	Árvore de Decisão	80,9	70,0	75,43	80,5	72,5	76,5
	Random Forest	74,9	78,4	82,85	84,8	78,9	81,85
Média:					83,37	76,2	79,8
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	79,4	73,0	76,2
	Naive Bayes	81,4	70,4	75,92	78,1	69,5	73,79
	Regressão Log.	86,4	77,5	81,95	81,1	75,1	78,1
	Multilayer Perc.	87,0	79,2	83,11	80	73,3	76,69
	Árvore de Decisão	80,9	70,0	75,43	78,1	68,7	73,39
	Random Forest	74,9	78,4	82,85	82,2	76,9	79,57
Média:					79,82	72,8	76,3
Random Forest	SVM Linear	84,4	74,7	79,57	83,1	74,3	78,67
	Naive Bayes	81,4	70,4	75,92	80,1	67,9	74
	Regressão Log.	86,4	77,5	81,95	83,8	75,3	79,57
	Multilayer Perc.	87,0	79,2	83,11	82,6	73,8	78,16
	Árvore de Decisão	80,9	70,0	75,43	81,5	71,4	76,44
	Random Forest	74,9	78,4	82,85	83,3	74,7	79,04
Média:					82,40	72,9	77,6

Fonte: Elaborada pelo autor.

Tabela 59 – Self-training adicionando 30% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,9	75,9	79,92
	Naive Bayes	81,4	70,4	75,92	80,1	71,1	75,61
	Regressão Log.	86,4	77,5	81,95	83,6	75,6	79,6
	Multilayer Perc.	87,0	79,2	83,11	83,3	75,4	79,34
	Árvore de Decisão	80,9	70,0	75,43	80,8	72,0	76,42
	Random Forest	74,9	78,4	82,85	85,2	79,0	82,12
		Média:			82,82	74,8	78,8
Naive Bayes	SVM Linear	84,4	74,7	79,57	79,9	72,1	76,00
	Naive Bayes	81,4	70,4	75,92	77,9	68,7	73,29
	Regressão Log.	86,4	77,5	81,95	80,1	71,9	76,01
	Multilayer Perc.	87,0	79,2	83,11	79,7	71,2	75,42
	Árvore de Decisão	80,9	70,0	75,43	77,6	68,4	72,98
	Random Forest	74,9	78,4	82,85	78,2	69,7	73,93
		Média:			78,90	70,3	74,6
Regressão Logística	SVM Linear	84,4	74,7	79,57	83,8	75,9	79,88
	Naive Bayes	81,4	70,4	75,92	80,2	70,8	75,47
	Regressão Log.	86,4	77,5	81,95	83,7	76,3	79,97
	Multilayer Perc.	87,0	79,2	83,11	84	76,6	80,28
	Árvore de Decisão	80,9	70,0	75,43	81,3	73,5	77,37
	Random Forest	74,9	78,4	82,85	83,9	77,6	80,75
		Média:			82,82	75,1	79,0
MLP	SVM Linear	84,4	74,7	79,57	83,6	76,2	79,89
	Naive Bayes	81,4	70,4	75,92	80,5	72,1	76,29
	Regressão Log.	86,4	77,5	81,95	83,8	76,8	80,31
	Multilayer Perc.	87,0	79,2	83,11	83,1	76,0	79,55
	Árvore de Decisão	80,9	70,0	75,43	80,6	72,4	76,53
	Random Forest	74,9	78,4	82,85	83,5	77,0	80,25
		Média:			82,52	75,1	78,8
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	79,3	73,0	76,14
	Naive Bayes	81,4	70,4	75,92	79,1	70,8	74,94
	Regressão Log.	86,4	77,5	81,95	79,8	73,5	76,68
	Multilayer Perc.	87,0	79,2	83,11	77	70,0	73,5
	Árvore de Decisão	80,9	70,0	75,43	75,3	66,3	70,79
	Random Forest	74,9	78,4	82,85	80,9	75,0	77,91
		Média:			78,57	71,4	75,0
Random Forest	SVM Linear	84,4	74,7	79,57	83,5	71,9	77,73
	Naive Bayes	81,4	70,4	75,92	80,1	63,9	72,01
	Regressão Log.	86,4	77,5	81,95	83,8	72,6	78,19
	Multilayer Perc.	87,0	79,2	83,11	83	73,5	78,21
	Árvore de Decisão	80,9	70,0	75,43	81,9	70,0	75,97
	Random Forest	74,9	78,4	82,85	83,1	70,0	76,55
		Média:			82,57	70,3	76,4

Fonte: Elaborada pelo autor.

Tabela 60 – *Self-training* adicionando 25% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84	76,1	80,01
	Naive Bayes	81,4	70,4	75,92	80,2	71,2	75,66
	Regressão Log.	86,4	77,5	81,95	84	76,5	80,26
	Multilayer Perc.	87,0	79,2	83,11	83,6	75,9	79,73
	Árvore de Decisão	80,9	70,0	75,43	80,8	72,4	76,62
	Random Forest	74,9	78,4	82,85	84,8	78,4	81,63
Média:					82,90	75,1	79,0
Naive Bayes	SVM Linear	84,4	74,7	79,57	78,9	71,7	75,3
	Naive Bayes	81,4	70,4	75,92	76,9	67,9	72,42
	Regressão Log.	86,4	77,5	81,95	79,2	71,8	75,48
	Multilayer Perc.	87,0	79,2	83,11	78,7	72,1	75,39
	Árvore de Decisão	80,9	70,0	75,43	75,8	67,7	71,77
	Random Forest	74,9	78,4	82,85	78,1	70,7	74,36
Média:					77,93	70,3	74,1
Regressão Logística	SVM Linear	84,4	74,7	79,57	83,9	76,0	79,95
	Naive Bayes	81,4	70,4	75,92	80,1	70,8	75,42
	Regressão Log.	86,4	77,5	81,95	84,2	76,9	80,56
	Multilayer Perc.	87,0	79,2	83,11	83,4	75,1	79,29
	Árvore de Decisão	80,9	70,0	75,43	82,5	75,2	78,88
	Random Forest	74,9	78,4	82,85	84,2	77,9	81,07
Média:					83,05	75,3	79,2
MLP	SVM Linear	84,4	74,7	79,57	84,1	77,2	80,63
	Naive Bayes	81,4	70,4	75,92	79,9	73,0	76,46
	Regressão Log.	86,4	77,5	81,95	83,9	77,3	80,56
	Multilayer Perc.	87,0	79,2	83,11	83,3	76,5	79,88
	Árvore de Decisão	80,9	70,0	75,43	80,9	73,6	77,25
	Random Forest	74,9	78,4	82,85	82,9	77,3	80,06
Média:					82,50	75,8	79,1
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	79,3	72,7	75,98
	Naive Bayes	81,4	70,4	75,92	78,4	69,4	73,85
	Regressão Log.	86,4	77,5	81,95	80,2	73,9	77,07
	Multilayer Perc.	87,0	79,2	83,11	77	70,1	73,57
	Árvore de Decisão	80,9	70,0	75,43	77,7	67,8	72,73
	Random Forest	74,9	78,4	82,85	81,4	75,2	78,26
Média:					79,00	71,5	75,2
Random Forest	SVM Linear	84,4	74,7	79,57	82,4	68,9	75,63
	Naive Bayes	81,4	70,4	75,92	80,4	63,0	71,69
	Regressão Log.	86,4	77,5	81,95	82,9	69,8	76,3
	Multilayer Perc.	87,0	79,2	83,11	82,4	71,1	76,73
	Árvore de Decisão	80,9	70,0	75,43	80,8	67,2	73,99
	Random Forest	74,9	78,4	82,85	82,1	67,0	74,55
Média:					81,83	67,8	74,8

Fonte: Elaborada pelo autor.

Tabela 61 – *Self-training* adicionando 20% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,5	75,4	79,44
	Naive Bayes	81,4	70,4	75,92	80,3	70,8	75,57
	Regressão Log.	86,4	77,5	81,95	83,6	75,6	79,59
	Multilayer Perc.	87,0	79,2	83,11	83,8	75,7	79,72
	Árvore de Decisão	80,9	70,0	75,43	82	73,0	77,49
	Random Forest	74,9	78,4	82,85	85,4	79,1	82,23
Média:					83,10	74,9	79,0
Naive Bayes	SVM Linear	84,4	74,7	79,57	78	71,5	74,75
	Naive Bayes	81,4	70,4	75,92	76,5	68,4	72,49
	Regressão Log.	86,4	77,5	81,95	78,5	72,3	75,38
	Multilayer Perc.	87,0	79,2	83,11	78,1	72,2	75,15
	Árvore de Decisão	80,9	70,0	75,43	76,2	69,6	72,89
	Random Forest	74,9	78,4	82,85	77,2	70,9	74,04
Média:					77,42	70,8	74,1
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,2	76,2	80,19
	Naive Bayes	81,4	70,4	75,92	80,7	71,3	75,98
	Regressão Log.	86,4	77,5	81,95	84,5	77,1	80,8
	Multilayer Perc.	87,0	79,2	83,11	83,6	75,7	79,68
	Árvore de Decisão	80,9	70,0	75,43	80,3	72,0	76,12
	Random Forest	74,9	78,4	82,85	84,3	77,6	80,93
Média:					82,93	75,0	78,9
MLP	SVM Linear	84,4	74,7	79,57	84,7	77,0	80,82
	Naive Bayes	81,4	70,4	75,92	82	72,9	77,46
	Regressão Log.	86,4	77,5	81,95	85,5	78,3	81,88
	Multilayer Perc.	87,0	79,2	83,11	84,5	77,2	80,84
	Árvore de Decisão	80,9	70,0	75,43	81,5	72,3	76,86
	Random Forest	74,9	78,4	82,85	85,6	77,9	81,71
Média:					83,97	75,9	79,9
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	79	72,5	75,74
	Naive Bayes	81,4	70,4	75,92	77,9	69,4	73,67
	Regressão Log.	86,4	77,5	81,95	80,2	74,0	77,08
	Multilayer Perc.	87,0	79,2	83,11	78,3	70,7	74,46
	Árvore de Decisão	80,9	70,0	75,43	75,8	66,8	71,31
	Random Forest	74,9	78,4	82,85	80,5	75,1	77,77
Média:					78,62	71,4	75,0
Random Forest	SVM Linear	84,4	74,7	79,57	82	64,6	73,32
	Naive Bayes	81,4	70,4	75,92	80,1	57,5	68,8
	Regressão Log.	86,4	77,5	81,95	82,1	64,8	73,47
	Multilayer Perc.	87,0	79,2	83,11	82,5	68,1	75,29
	Árvore de Decisão	80,9	70,0	75,43	81,1	64,8	72,97
	Random Forest	74,9	78,4	82,85	81,9	63,1	72,48
Média:					81,62	63,8	72,7

Fonte: Elaborada pelo autor.

Tabela 62 – *Self-training* adicionando 10% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,0	75,7	79,85
	Naive Bayes	81,4	70,4	75,92	80,4	71,2	75,82
	Regressão Log.	86,4	77,5	81,95	84,1	76,2	80,12
	Multilayer Perc.	87,0	79,2	83,11	83,6	75,7	79,68
	Árvore de Decisão	80,9	70,0	75,43	80,6	72,0	76,32
	Random Forest	74,9	78,4	82,85	84,8	78,5	81,64
		Média:			82,92	74,9	78,9
Naive Bayes	SVM Linear	84,4	74,7	79,57	64,3	66,4	65,36
	Naive Bayes	81,4	70,4	75,92	66,8	65,8	66,32
	Regressão Log.	86,4	77,5	81,95	64	66,3	65,15
	Multilayer Perc.	87,0	79,2	83,11	68,2	67,4	67,79
	Árvore de Decisão	80,9	70,0	75,43	67	66,4	66,66
	Random Forest	74,9	78,4	82,85	64,2	65,9	65,03
		Média:			65,75	66,4	66,1
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,4	75,6	80,01
	Naive Bayes	81,4	70,4	75,92	80,6	68,9	74,75
	Regressão Log.	86,4	77,5	81,95	84,6	75,9	80,29
	Multilayer Perc.	87,0	79,2	83,11	84,1	75,0	79,58
	Árvore de Decisão	80,9	70,0	75,43	81,4	71,1	76,24
	Random Forest	74,9	78,4	82,85	84,8	75,3	80,05
		Média:			83,32	73,6	78,5
MLP	SVM Linear	84,4	74,7	79,57	84,4	77,1	80,76
	Naive Bayes	81,4	70,4	75,92	80,5	72,6	76,56
	Regressão Log.	86,4	77,5	81,95	84,5	77,6	81,03
	Multilayer Perc.	87,0	79,2	83,11	84,3	77,3	80,81
	Árvore de Decisão	80,9	70,0	75,43	80,6	72,3	76,43
	Random Forest	74,9	78,4	82,85	84	77,8	80,89
		Média:			83,05	75,8	79,4
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	80,4	74,6	77,47
	Naive Bayes	81,4	70,4	75,92	77,9	69,4	73,67
	Regressão Log.	86,4	77,5	81,95	80,3	74,5	77,4
	Multilayer Perc.	87,0	79,2	83,11	77	70,5	73,76
	Árvore de Decisão	80,9	70,0	75,43	74,9	68,4	71,65
	Random Forest	74,9	78,4	82,85	80,4	75,1	77,73
		Média:			78,48	72,1	75,3
Random Forest	SVM Linear	84,4	74,7	79,57	80,3	53,8	67,01
	Naive Bayes	81,4	70,4	75,92	79,1	48,9	64
	Regressão Log.	86,4	77,5	81,95	80,4	54,8	67,58
	Multilayer Perc.	87,0	79,2	83,11	80,1	55,9	67,99
	Árvore de Decisão	80,9	70,0	75,43	79,3	54,4	66,82
	Random Forest	74,9	78,4	82,85	79,2	50,7	64,96
		Média:			79,73	53,1	66,4

Fonte: Elaborada pelo autor.

Tabela 63 – *Self-training* adicionando 5% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	83,9	75,3	79,6
	Naive Bayes	81,4	70,4	75,92	80,9	70,7	75,81
	Regressão Log.	86,4	77,5	81,95	83,9	75,5	79,7
	Multilayer Perc.	87,0	79,2	83,11	83,3	74,3	78,77
	Árvore de Decisão	80,9	70,0	75,43	80,3	70,9	75,59
	Random Forest	74,9	78,4	82,85	85,6	78,2	81,89
Média:					82,98	74,2	78,6
Naive Bayes	SVM Linear	84,4	74,7	79,57	56,2	64,2	60,17
	Naive Bayes	81,4	70,4	75,92	57,9	63,4	60,65
	Regressão Log.	86,4	77,5	81,95	56,5	64,3	60,38
	Multilayer Perc.	87,0	79,2	83,11	56,4	63,8	60,09
	Árvore de Decisão	80,9	70,0	75,43	60,2	63,6	61,9
	Random Forest	74,9	78,4	82,85	57,4	63,9	60,63
Média:					57,43	63,9	60,6
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,2	73,0	78,57
	Naive Bayes	81,4	70,4	75,92	80,4	65,1	72,73
	Regressão Log.	86,4	77,5	81,95	84,7	73,8	79,21
	Multilayer Perc.	87,0	79,2	83,11	84,7	73,7	79,2
	Árvore de Decisão	80,9	70,0	75,43	81,9	68,2	75,03
	Random Forest	74,9	78,4	82,85	84,2	70,3	77,24
Média:					83,35	70,7	77,0
MLP	SVM Linear	84,4	74,7	79,57	84,2	76,9	80,57
	Naive Bayes	81,4	70,4	75,92	80,8	73,0	76,90
	Regressão Log.	86,4	77,5	81,95	84,3	77,5	80,92
	Multilayer Perc.	87,0	79,2	83,11	84,3	77,3	80,75
	Árvore de Decisão	80,9	70,0	75,43	80,4	72,0	76,23
	Random Forest	74,9	78,4	82,85	83,9	77,8	80,85
Média:					82,98	75,8	79,4
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	80,8	74,2	77,49
	Naive Bayes	81,4	70,4	75,92	79	70,0	74,51
	Regressão Log.	86,4	77,5	81,95	81,4	74,9	78,17
	Multilayer Perc.	87,0	79,2	83,11	78,4	70,9	74,67
	Árvore de Decisão	80,9	70,0	75,43	79	70,1	74,58
	Random Forest	74,9	78,4	82,85	81,8	75,9	78,84
Média:					80,07	72,7	76,4
Random Forest	SVM Linear	84,4	74,7	79,57	79,8	52,6	66,16
	Naive Bayes	81,4	70,4	75,92	79	48,5	63,74
	Regressão Log.	86,4	77,5	81,95	80,3	55,0	67,66
	Multilayer Perc.	87,0	79,2	83,11	80,6	58,3	69,46
	Árvore de Decisão	80,9	70,0	75,43	79,2	54,5	66,86
	Random Forest	74,9	78,4	82,85	79,8	53,2	66,49
Média:					79,78	53,7	66,7

Fonte: Elaborada pelo autor.

Tabela 64 – *Self-training* adicionando 1% dos documentos por iteração com TTsBR reduzido na classificação binária.

Método de Expansão	Classificador	TTsBR			TTsBR estendido		
		F-pos	F-neg	F-Measure	F-pos	F-neg	F-Measure
SVM	SVM Linear	84,4	74,7	79,57	84,4	74,6	79,49
	Naive Bayes	81,4	70,4	75,92	80,9	68,2	74,53
	Regressão Log.	86,4	77,5	81,95	84,2	74,1	79,15
	Multilayer Perc.	87,0	79,2	83,11	83,7	73,2	78,44
	Árvore de Decisão	80,9	70,0	75,43	80,8	68,3	74,56
	Random Forest	74,9	78,4	82,85	85,6	74,9	80,27
Média:					83,27	72,2	77,7
Naive Bayes	SVM Linear	84,4	74,7	79,57	52,5	62,7	57,61
	Naive Bayes	81,4	70,4	75,92	50	61,3	55,64
	Regressão Log.	86,4	77,5	81,95	52,3	63,0	57,62
	Multilayer Perc.	87,0	79,2	83,11	58,2	64,3	61,27
	Árvore de Decisão	80,9	70,0	75,43	59,2	63,9	61,56
	Random Forest	74,9	78,4	82,85	52,8	62,4	57,62
Média:					54,17	62,9	58,6
Regressão Logística	SVM Linear	84,4	74,7	79,57	84,5	70,2	77,35
	Naive Bayes	81,4	70,4	75,92	80,6	61,5	71,05
	Regressão Log.	86,4	77,5	81,95	84,4	70,1	77,26
	Multilayer Perc.	87,0	79,2	83,11	84,2	70,0	77,09
	Árvore de Decisão	80,9	70,0	75,43	81,7	62,8	72,27
	Random Forest	74,9	78,4	82,85	83,2	61,5	72,37
Média:					83,10	66,0	74,6
MLP	SVM Linear	84,4	74,7	79,57	84,2	77,0	80,59
	Naive Bayes	81,4	70,4	75,92	80,2	71,9	76,04
	Regressão Log.	86,4	77,5	81,95	84,7	78,0	81,32
	Multilayer Perc.	87,0	79,2	83,11	84,6	77,0	80,77
	Árvore de Decisão	80,9	70,0	75,43	81,4	73,8	77,59
	Random Forest	74,9	78,4	82,85	83,7	77,5	80,56
Média:					83,13	75,9	79,5
Árvore de Decisão	SVM Linear	84,4	74,7	79,57	80,1	73,5	76,77
	Naive Bayes	81,4	70,4	75,92	79,1	70,4	74,74
	Regressão Log.	86,4	77,5	81,95	80,4	73,9	77,18
	Multilayer Perc.	87,0	79,2	83,11	77,1	69,1	73,1
	Árvore de Decisão	80,9	70,0	75,43	76,8	67,1	71,92
	Random Forest	74,9	78,4	82,85	81,1	75,2	78,11
Média:					79,10	71,5	75,3
Random Forest	SVM Linear	84,4	74,7	79,57	79,5	56,7	68,07
	Naive Bayes	81,4	70,4	75,92	79,7	53,1	66,39
	Regressão Log.	86,4	77,5	81,95	79,6	57,2	68,39
	Multilayer Perc.	87,0	79,2	83,11	80,2	59,8	69,96
	Árvore de Decisão	80,9	70,0	75,43	79,1	58,5	68,79
	Random Forest	74,9	78,4	82,85	79,8	58,3	69,05
Média:					79,65	57,3	68,4

Fonte: Elaborada pelo autor.

AVALIAÇÃO DO *FRAMEWORK* DE EXPANSÃO NOS CÓRPUS DA LITERATURA

Tabela 65 – Avaliação do framework de expansão usando o corpúscos Buscapé-2.

<i>Threshold</i> de Adição	Classificador	F-pos	F-neg	F-Measure
40%	SVM Linear	78,1	65,2	71,68
	Naive Bayes	72,3	71,3	71,80
	Regressão Logística	84,4	83,1	83,77
	MLP	84,4	84,1	84,23
	Árvore de Decisão	74,7	74,6	74,65
	<i>Random Forest</i>	77,1	78,5	77,78
30%	SVM Linear	72,9	68,0	70,43
	Naive Bayes	72,8	73,5	73,15
	Regressão Logística	84,6	83,7	84,13
	MLP	84,2	84,0	84,07
	Árvore de Decisão	76,3	76,3	76,28
	<i>Random Forest</i>	77,3	77,8	77,53
25%	SVM Linear	27,7	62,0	44,87
	Naive Bayes	72,9	71,3	72,11
	Regressão Logística	84,1	84,0	84,04
	MLP	83,7	84,3	83,97
	Árvore de Decisão	75,7	75,4	75,56
	<i>Random Forest</i>	77,9	79,2	78,56
20%	SVM Linear	73,7	76,9	75,29
	Naive Bayes	73,1	72,1	72,57
	Regressão Logística	84,4	83,3	83,83
	MLP	85,1	84,1	84,59
	Árvore de Decisão	76,7	76,5	76,59
	<i>Random Forest</i>	76,2	77,8	76,98
10%	SVM Linear	80,3	73,4	76,81
	Naive Bayes	73,1	70,5	71,80
	Regressão Logística	84,3	83,7	84,01
	MLP	84,3	84,1	84,22
	Árvore de Decisão	76,3	75,8	76,01
	<i>Random Forest</i>	74,3	77,1	75,66
5%	SVM Linear	63,2	74,3	68,74
	Naive Bayes	73,0	71,6	72,32
	Regressão Logística	84,4	83,6	83,95
	MLP	85,1	83,9	84,50
	Árvore de Decisão	74,6	74,5	74,56
	<i>Random Forest</i>	74,0	77,2	75,61
1%	SVM Linear	73,0	47,7	60,33
	Naive Bayes	74,0	70,6	72,26
	Regressão Logística	85,2	84,0	84,62
	MLP	85,0	84,4	84,74
	Árvore de Decisão	75,3	73,9	74,61
	<i>Random Forest</i>	74,0	77,7	75,83

Fonte: Elaborada pelo autor.

Tabela 66 – Avaliação do *framework* de expansão usando o *corpus* Mercado Livre.

<i>Threshold</i> de Adição	Classificador	F-pos	F-neg	F-Measure
40 %	SVM Linear	69,8	32,1	50,93
	Naive Bayes	73,4	71,9	72,65
	Regressão Logística	92,8	92,6	92,66
	MLP	92,8	92,6	92,68
	Árvore de Decisão	84,7	84,2	84,47
	<i>Random Forest</i>	87,7	88,1	87,86
30 %	SVM Linear	86,8	83,8	85,34
	Naive Bayes	72,2	70,9	71,52
	Regressão Logística	93,1	93,0	93,02
	MLP	93,1	92,9	93,01
	Árvore de Decisão	84,2	84,4	84,28
	<i>Random Forest</i>	86,8	87,2	86,98
25 %	SVM Linear	84,2	78,3	81,24
	Naive Bayes	72,9	71,2	72,03
	Regressão Logística	93,0	92,8	92,92
	MLP	92,8	92,9	92,83
	Árvore de Decisão	83,7	83,4	83,54
	<i>Random Forest</i>	87,6	87,8	87,68
20 %	SVM Linear	78,5	65,2	71,82
	Naive Bayes	73,1	72,5	72,79
	Regressão Logística	92,9	92,7	92,81
	MLP	92,7	92,5	92,60
	Árvore de Decisão	83,4	83,2	83,30
	<i>Random Forest</i>	87,1	87,4	87,21
10 %	SVM Linear	84,8	79,5	82,14
	Naive Bayes	70,8	68,0	69,35
	Regressão Logística	93,2	93,1	93,17
	MLP	93,1	93,0	93,04
	Árvore de Decisão	84,8	84,6	84,70
	<i>Random Forest</i>	85,8	86,1	85,97
5 %	SVM Linear	80,8	69,7	75,23
	Naive Bayes	71,2	68,7	69,93
	Regressão Logística	93,0	92,9	92,93
	MLP	93,0	92,8	92,86
	Árvore de Decisão	83,3	82,7	83,01
	<i>Random Forest</i>	82,7	84,0	83,35
1 %	SVM Linear	80,0	67,5	73,77
	Naive Bayes	70,0	65,1	67,59
	Regressão Logística	92,8	92,6	92,67
	MLP	93,0	92,7	92,88
	Árvore de Decisão	83,0	82,9	82,92
	<i>Random Forest</i>	81,5	84,1	82,79

Fonte: Elaborada pelo autor.

Tabela 67 – Avaliação do framework de expansão usando o corpú Eleições-Dilma.

<i>Threshold</i> de Adição	Classificador	F-pos	F-neg	F-Measure
40%	SVM Linear	83,4	41,9	62,65
	Naive Bayes	86,4	64,4	75,39
	Regressão Logística	90,2	73,9	82,06
	MLP	90,1	77,1	83,57
	Árvore de Decisão	88,2	71,0	79,61
	<i>Random Forest</i>	90,3	69,6	79,94
30%	SVM Linear	85,1	45,3	65,21
	Naive Bayes	84,2	55,2	69,68
	Regressão Logística	90,1	73,0	81,58
	MLP	90,6	76,7	83,65
	Árvore de Decisão	88,2	70,8	79,49
	<i>Random Forest</i>	89,9	68,7	79,31
25%	SVM Linear	81,5	35,8	58,63
	Naive Bayes	85,4	63,0	74,19
	Regressão Logística	90,5	74,6	82,55
	MLP	90,6	75,5	83,05
	Árvore de Decisão	88,4	71,4	79,91
	<i>Random Forest</i>	90,0	68,6	79,34
20%	SVM Linear	84,0	35,4	59,69
	Naive Bayes	83,4	53,2	68,33
	Regressão Logística	90,2	73,4	81,82
	MLP	90,5	75,7	83,12
	Árvore de Decisão	88,1	70,4	79,26
	<i>Random Forest</i>	89,7	66,4	78,06
10%	SVM Linear	84,1	23,7	53,89
	Naive Bayes	83,8	52,9	68,34
	Regressão Logística	90,1	73,1	81,62
	MLP	90,2	75,7	82,90
	Árvore de Decisão	88,3	71,1	79,70
	<i>Random Forest</i>	89,3	63,7	76,46
5%	SVM Linear	81,1	0,9	41,02
	Naive Bayes	85,1	56,9	70,99
	Regressão Logística	90,0	72,2	81,09
	MLP	90,6	76,5	83,54
	Árvore de Decisão	88,0	71,6	79,82
	<i>Random Forest</i>	89,1	60,1	74,57
1%	SVM Linear	82,0	02,2	42,11
	Naive Bayes	82,8	46,4	64,62
	Regressão Logística	89,6	70,4	79,98
	MLP	90,7	76,7	83,69
	Árvore de Decisão	88,1	71,6	79,82
	<i>Random Forest</i>	88,8	58,1	73,41

Fonte: Elaborada pelo autor.

Tabela 68 – Avaliação do *framework* de expansão usando o corpus Eleições-Serra.

<i>Threshold</i> de Adição	Classificador	F-pos	F-neg	F-Measure
40 %	SVM Linear	78,4	96,5	87,47
	Naive Bayes	38,2	88,4	63,29
	Regressão Logística	75,9	96,5	86,20
	MLP	45,0	94,3	69,65
	Árvore de Decisão	74,5	95,9	85,22
	<i>Random Forest</i>	76,6	96,6	86,58
30 %	SVM Linear	76,9	96,5	86,68
	Naive Bayes	35,7	91,7	63,72
	Regressão Logística	72,0	96,1	84,08
	MLP	65,8	95,7	80,72
	Árvore de Decisão	72,5	95,3	83,88
	<i>Random Forest</i>	79,5	96,9	88,23
25 %	SVM Linear	77,0	96,3	86,65
	Naive Bayes	34,1	73,2	53,64
	Regressão Logística	75,5	96,5	85,97
	MLP	23,5	93,3	58,40
	Árvore de Decisão	73,0	95,4	84,22
	<i>Random Forest</i>	76,5	96,6	86,56
20 %	SVM Linear	79,7	96,7	88,16
	Naive Bayes	30,5	49,2	39,80
	Regressão Logística	74,6	96,2	85,41
	MLP	29,4	93,6	61,51
	Árvore de Decisão	76,8	96,1	86,44
	<i>Random Forest</i>	72,7	96,4	84,56
10 %	SVM Linear	78,1	96,5	87,34
	Naive Bayes	32,2	89,0	60,60
	Regressão Logística	72,3	96,1	84,21
	MLP	56,3	95,0	75,68
	Árvore de Decisão	75,9	96,1	86,00
	<i>Random Forest</i>	75,9	96,7	86,26
5 %	SVM Linear	78,5	96,6	87,52
	Naive Bayes	35,3	91,8	63,56
	Regressão Logística	77,3	96,6	86,95
	MLP	07,9	92,7	50,31
	Árvore de Decisão	74,7	95,6	85,14
	<i>Random Forest</i>	69,2	96,2	82,68
1 %	SVM Linear	77,4	96,7	87,04
	Naive Bayes	38,7	91,9	65,25
	Regressão Logística	77,0	96,5	86,75
	MLP	29,9	93,6	61,76
	Árvore de Decisão	74,8	95,9	85,36
	<i>Random Forest</i>	77,0	96,6	86,81

Fonte: Elaborada pelo autor.