UNIVERSIDADE FEDERAL DE OURO PRETO

Milton Stiilpen Júnior

Um Arcabouço de Processamento de Textos Informais em Português Brasileiro para Aplicações de Mineração de Dados

Ouro Preto

2016

UNIVERSIDADE FEDERAL DE OURO PRETO

Milton Stiilpen Júnior

Um Arcabouço de Processamento de Textos Informais em Português Brasileiro para Aplicações de Mineração de Dados

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do título de Mestre.

Orientador:
Luiz Henrique de Campos Merschmann

Ouro Preto

2016

S855u Stiilpen Jr., Milton.
Um arcabouço de processamento de textos informais em português brasileiro para aplicações de mineração de dados [manuscrito] / Milton Stiilpen Jr.. - 2016.
41f.: il.: color; grafs; tabs.

Orientador: Prof. Dr. Luiz Henrique de Campos Merschmann.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Mineração de dados (Computação). 2. Redes sociais on-line. 3. Processamento da linguagem natural (Computação). I. Merschmann, Luiz Henrique de Campos . II. Universidade Federal de Ouro Preto. III. Titulo.

CDU: 004

Catalogação: www.sisbin.ufop.br

Ata da Defesa Pública de Dissertação de Mestrado

Aos 29 dias do mês de setembro de 2016, às 10 horas na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: **Prof. Dr. Luiz Henrique de Campos Merschmann (presidente e orientador), Prof. Dr. Anderson Almeida Ferreira e Prof. Dr. Fabrício Benevenuto de Souza**, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem o mestrando **Milton Stiilpen Júnior**, com o título **"Um Arcabouço de Processamento de Textos Informais em Português Brasileiro para Aplicações de Mineração de Dados"**. Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:

(X) Aprovada sem recomendações

( ) Reprovada

( ) Aprovada com recomendações: _____

Banca Examinadora:

Prof. Dr. Luiz Henrique de Campos Merschmann

Prof. Dr. Anderson Almeida Ferreira

Prof. Dr. Fabrício Benevenuto de Souza

Prof. Dr. Anderson Almeida Ferreira
Coordenador do Programa de Pós-Graduação em Ciência da Computação
DECOM/ICEB/UFOP

Ouro Preto, 29 de setembro de 2016.

Resumo

Redes Sociais *online* (RSO) surgiram no início do século XXI e dão indícios de que terão vida longa. Cerca de 64% dos usuários de mídias sociais dizem acessar ao menos uma rede social todos os dias. Desse modo, é imensa a quantidade de dados gerados por esses canais de comunicação. O Processamento de Linguagem Natural em textos de redes sociais é um tema de pesquisa recente que vem atraindo um número cada vez maior de pesquisadores. Portanto, neste trabalho, é proposta um arcabouço capaz de lidar com a diversidade do português brasileiro, com o informalismo, com a natureza de tempo real e com a falta de contextualização de textos publicados em redes sociais. O arcabouço proposto foi avaliado em duas tarefas (Categorização de Texto e Mineração de Opinião) e os resultados experimentais mostraram que os mecanismos de pré-processamento existentes no arcabouço foram importantes para obtenção de bons resultados.

Palavras-chave: Mineração de Dados, Redes Sociais *Online*, Processamento de Linguagem Natural.

Abstract

Social Networks emerged at the beginning of 21st century and give us evidence that they are going to have a long life. Almost two-thirds of overall social media users affirm an everyday usage of a social media website and, therefore, the data volume across this platforms is huge. Natural language processing of social media texts is an attractive topic among researchers of this area. While there are many studies about natural language processing of social media texts for some languages (e.g., English), the researches for Brazilian Portuguese language are still limited. Then, in this work, a framework is proposed to deal with peculiarities of the Brazilian Portuguese language in informal, short and noisy texts, where the lack of context poses obstacles in text mining. The proposed framework has been evaluated in two tasks (Text Categorization and Opinion Mining) and experiments showed that the preprocessing mechanisms included in this framework were important to achieve better results.

Keywords: Data Mining; Social Networks; Natural Language Processing.

Glossário

RSO	:	Redes Sociais Online;
PLN	:	Processamento de Linguagem Natural;
EI	:	Extração de Informação;
KDD	:	*Knowledge Discovery in Databases*;
MD	:	Mineração de Dados;
MT	:	Mineração de Textos;
POS	:	*Part of Speech*;
CT	:	Categorização Textual;
REN	:	Reconhecimento de Entidades Nomeadas;
MO	:	Mineração de Opinião;
SVM	:	*Support Vector Machine*;
API	:	Interface de Programação de Aplicações;

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Redes sociais são alvo de pesquisas, principalmente por parte dos sociólogos. Nos últimos anos, com a eminência da Internet, as redes sociais online (RSO) se estabeleceram como um canal de comunicação de alto valor comercial. Por meio delas, as pessoas trocam informações e opinam publicamente sobre diversos assuntos, formando um grande repositório de dados não estruturados.

Cerca de 64% dos usuários de mídias sociais acessam ao menos uma rede todos os dias. Desse modo, é imensa a quantidade de dados gerados por esses canais de comunicação (Nielsen, 2014). O Twitter, uma plataforma social de publicações curtas (*tweets*), reportou receber meio bilhão de publicações diárias[1] em 2014. No período das eleições presidenciais brasileiras no ano de 2015, quase 40 milhões de *tweets* foram publicados nessa plataforma[2]. O Facebook, uma outra rede social que também recebe diariamente um grande volume de publicações, já alegou ter problemas para processá-las[3]. Nota-se, portanto, um grande desafio para se processar essa quantidade de dados não estruturados e extrair informações relevantes para pequenas e grandes organizações.

A fim de enfrentar esse desafio, diversos trabalhos na literatura apresentam propostas que utilizam técnicas de Processamento de Linguagem Natural (PLN) para derivar entendimento dos textos publicados e, assim, reconhecer assuntos (Categorização Textual), correlacionar entidades (Reconhecimento de Entidades Nomeadas) e descobrir a opinião de usuários (Mineração de Opinião) (Pang and Lee, 2008; Ritter et al., 2011; Gonçalves et al., 2013).

Contudo, no cenário das redes sociais online, onde os textos são curtos e boa parte deles

[1] http://about.twitter.com/company (acesso em 06/2015)
[2] http://glo.bo/1Ct4ulu (acesso em 06/2015)
[3] http://techcrunch.com/2014/12/28/mining-the-hive-mind (acesso em 06/2015)

não apresenta contexto do assunto ou uma revisão textual, os algoritmos tradicionalmente utilizados para resolver problemas como CT e REN a partir de textos formais, têm seu desempenho consideravelmente degradado (Ritter et al., 2011). Dessa forma, existe o desafio de se lidar com o informalismo textual, com a diversidade da língua, com a natureza de tempo real, com a falta de contextualização e ainda assim manter a eficácia desses algoritmos (Oliveira et al., 2013).

O processamento de linguagem natural em textos de redes sociais é um tema de pesquisa recente em que pesquisadores vêm propondo adaptações das abordagens que lidam com textos formais (Ritter et al., 2011; Oliveira et al., 2013; Bontcheva et al., 2013). Nesse tema, uma atenção particular tem sido dada à uma tarefa de PLN denominada Normalização Textual (Han and Baldwin, 2011; Liu et al., 2012; Duran et al., 2014; Avanço et al., 2014; Xie et al., 2016). No contexto das redes sociais, a normalização tem como objetivos remover ruídos linguísticos e formalizar o texto, a fim de que técnicas de PLN adotadas em textos formais possam ser utilizadas e alcancem resultados similares aos reportados na literatura para aquele cenário.

Portanto, neste trabalho propõe-se um arcabouço capaz de lidar com a complexidade dos textos publicados nas redes sociais. A hipótese é que, com uma ideia simples e adaptável, o arcabouço proposto nos permitirá trabalhar com as peculiaridades do vernáculo. Utilizando-se uma coleção de textos do Twitter (*tweets*) e avaliações de produtos obtidas na Google Play, o arcabouço foi avaliado para as tarefas de Classificação Textual e Mineração de Opinião, respectivamente.

As principais contribuições deste trabalho são:

- Uma revisão dos principais trabalhos com estratégias propostas para lidar com textos informais em português brasileiro (Capítulo 3).
- Proposta de um arcabouço para pré-processar textos curtos e informais, publicados no português brasileiro. Essa proposta combina diferentes técnicas de pré-processamento textual, dispersas em diferentes trabalhos reportados na literatura (Capítulo 4).
- Implementação e avaliação do arcabouço proposto em duas aplicações de mineração de textos: Categorização Textual (CT) e Mineração de Opinião (MO) (Capítulo 5).

Capítulo 2

Fundamentação Teórica

Este capítulo traz uma visão geral dos conceitos de Processamento de Linguagem Natural e Mineração de Dados utilizados neste trabalho.

O Processamento de Linguagem Natural é a área da Inteligência Artificial (IA) responsável por fazer o computador reconhecer, entender e sintetizar a linguagem utilizada na comunicação humana, tanto falada quanto escrita.

A título de exemplo, algumas aplicações de PLN são: reconhecimento de fala, síntese de voz, tradução automática, correção automática de texto, extração de informação em textos e sumarização automática.

Ao trabalhar com PLN, uma prática comum é dividir os problemas em tarefas menores: segmentação de sentenças, tokenização, *part-of-speech* (POS) *tagging*, *chunking*, *parsing*, entre outras (Jurafsky and Martin, 2000). Para resolução dessas tarefas utilizam-se sistemas baseados em regras, modelos probabilísticos ou aqueles que combinam as duas abordagens (híbridos).

Um das áreas de grande sinergia com os problemas de PLN é a Mineração de Dados (MD). MD é uma área interdisciplinar em Ciência da Computação responsável pela descoberta de padrões em grandes bases de dados. Ela é uma das etapas de um processo maior denominado *Knowledge Discovery in Databases – (KDD)*, que visa a extração de conhecimento de grandes massas de dados. A Figura 2.1 apresenta cada uma das etapas desse processo: seleção dos dados, pré-processamento, transformação, mineração e interpretação/avaliação dos resultados (Fayyad et al., 1996).

A Mineração de Dados aplicada no domínio textual é denominada Mineração de Textos. Porém, há uma particularidade em Mineração de Textos: os dados não são numéricos

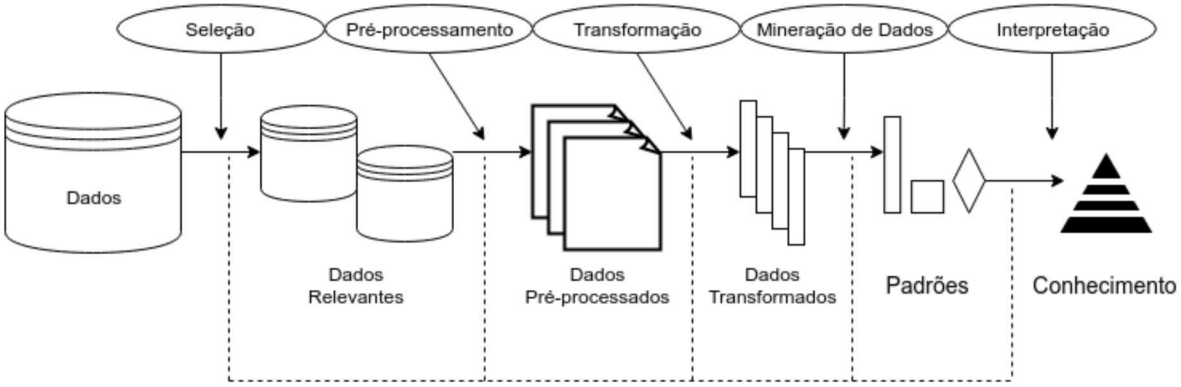

Figura 2.1: Fases do processo KDD

e estruturados, como os métodos de MD normalmente requerem. Uma coleção de textos precisa ser processada e transformada numa representação numérica (Sholom M. Weiss, 2015), a fim de servir como entrada para técnicas de Mineração de Dados. Eis que, em contra-partida, PLN pode auxiliar no processo de estruturar corretamente as características textuais e melhorar os resultados de procedimentos de MD.

Nas seções seguintes são explicitados os conceitos de PLN e MD utilizados neste trabalho.

2.1 Corpus

A maior parte das técnicas de PLN utiliza um corpus linguístico, que é uma coleção de textos escritos ou registros orais, estruturados em uma determinada língua. Esse recurso é utilizado para modelagem estatística do idioma, aprendizado de regras de coocorrência entre outras análises. Em alguns casos, o corpus contém informações adicionais obtidas a partir de um processo de varredura e etiquetagem de trechos do texto (conhecido na língua inglesa como *tagging*). Exemplos de corpus na língua portuguesa são o Corpus Brasileiro[1] e o CETENFolha [2].

2.2 Tarefas de Processamento de Linguagem Natural

Esta seção apresenta as principais tarefas de PLN, a saber, segmentação de sentenças, tokenização, etiquetagem morfológica, *chunking* e análise sintática.

[1] `http://corpusbrasileiro.pucsp.br/`
[2] `http://www.linguateca.pt/`

2.2.1 Segmentação de Sentenças

A segmentação de sentenças de um texto é a tarefa de definir limite entre as sentenças, comumente baseada em regras (Manning and Schütze, 1999). Nessa tarefa, o texto é percorrido a fim de encontrar uma delimitação e, assim que encontrada, caso não seja parte de uma regra de exceção, a sentença é segmentada do texto.

Os passos comuns da tarefa de segmentação de um texto em sentenças são: percorrer os caracteres; encontrar uma pontuação; checagem de padrões como links e e-mails; checagem do próximo caractere (espaço em branco ou letra capitalizada) e checagem de outras regras possíveis (por exemplo, se é uma abreviação ou se a sentença está inclusa em uma citação). O exemplo da Figura 2.2 mostra a segmentação de um texto realizada pela ferramenta NLTK [3].

```
>>> sentences = nltk.sent_tokenize('Pedro fala de Paulo, pois ele está no A.A.
 Portanto, sabe-se mais de Pedro do que de Paulo.', language='portuguese')
>>> print sentences[0]
Pedro fala de Paulo, pois ele está no A.A.
>>> print sentences[1]
Portanto, sabe-se mais de Pedro do que de Paulo.
```

Figura 2.2: Exemplo do procedimento de segmentação de sentenças

2.2.2 Tokenização

O procedimento de tokenização corresponde à separação de cada sentença em um conjunto de termos. Esses termos são comumente representados por palavras, numerais e pontuações e são geralmente limitados por espaços em branco. O resultado dessa tarefa pode ser utilizado como entrada de outras tarefas menos triviais, tal como etiquetagem morfológica (ver Seção 2.2.3).

2.2.3 Etiquetagem Morfológica

Na Linguística Computacional, a etiquetagem morfológica (no inglês *Part-of-speech tagging – POS tagging*) é o procedimento de anotar os termos do texto em alguma categoria morfológica.

A morfologia é o estudo da estrutura, da formação e da classificação das palavras. De forma simplificada, na língua portuguesa as categorias morfológicas são: artigo, substantivo, verbo, adjetivo, advérbio, pronome, preposição, conjunção e interjeição.

[3] http://http://www.nltk.org/

```
>>> sentence = 'Saudades da Dilma #CerimoniaDeAbertura <3'
>>> terms = nltk.word_tokenize(sentence, language='portuguese')
>>> print terms[0]
Saudades
>>> print terms[1]
da
>>> print terms[2]
Dilma
>>> print terms[3]
#
>>> print terms[4]
CerimoniaDeAbertura
>>> print terms[5]
<
>>> print terms[6]
3
```

Figura 2.3: Exemplo do procedimento de tokenização

A tarefa de *POS tagging* é crucial no entendimento da formação de uma sentença. O resultado dessa tarefa pode ser utilizado como entrada de outras mais complexas, tais como as apresentadas nas Seções 2.2.4 e 2.2.5, ou para a extração de características em algoritmos de aprendizado de máquina (Bird et al., 2009).

O exemplo apresentado na Figura 2.4 apresenta o resultado da tarefa de *POS tagging* quando aplicada aos termos gerados no exemplo da Figura 2.3. Nesse exemplo, baseado nas características observadas pelo etiquetador, o termo "Saudades" foi categorizado como substantivo (N), o termo "da" foi categorizado como uma contração de preposição com artigo (PREP+ART), o termo "Dilma" foi categorizado como nome próprio (NPROP), o termo "*hash tag*" foi categorizado como um nome próprio (NPROP), o termo "CerimoniaDeAbertura" foi categorizado como nome próprio (NPROP), o termo "menor que" foi categorizado como pontuação (PU) e, por fim, o termo "três" foi categorizado como substantivo (N).

```
>>> tags = tagger.tag('Saudades da Dilma #CerimoniaDeAbertura <3')[0]
>>> for tag in tags:
...   print tag[0] + ' = ' + tag[1]
...
Saudades = N
da = PREP+ART
Dilma = NPROP
# = NPROP
CerimoniaDeAbertura = NPROP
< = PU
3 = N
```

Figura 2.4: Exemplo do procedimento de *Part-of-Speech tagging*

2.2.4 Chunking

A tarefa de *Chunking*, também conhecida como análise sintática superficial, consiste em etiquetar e agrupar termos sintaticamente correlacionados (sintagmas). As unidades mais simples de sintagmas podem ser definidas a partir de regras (baseado nas categorias morfológicas das palavras) como, por exemplo, sintagmas nominais e sintagmas verbais.

Na Figura 2.5 é possível notar o resultado dessa tarefa. Substantivos (N) e nomes próprios (NPROP) foram anotados como sintagmas nominais. Além disso, com uma regra do tipo "NPROP|N * NPROP|N" foi possível agrupar "Cerimônia de Abertura" no último sintagma nominal.

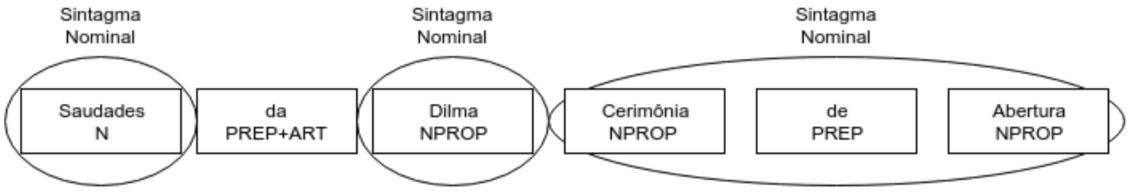

Figura 2.5: Exemplo da tarefa de *Chunking*

2.2.5 Análise Sintática

A tarefa denominada Análise Sintática (*Parsing*, em inglês) realiza a construção da estrutura sintática de uma sentença, baseada em uma gramática formal. A representação de sua estrutura é, usualmente, uma árvore sintática.

O objetivo dessa tarefa é reconhecer as funções dos elementos de uma sentença e, assim, estruturá-los de acordo com suas relações. Os elementos considerados nesta tarefa podem ser, dependendo da necessidade, desde os termos essenciais (sujeito e predicado) até termos integrantes (complemento verbal, complemento nominal e agente da passiva) e termos acessórios (adjuntos e aposto).

Na Figura 2.6, é mostrado o resultado do LX-Parser[4], serviço online gratuito de análise sintática, anotando e construindo a ávore dado o último exemplo "Saudades da Dilma Cerimônia de Abertura".

Como é possível notar, o sintagma nominal (NP) "Cerimônia de Abertura" foi classificado como um sintagma verbal (VP). Deste erro, deriva-se o termo "Cerimônia" como verbo (V), o sintagma preposicional (PP) "de abertura". Já a outra parte da estrutura

[4]`http://lxcenter.di.fc.ul.pt/services/online_parser/caracteristicas.html`

(corretamente etiquetado) representa o sintagma nominal principal, que foi composto por um substantivo (N) e um sintagma preposicional (PP).

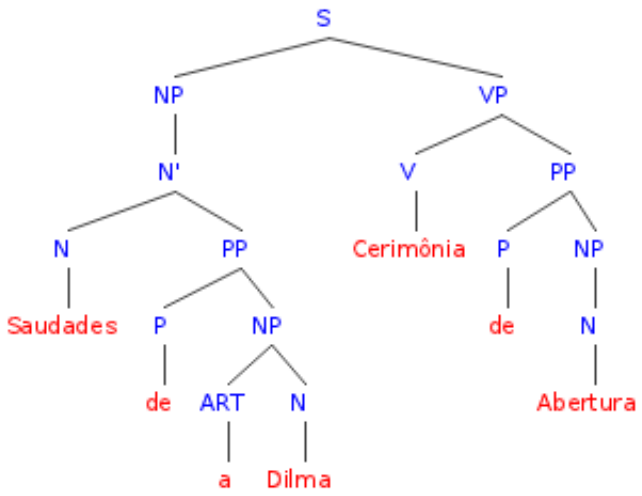

Figura 2.6: Exemplo de construção de árvore sintática após etiquetagem automática pelo LX-PARSER

2.3 Classificação

Também conhecida como Aprendizado Supervisionado, a tarefa de classificação é uma das principais em Mineração de Dados. Ela consiste em predizer uma ou mais classes de uma instância a partir de suas características.

O processo de classificação pode ser dividido em duas etapas: treinamento do modelo preditivo e a avaliação do mesmo. O modelo é obtido a partir da análise de um conjunto de instâncias que possuem suas características e classes conhecidas. A esse conjunto de instâncias dá-se o nome de base de dados de treinamento. Após o treinamento do modelo de classificação, um outro conjunto de instâncias que também possui características e classes conhecidas, mas que não foi considerado no treinamento do modelo, é utilizado na avaliação da capacidade preditiva do modelo treinado. Esse segundo conjunto de instâncias é denominado base de dados de teste.

Para treinar um modelo de classificação, diversas técnicas já foram propostas na literatura. A próxima seção apresenta a técnica *Support Vector Machine (SVM)*, utilizada na construção dos modelos preditivos utilizados neste trabalho. Em seguida, na Seção 2.3.2, descreve-se a técnica denominada Vetorização de Textos, utilizada para transformar um texto em uma instância que possa ser utilizada no treinamento/teste de classificadores.

Por fim, na Seção 2.3.3 são apresentados os conceitos relacionados à avaliação do desempenho preditivo de classificadores.

2.3.1 Support Vector Machine (SVM)

Considere um problema onde suas instâncias são divididas em duas classes opostas e em uma dimensão de duas características (X_1 e X_2). A técnica denominada SVM tem como objetivo encontrar uma fronteira no espaço (hiperplano) que separe as instâncias de acordo com as suas respectivas classes. Ela executa uma varredura entre as diversas instâncias de uma base de dados com o objetivo de calcular o hiperplano ótimo a partir do uso de vetores de suporte (isto é, aquele que possui a maior margem de distância entre duas instâncias de classes distintas).

No exemplo da Figura 2.7 nota-se que é possível posicionar um hiperplano H de diferentes formas, tais como H_1, H_2 ou H_3. Contudo, ainda que os hiperplanos H_2 e H_3 acertem na separação entre as instâncias das duas classes, o H_3 é considerado melhor do que o H_2 por possuir uma margem de separação maior para as instâncias de classes distintas e, portanto, apresentar maior poder de generalização. Sendo assim, para esse conjunto de instâncias, o H_3 é considerado o hiperplano ótimo.

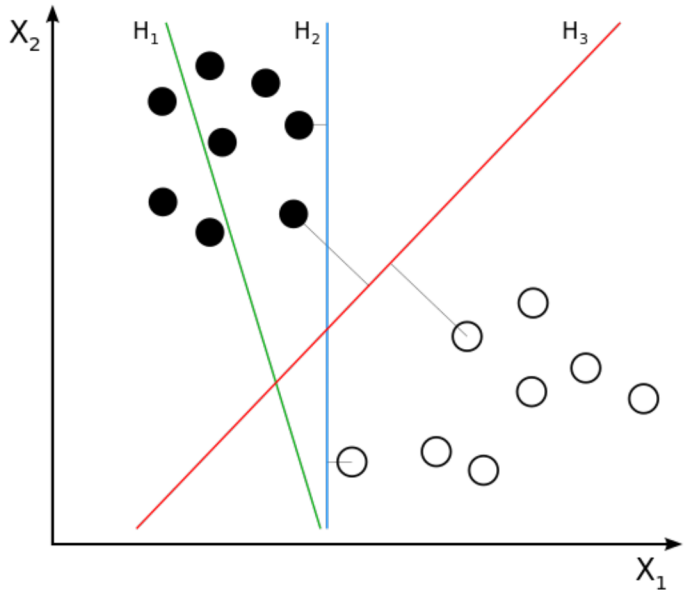

Figura 2.7: Exemplo de funcionamento do algoritmo SVM

Porém, é comum que instâncias de problemas reais possuam mais do que duas dimensões de características e, diferentemente do exemplo apresentado na Figura 2.7, não sejam linearmente separáveis, dificultando a computação desse hiperplano ótimo. Para lidar com o problema de instâncias não linearmente separáveis, o SVM utiliza uma função

denominada *kernel* para transformar os dados originais da base de dados colocando-os em um novo espaço dimensional onde as instâncias passam a ser linearmente separáveis.

Com o SVM, a etapa de treinamento do modelo de classificação consiste em encontrar os parâmetros que determinam o hiperplano ótimo. Uma vez treinado o modelo, a classificação de uma instância desconhecida é feita determinando-se a posição da mesma em relação ao hiperplano de separação.

2.3.2 Vetorização de Textos

Para que uma coleção de documentos textuais seja convertida em uma base de dados para o treinamento de classificadores, os textos precisam ser transformados em vetores numéricos que os representem. O processo denominado Vetorização de Textos, responsável por essa transformação, permite representar um texto a partir de um vetor em que cada uma de suas posições está associada a um termo conhecido de uma coleção. Nesse caso, o valor numérico de cada posição do vetor corresponde ao grau de relevância que o termo associado àquela posição possui no texto em questão.

O cálculo do grau de relevância pode ser realizado de diferentes maneiras. A função mais simples de relevância é a binária, que gera um valor igual a um numa determinada posição do vetor se o termo associado àquela posição estiver presente no texto ou, caso contrário, gera o valor zero. Outras formas mais elaboradas de calcular o grau de relevância envolvem o cálculo de frequência dos termos ou a computação de correlação entre os termos.

A Figura 2.8 exemplifica os passos desse processo de vetorização. Nesse exemplo considera-se uma coleção formada por três avaliações textuais sobre um filme e suas respectivas classificações sobre o sentimento (“positivo” ou “negativo”) de quem o assistiu. O primeiro passo nesse processo é listar e contar todos os termos presentes na coleção, formando um vocabulário. Com o vocabulário pronto, é possível representar as três avaliações utilizando-se como base um vetor de termos que possui em cada uma de suas posições a frequência de cada termo na coleção. O último passo do processo consiste em preencher o vetor que representa cada avaliação utilizando o grau de relevância contido no vetor de termos quando um determinado termo está presente na avaliação ou, caso contrário, com o valor zero.

Por questão de simplificação do exemplo apresentado na Figura 2.8, a frequência absoluta do termo na coleção foi utilizada como grau de relevância. No entanto, outras

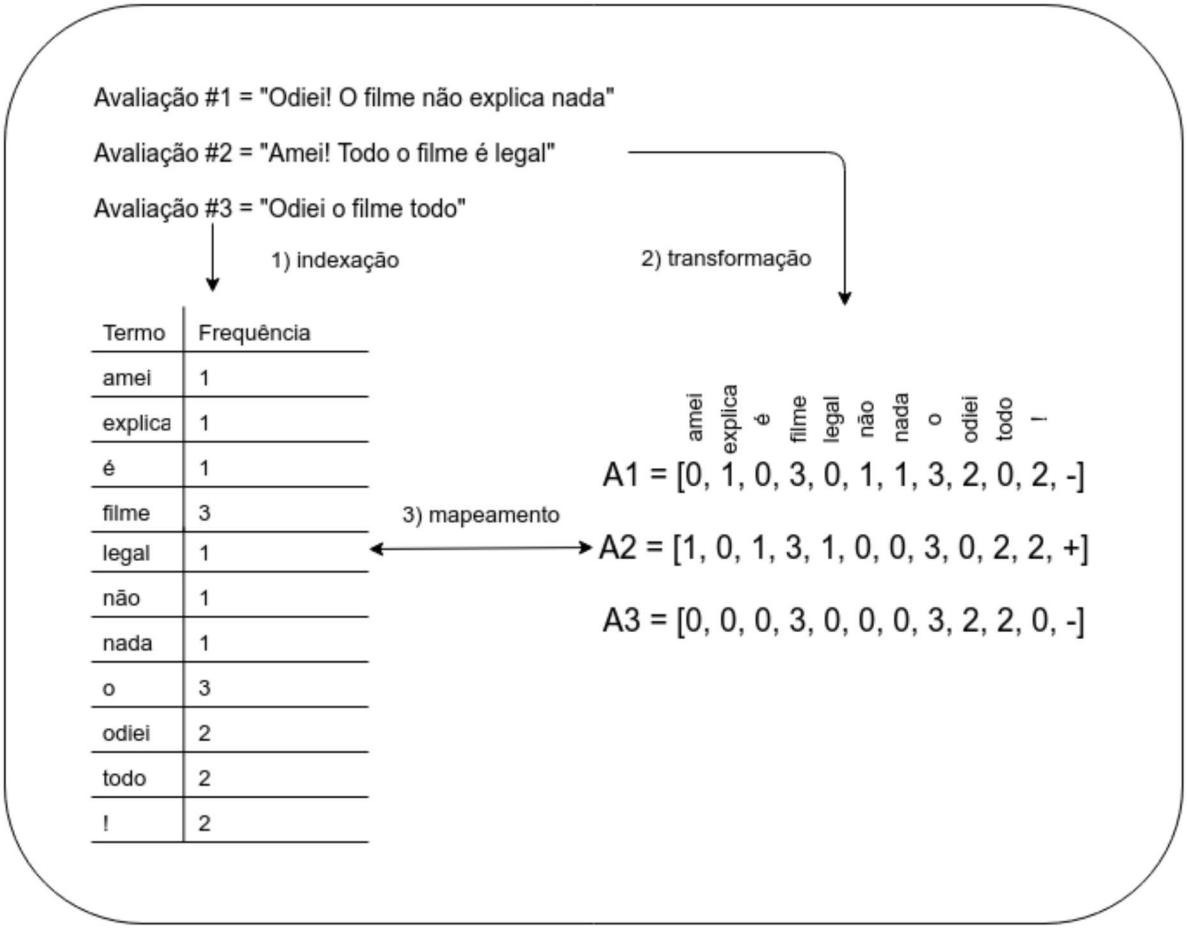

Figura 2.8: Exemplo da transformação de textos para vetores numéricos

formas de calcular a importância de um termo no texto, tais como frequência relativa e frequência inversa de documentos, podem ser utilizadas.

Tendo em vista os resultados de Soucy and Mineau (2005), neste trabalho, o grau de relevância de um termo foi calculado utilizando-se a medida estatística $tf.idf$ (do inglês, *term frequency–inverse document frequency*). Essa métrica leva em consideração a frequência do termo no texto e também a proporção de documentos (textos) da coleção que o contêm. A Equação 2.1 apresenta a medida $tf.idf$ para um termo t num texto d considerando-se a coleção de documentos D.

$$tf.idf(d, t, D) = tf(d, t) * idf(t, D), \tag{2.1}$$

onde $tf(d, t)$ (Equação 2.2) é a frequência do termo t no texto d e $idf(t, D)$ (Equação 2.3) corresponde ao inverso da razão entre a quantidade de documentos da coleção que contêm o termo t e o número total de documentos na coleção D.

$$tf(d,t) = 1 + log(freq(t,d)), \tag{2.2}$$

onde a função $freq$ representa a contagem de um termo t no documento d.

$$idf(t,D) = log(\frac{N}{n_t}), \tag{2.3}$$

onde N representa o número total de documentos na coleção D e n_t o número de documentos em que o termo t está contido.

2.3.3 Avaliação de Classificadores

A segunda etapa de um processo de classificação corresponde à avaliação do modelo preditivo construído na etapa de treinamento. Usualmente essa avaliação é feita estimando-se o valor de alguma medida de desempenho preditivo por meio da técnica denominada k-validação cruzada.

A técnica k-validação cruzada consiste em dividir a base de dados original em k partições de tamanho (quantidade de instâncias) idealmente iguais. Em seguida, para cada uma das k iterações executadas, um modelo classificação é treinado utilizando-se $k-1$ partições (base de treinamento) e avaliado a partir das instâncias da partição restante (base de teste). Vale ressaltar que em cada iteração uma partição distinta é utilizada na avaliação do modelo, de modo que, no final do processo, todas as partições terão sido utilizadas uma vez na avaliação do modelo de classificação.

Diferentes medidas podem ser utilizadas na avaliação do modelo de classificação na etapa de teste. Neste trabalho, foram adotadas medidas comumente empregadas em avaliação de classificadores, a saber, precisão, revocação e F1 (Aggarwal, 2015).

A precisão (P) e a revocação (R) de uma classe i são definidas como:

$$P_i = \frac{VP_i}{VP_i + FP_i}, \quad R_i = \frac{VP_i}{VP_i + FN_i}, \tag{2.4}$$

onde VP_i é o número de instâncias corretamente classificadas como classe i, FN_i corresponde ao número de instâncias que pertencem à classe i mas não foram classificadas como tal e FP_i é o número de instâncias que não pertencem à classe i mas foram incorretamente classificadas como sendo da classe i.

A medida F1, uma métrica tradicionalmente utilizada para representar o desempenho

de classificadores, é conveniente por combinar a precisão e a revocação em uma única medida de qualidade. Ela corresponde à média harmônica entre a precisão e a revocação.

$$F1_i = 2 \times \frac{P_i \times R_i}{P_i + R_i} \tag{2.5}$$

Capítulo 3

Trabalhos Relacionados

Conforme observado em (Irfan et al., 2015), o pré-processamento tem papel importante para a tarefa de mineração de textos em RSO, dada a informalidade dos textos nesse meio. Nesse trabalho, os autores citam a importância das análises morfológicas (por exemplo, remoção de *stopwords* e *stemming*), sintáticas (por exemplo, *POS tagger* e *Parser*) e semânticas (por exemplo, utilização de dicionários semânticos para contextualizar sentenças). Além disso, eles verificaram que os principais algoritmos de classificação que vêm sendo utilizados em trabalhos de mineração de textos em RSO são Árvores de Decisão, KNN, SVM e Redes Neurais.

Em (Ritter et al., 2011), o desempenho de ferramentas de PLN (originalmente propostas para lidar com textos formais) é avaliado em um conjunto de publicações de redes sociais. Com uma base de 800 *tweets* rotulados manualmente, os autores propuseram o T-NER, que corresponde a uma adaptação de todos os módulos envolvidos na tarefa de REN em textos formais. Desse modo, eles apresentaram ganhos de desempenho desde o primeiro módulo, que é responsável pela etiquetagem morfológica das palavras (*POS tagging*), até a classificação das entidades nomeadas. Ainda, os autores experimentaram a adição de um módulo no processo que é responsável por classificar automaticamente a qualidade da capitalização do texto. A abordagem proposta nesse trabalho mostrou-se promissora, tendo em vista o comparativo dos resultados contra ferramentas como Stanford NLP (http://nlp.stanford.edu) e OpenNLP (http://opennlp.apache.org). No entanto, o T-NER foi implementado para a língua inglesa e não resolve a questão dos erros de ortografia e de capitalização.

Já Oliveira et al. (2013) apresentam um outro tipo de adaptação para REN, inspirado no alto volume de mensagens no Twitter. Essencialmente, essa abordagem pode ser vista como uma hierarquia de filtros, que possuem a finalidade de detectar e classificar as

entidades. Os filtros são rápidos, compostos por dicionários e atributos comuns à tarefa de REN, independentes de regras de sintaxe e que podem ser combinados ou paralelizados para enfatizar precisão ou revocação. O objetivo do trabalho descrito nesse artigo é alcançar um alto desempenho e não perder acurácia do método (tendo como ponto de comparação o *Conditional Random Fields*, um algoritmo estado da arte em tarefas de anotação textual). Os autores superam a questão do desempenho, mas a acurácia do método decaiu mais do que o esperado.

Os autores de (Bontcheva et al., 2013), em um trabalho mais recente e baseado em (Ritter et al., 2011), trazem o TwitIE como uma adaptação completa dos módulos do GATE - Generic Architeture for Text Engineering (uma ferramenta de PLN para textos formais) para lidar com textos de RSO. Seu diferencial encontra-se em dois módulos: tokenização de termos e normalização textual. O primeiro módulo é responsável por um tratamento especial das *hashtags*, URLs e *emoticons*. Já o segundo é responsável pela autocorreção ortográfica. Vale ressaltar que o TwitIE também não suporta textos na língua portuguesa.

A abordagem de autocorreção ortográfica utilizada em (Bontcheva et al., 2013) foi explorada para a língua portuguesa em (Avanço et al., 2014). Nesse trabalho, os autores desenvolveram um corretor ortográfico automático para o vernáculo que é capaz de alcançar melhor desempenho que o conhecido Aspell (http://aspell.net). A heurística proposta nesse trabalho combina fundamentalmente três características: regras fonéticas (propostas pelos autores), distância Levenshtein e frequência dos termos candidatos à substituição do termo identificado como erro. Como resultado, obtiveram uma acurácia de 65% para um experimento realizado a partir de um conjunto de 1323 palavras amostradas em uma coleção de avaliações textuais de produtos.

Ademais, a correção ortográfica é dependente de um léxico formal, ou seja, palavras corretas existentes no idioma nacional. No cenário de redes sociais, os textos contêm várias abreviações informais, neologismos, memes e outras palavras fora do vocabulário (no inglês *Out of Vocabulary - OOV*) que não representam, de fato, um erro e, assim, dificultam a tarefa de autocorreção. Em (Hartmann et al., 2014) , foram listados e avaliados tipos de OOV em um corpus de avaliações de produtos obtido no comparador de preços Buscapé (http://www.buscape.com.br). Os principais tipos de OOV são: acrônimos, nomes próprios (a maioria sendo nome de companhias e produtos), abreviações, gírias da Internet, estrangeirismo e unidades de medida (ex.: terabytes). A identificação e o tratamento das OOV nesse trabalho são implementados a partir de transformações

lexicais.

O trabalho proposto por Duran et al. (2014) mostra que a normalização textual não pode-se resumir simplesmente à correção ortográfica ou ao tratamento de termos não encontrados no vocabulário. Nesse trabalho, foram levantados os principais problemas de normalização em textos de avaliações de produtos que apresentam as mesmas características dos *tweets* (ruidosos e curtos). Inicialmente, a partir de uma amostra de textos de avaliações de produtos, uma correção manual dos textos foi realizada para se identificar quais tipos de normalização traziam melhoria na precisão de um anotador automático (*tagger*). Com isso, os autores concluíram que as normalizações mais importantes foram: correção de capitalização e correção de pontuação. Em seguida, abordagens para se automatizar o processo de normalização foram implementadas, todavia não obtiveram sucesso.

Por fim, os trabalhos relacionados à Mineração de Opinião (Petz et al., 2014; Dey and Haque, 2009) demonstram a relevância do pré-processamento textual e da adaptação de algoritmos de PLN (propostos para textos formais) para a extração de opiniões de usuários na Web 2.0. Em (Petz et al., 2014), é realizada uma caracterização detalhada sobre os canais sociais (Facebook, Twitter, blogs e fóruns). Os resultados atestam o desafio envolvendo a contextualização e normalização de textos do Twitter, porquanto 82% de suas publicações são subjetivas e aproximadamente 40% das mesmas possuem pelo menos 75% do seu conteúdo textual repleto de erros. Esses números são compatíveis com aqueles apresentados no trabalho (Dey and Haque, 2009), que propõe um método de limpeza (ex: remoção de caracteres especiais, tratamento de abreviações) e fusão de sentenças, a fim de normalizar textos de opiniões e, consequentemente, melhorar seus resultados.

Este trabalho se diferencia das propostas apresentadas nos principais trabalhos relacionados (Ritter et al., 2011; Bontcheva et al., 2013; Dey and Haque, 2009) por tratar o problema de textos informais a partir de uma normalização mais completa (correção de capitalização, correção gramatical e processamento dos apelidos de perfis sociais) e lidar com a questão de contextualização da publicação. Além disso, enfrenta-se o desafio de lidar com textos escritos na língua portuguesa.

Capítulo 4

Arcabouço Proposto

Nesta seção é apresentado o arcabouço proposto para pré-processar e extrair informação de textos escritos em canais de comunicação social.

O arcabouço é composto por um conjunto de procedimentos encadeados, onde cada um deles executa uma tarefa específica. A sequência de procedimentos, apresentada na Figura 4.1, é essencialmente modelada pelo padrão *Pipes and Filters* (Monroe et al., 1996), que garante a escalabilidade e extensibilidade da mesma.

A função de cada procedimento do arcabouço proposto é descrita a seguir:

1. Detecção de Linguagem: responsável por filtrar textos para que somente aqueles escritos na língua portuguesa sigam para os próximos procedimentos;

2. Normalização Textual: responsável por formalizar o texto a partir de diversos tipos de transformações (p. ex., limpeza de caracteres especiais e tratamento de gírias e de termos fora do vocabulário), correções (p. ex., ortográfica, gramatical, pontuação e capitalização) e expansão de contrações.

3. Tokenização: realiza a segmentação de sentenças, de termos e de combinações específicas de caracteres de textos de redes sociais;

4. Extração de Características: responsável por extrair características (morfológicas, lexical e sintáticas) do texto.

5. Contextualização: tem como finalidade a expansão da informação contida no texto a partir de dados estruturados ou semiestruturados provenientes de fontes externas, tais como Wikipédia, OpenWordNet (Paiva et al., 2012), LIWC (Pennebaker et al., 2001) e outras;

Figura 4.1: Diagrama da sequência de procedimentos do arcabouço

6. Transformação e Redução dos Dados (TRD): responsável por realizar a vetorização dos textos, a discretização e seleção de atributos. Sendo a última etapa do arcabouço, a sua saída pode ser utilizada diretamente por algoritmos de mineração de dados visando a resolução de problemas como Categorização Textual (CT), Reconhecimento de Entidades Nomeadas (REN), Mineração de Opinião (MO), Modelagem de Tópicos (MT), Sumarização Automática (SA), entre outros (Petz et al., 2014).

As seções a seguir apresentam mais detalhes de cada uma das etapas (procedimentos) utilizadas na composição do arcabouço proposto.

4.1 Detecção de Língua

O primeiro procedimento do arcabouço é a identificação da língua regente no texto. O objetivo desse módulo é fazer com que qualquer conteúdo que difira do vernáculo não seja processado pela abordagem proposta.

Apesar de as ferramentas de mídias sociais geralmente realizarem a classificação da linguagem utilizada pelo usuário, essa classificação nem sempre é precisa, uma vez que ela pode ser feita considerando-se outras características (p. ex., informações do usuário) que vão além do texto escrito.

Neste trabalho, uma biblioteca de detecção de linguagem desenvolvida em Java foi utilizada com modelos preditivos treinados a partir de um classificador para o cenário de redes sociais (Shuyo, 2010).

A importância dessa etapa no arcabouço proposto pode ser observada a partir do exemplo a seguir. Considere a seguinte publicação realizada no Twitter:

> "LÁMINA DE VIDRIO TEMPLADO ULTRA RESISTENTE! PARA SAMSUNG, MOTO Y IPHONE"

Enquanto o Twitter classifica essa publicação como tendo sido escrita na língua portuguesa, o módulo de detecção utilizado no arcabouço o classifica, com 99% de confiança, como sendo da língua espanhola.

4.2 Normalização Textual

A normalização textual encarrega-se de converter um texto para sua forma canônica. Existem diversas técnicas para tanto, que vão desde as mais simples, envolvendo a redução do texto para caixa baixa e remoção de caracteres especiais, até outras menos triviais, que realizam auto capitalização, correção gramatical e outros procedimentos.

A normalização textual é comumente aplicada a textos resultantes do processo de conversão de uma fala em texto, com a finalidade de reestruturá-lo como um texto escrito, ou na revisão automática, transformando um texto informal para sua versão formal. O problema tratado aqui é de revisão automática, pelo fato de que a linguagem utilizada por usuários na *web*, normalmente, destoa da norma culta.

A normalização é uma etapa de fundamental importância para o sucesso do arcabouço proposto neste trabalho. Nessa etapa, as seguintes tarefas são executadas: limpeza textual, reconhecimento de gírias e de palavras fora do vocabulário, auto capitalização, correção de pontuação, correção ortográfica, correção gramatical e, opcionalmente, expansão de contrações. Para tal, utilizam-se de léxicos pré-definidos, sistemas baseado em regras (Dey and Haque, 2009) e modelos probabilísticos, tal como em (Nebhi et al., 2015),

computados a partir do corpus de artigos da Wikipédia.

As subseções a seguir apresentam cada uma das tarefas realizadas na etapa de normalização.

4.2.1 Limpeza Textual

A primeira fase da normalização executa as tarefas de limpeza de ruídos comumente encontrados nos textos de canais sociais (como por exemplo, textos automaticamente inseridos por ferramentas de mídias sociais) e o tratamento de gírias, memes e outras palavras fora do vocabulário. Neste trabalho, o processo de tratamento de gírias, memes e palavras fora do vocabulário é baseado na proposta apresentada por Hartmann et al. (2014).

A partir da detecção de palavras fora do vocabulário, transformações lexicais são realizadas a fim de converter gírias e abreviações frequentemente utilizadas na Internet em palavras conhecidas do vernáculo. A seguir, dois exemplos ilustram essa tarefa de limpeza.

1. **Texto com ruído:** "RT @YouTube: Aprenda a fazer hamburger com cheddar Sadia".

   Após a limpeza: "Aprenda a fazer hamburger com cheddar Sadia".

2. **Texto com gíria e palavra fora do vocabulário:** "miga é miga sim todo mundo é miga hahahah".

   Após a limpeza: "Amiga é amiga sim todo mundo é amiga, risos".

4.2.2 Correção de Pontuação

A correção de pontuação é a tarefa que auxilia diretamente a segmentação de sentenças e termos (processos posteriores). Segundo Duran et al. (2014) ela é uma das principais correções para melhorar o resultado de um etiquetador automático responsável por predizer a classe morfológica, por exemplo.

Neste trabalho implementou-se um sistema baseado em regras (expressões regulares,

como por exemplo "[a-z]p{Punct}+?[a-zˆ.]+"), semelhante ao proposto em (Dey and Haque, 2009). A seguir, um exemplo ilustra essa tarefa de correção da pontuação no texto.

> **Texto com pontuação incorreta:** "vou amar android p sempri sim...vou amar td q a google faz sim...
>
> a gente comprou iphone pq foi o q apareceu c boa oferta de parcelamento né nom"
>
> **Após correção:** "vou amar android p sempri sim... vou amar td q a google faz sim.
>
> a gente comprou iphone pq foi o q apareceu c boa oferta de parcelamento né nom."

4.2.3 Auto Capitalização

Em (Duran et al., 2014), os autores observaram que a correção da capitalização foi a tarefa que mais impactou na precisão de um etiquetador automático. Para ilustrar a importância dessa tarefa considere o seguinte texto:

> "skype é um aplicativo melhor que hangouts"

Nesse exemplo, os termos "skype" e "hangouts", por se tratarem de nomes próprios, deveriam ter sua primeira letra capitalizada. Como não estão capitalizados, um tagueador poderia classificá-los erroneamente como um substantivo. Portanto, o processo de auto capitalização tem como objetivo a correção da capitalização dos termos existentes em um texto.

Neste trabalho, um modelo probabilístico, como o proposto em Nebhi et al. (2015), foi implementado para realizar o processo de auto capitalização de textos. Este modelo consiste na computação das probabilidades de termos 3-grama, considerando se o termo do meio é capitalizado ou não. Dessa forma é possível treinar um classificador com os exemplos e predizer, a partir do contexto das palavras, se aquela palavra central deve ser classificada ou não.

4.2.4 Correção Ortográfica e Gramatical

Sendo as tarefas mais comuns num processo de revisão textual, a correção ortográfica e a correção gramatical são incorporadas na fase final da normalização textual.

As tarefas funcionam de forma similar. A correção ortográfica, que é menos complexa, realiza uma busca no texto por palavras erradas (normalmente palavras fora do vocabulário) e, a partir do contexto da palavra desconhecida ou de uma lista de erros comuns, é computado um ranking de palavras candidatas viáveis para substituição da palavra que contém erros. Por fim, a palavra melhor ranqueada substitui a palavra desconhecida.

A segunda tarefa acontece logo em sequência. A correção gramatical funciona, normalmente, com uma grande lista de regra gramaticais a serem respeitadas. Contudo, a sugestão de solução automática é complexa, pois há a necessidade de se computar corretamente todas as características da sentença (p. ex., o tempo verbal regente).

Neste trabalho, a implementação do corretor ortográfico foi realizada segundo a proposta apresentada por Avanço et al. (2014). Já a implementação do corretor gramatical, proposto por Moura Silva (2013), foi obtida em `http://cogroo.sourceforge.net/`.

A seguir, dois exemplos ilustram as respectivas tarefas.

1. **Texto com erro ortográfico:** “Vou amar android para sempri sim”.

   Após correção: “Vou amar android para sempre sim”.

2. **Texto com erro gramatical:** “Nós ama android pra sempre sim”.

   Após correção: “Nós amamos android para sempre sim”.

4.3 Tokenização

O procedimento de tokenização é responsável pela segmentação de sentenças e termos de um texto. Ainda, por se tratar de textos informais, são consideradas as possibilidades de tokenização de *hashtags* e menções (p. ex.: “@DilmaRousseff” é segmentada em "Dilma Rousseff") e tratamento de termos do tipo URL e *emoticons*.

A tarefa de segmentar *hashtags* e menções utiliza o mesmo conceito de segmentação de sentenças apresentado na Seção 2.2.1. Entretanto, para esse caso, o algoritmo é customizado para segmentar um termo que começe com o caracter “#” ou “@”. No caso das menções de usuários, sendo o nome do perfil conhecido, o algoritmo simplesmente substitui a menção de *username* pelo nome original como um único termo.

A biblioteca utilizada para a implementação deste procedimento foi Apache OpenNLP[1].

O exemplo a seguir ilustra o resultado do procedimento de tokenização (onde a segmentação é representada por uma barra invertida) para uma publicação de um veículo de notícia no Twitter.

> **Texto da publicação**: "Entenda a investigação e o possível crime fiscal do governo Dilma Rousseff http://economia.estadao.com.br/blogs/... (via @estadao e @estadaoEconomia)"

> **Resultado da tokenização (termos segmentados)**: Entenda/ a/ investigação/ e/ o/ possível/ crime/ fiscal/ do/ governo/ Dilma Rousseff/ (/ via/ Estadão/ e/ Economia Estadão/)/

Notam-se os resultados desse processo: tratamento (remoção) da URL e derivação de perfis da rede (@Estadao e @EstadaoEconomia). Ademais, importante citar que termos compostos, como "Dilma Rousseff" e "Economia Estadão", foram agrupados em um único termo nesta abordagem, diferindo de outras abordagens encontradas na literatura.

4.4 Extração de Características

Com os termos e sentenças segmentados, essa etapa do arcabouço extrai uma série de características do texto a partir de tarefas comuns de PLN.

No arcabouço proposto, as características extraídas em nível de termo (ver Tabela 4.1) são: o próprio termo, o lema do termo, o radical (no inglês, *stem*) do termo, a classe morfológica do termo e a anotação de *chunk* correspondente ao termo.

Após a extração dessas características, há possibilidade de filtrá-las via definição de regras (p. ex., escolher somente substantivos e adjetivos para os utilizá-los como características de um texto) ou combiná-las utilizando, por exemplo, um gerador de n-grama baseado em sintagmas nominais (ex.: "governo" + "Dilma Rousseff" = "governo Dilma Rousseff").

A biblioteca Apache OpenNLP foi utilizada na implementação do procedimento de extração de características.

[1] `https://opennlp.apache.org/`

Tabela 4.1: Características textuais extraída na arcabouço proposto

Característica	Descrição
Termo	A característica mais comum de uma sentença é o próprio conjunto de termos
Lema	O procedimento de lematização é a transformação do termo em sua representação mais simples, desconsiderando gênero, número e grau
Radical	O procedimento de *stemming* é a transformação do termo em sua raiz
POS tag	A *POS tag* é a classe morfológica da palavra
Chunk tag	A *Chunk tag* é a classe sintática de um ou mais termos agrupados

4.5 Contextualização

Dado que boa parte das publicações em RSO são curtas e o conteúdo tem carater subjetivo (Petz et al., 2014), o procedimento de contextualização tem como objetivo a expansão da informação presente nas publicações.

O procedimento dá-se a partir de consultas em dicionários semânticos e implementação baseada nos trabalhos (Meij et al., 2012) e (Bontcheva and Rout, 2014)). Neste trabalho, foram utilizados dois dicionários. O primeiro é o de categorias da Wikipédia. Os sintagmas nominais existentes em uma publicação são consultados em um índice onde foram armazenados os títulos de páginas da Wikipédia e categorias relevantes. Havendo uma categoria como resultado dessa consulta, a mesma é adicionada como característica do texto. O segundo dicionário é o LIWC, que fornece a polaridade das palavras. Todos os adjetivos existentes em uma publicação são consultados nesse dicionário e, havendo polaridade negativa ou positiva para um adjetivo, ele é marcado com a sua respectiva polaridade.

As ferramentas e recursos utilizados para a implementação desse procedimento foram a biblioteca Apache Lucene [2], o corpus de artigos da Wikipédia e o dicionário LIWC.

O exemplo a seguir ilustra o resultado do procedimento de contextualização (novas características que foram adicionadas ao texto) para a uma publicação realizada no Twitter.

> **Publicação**: "Entenda a investigação e o horrível crime fiscal do governo Dilma Rousseff (via Estadao e Estadao Economia)"
>
> **Características adicionadas**:
>
> - "investigação" deriva a característica "Categoria: Pesquisa"

[2] https://lucene.apache.org/core/

- "horrível" deriva a característica "LIWC: Negativo"
- "crime" deriva a característica "Categoria: Crimes"
- "Dilma Rousseff" deriva a característica "Categoria: Política do Brasil"

Em todas consultas aos dicionários, termos que trouxeram resultados ambíguos não foram considerados.

4.6 Transformação e Redução de Dados

O último procedimento do arcabouço proposto é responsável por transformar cada texto (publicação) em um vetor numérico que representa todas as suas características (obtidas na etapa anterior). Além disso, procedimentos como discretização e seleção de atributos podem ser realizados para adequar as bases de dados que serão utilizadas na etapa de mineração de dados.

A técnica de vetorização de textos proposta por Soucy and Mineau (2005) foi utilizada para transformar, a partir das características extraídas, os textos em vetores de atributos, permitindo a construção das bases de dados utilizadas no treinamento de classificadores.

O ferramental utilizado para implementar esse procedimento pode ser encontrado na biblioteca Smile [3].

4.7 Exemplo de Aplicação

Para exemplificar o processamento de uma publicação utilizando-se o arcabouço proposto, seguem os resultados de cada um dos seus procedimentos para a seguinte publicação hipotética:

> "RT @paulim123: esse comercial da #heineken em são paulo com mulheres cantando por homens q bebe + conscientemente e ótm :P"

O procedimento de detecção de linguagem recebe como entrada a publicação original e retorna o resultado confirmando que o texto é da língua portuguesa.

Desse modo, o texto segue para o procedimento de normalização textual, onde são executadas em sequência as tarefas de (a) limpeza textual, (b) tratamento de gírias e

[3] `https://github.com/haifengl/smile`

abreviações comuns na internet, (c) auto capitalização, (d) correção de pontuação, (e) correção ortográfica e (f) correção gramatical. O resultado da normalização textual é apresentado a seguir, onde as letras (a), (b), ..., (f) indicam a tarefa de normalização que foi executada naquele ponto do texto.

> "(a) Esse (c) comercial da #Heineken (c) em São Paulo (c) com mulheres cantando por homens (f) que (b) bebem mais (b) conscientemente e ótimo (b). (d) :P"

O resultado da normalização serve como entrada para o procedimento de tokenização, que fornece como resultado a sentença segmentada no seguinte conjunto de termos (separados por /):

> Esse/ comercial/ da/ #Heineken/ em/ São Paulo/ com/ mulheres/ cantando/ por/ homens/ que/ bebem/ mais/ conscientemente/ e/ ótimo/ ./ :P/

O resultado da tokenização alimenta a próxima etapa do processo, onde o procedimento de extração de característica é aplicado gerando o resultado apresentado na Figura 4.2.

Após a extração das características, o procedimento de contextualização complementa o texto com significados a partir de dois dicionários semânticos, incorporando as características: (a) categorias de conceitos encontrados no texto e (b) polaridade dos adjetivos. O resultado desse procedimento é apresentado a seguir.

> Heineken = Cervejaria (a), Cerveja (a), São Paulo = Unidades federativas do Brasil (a), Região Sudeste do Brasil (a), Ótimo = Palavra Positiva (b).

O último procedimento, denominado Transformação e Redução de Dados, converte o texto em um vetor de atributos (ver Figura 4.3) utilizando os termos e algumas características obtidas por meio dos procedimentos anteriores, como por exemplo o lema das palavras (p. ex. "mulher" e "cantar"), o radical das palavras que não possuiam o lema (p. ex. "conscient"), o sintagma nominal primário (p. ex. "Esse comercial Heineken") e resultados provenientes das consultas nos dicionários semânticos (p. ex. "WIKI_Cerveja").

Os valores contidos no vetor de atributos representam o grau de relevância (a partir da medida tf.idf) de cada termo do vocabulário no texto em específico. Esse vetor numérico irá representar cada uma das instâncias (textos) na base de dados que será utilizada na etapa de mineração de dados.

Pos ▲	Termo	Lema	Radical	POS	Chunk
0	Esse	esse	ess	pronome	B-NP
1	comercial	comercial	comercial	adj	I-NP
2	da	de	da	prep	B-PP
3	Heineken		heineken	nome	I-NP
4	em	em	em	prep	B-PP
5	São Paulo			nome	B-NP
6	com	com	com	prep	B-PP
7	mulheres	mulher	mulh	subs	B-NP
8	cantando	cantar	cant	verbo	B-VP
9	por	por	por	prep	B-PP
10	homens	homem	homens	subs	B-NP
11	que	que	que	pronome	B-NP
12	bebem	beber	beb	verbo	B-VP
13	mais	mais	mais	adv	B-ADVP
14	conscientemente	consciente	conscient	adv	B-ADVP
15	e	e	e	conj-c	O
16	ótimo	ótimo	otim	adj	B-NP
17	.	.	.	.	O

Figura 4.2: Exemplo do procedimento de extração de características de uma publicação

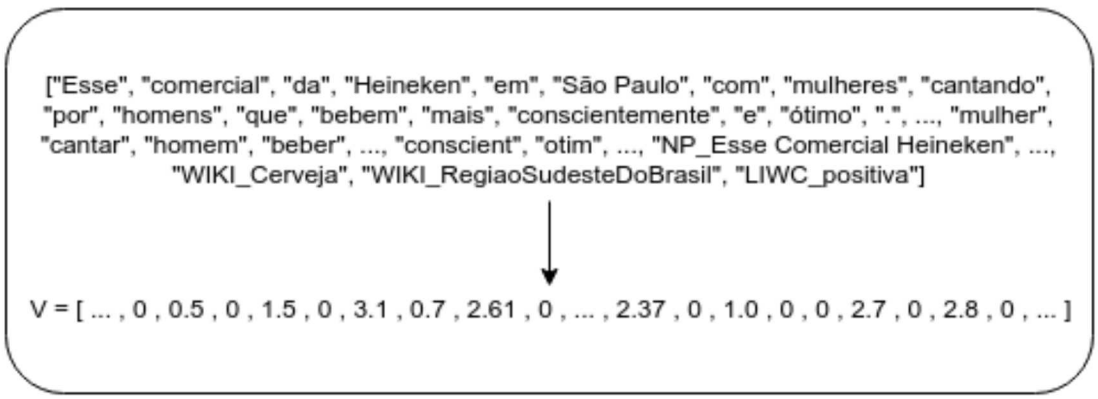

Figura 4.3: Exemplo da vetorização de uma publicação

Capítulo 5

Experimentos Computacionais

Com o objetivo de validar o arcabouço proposto neste trabalho, este capítulo apresenta uma avaliação da mesma no processamento de textos envolvendo as aplicações de Categorização de Texto (CT) e Mineração de Opinião (MO).

A tarefa de CT, que foi realizada a partir de textos provenientes da plataforma Twitter, pode ser considerada um grande desafio, dado que as publicações (*tweets*) desse microblog possuem limitação de caracteres, deixando-as com pouco contexto, muitas abreviações e erros ortográficos e gramaticais.

Já na tarefa de MO, que utilizou textos de avaliações de produtos, geralmente o desafio não está relacionado com limitação de caracteres do texto, mas sim com o grau de abstração do mesmo, uma vez que a solução do problema depende da identificação do significado das palavras no contexto em questão.

Os experimentos computacionais foram realizados em um PC Intel Xeon E5-2690-V3 Dodeca-Core com Linux Ubuntu 14.04 (64 bits) e 512GB RAM. As bases de dados, os recursos utilizados na implementação do arcabouço e uma API que o implementa estão disponíveis em `https://github.com/mstiilpenj/brs-nlp-api`.

A seguir, na Seção 5.1, apresentam-se os experimentos referentes à tarefa de CT e, na Seção 5.2, são descritos os experimentos relacionados com tarefa de MO.

5.1 Categorização Textual

O primeiro experimento utilizando o arcabouço proposto envolve a tarefa de Categorização Textual. Resumidamente, ela consiste em classificar um texto em uma categoria

específica. Por exemplo, dada uma coleção de notícias pertencentes a duas categorias (saúde e tecnologia), deseja-se classificar cada notícia em uma dessas duas categorias.

A seguir, serão apresentados a coleção de dados utilizada na formação das bases de dados adotadas nesses experimentos, os cenários avaliados e os resultados obtidos.

5.1.1 Coleção de Dados

A primeira coleção de publicações utilizada para avaliação do arcabouço foi cedida por uma grande agência de comunicação brasileira que monitora e classifica diariamente milhares de publicações provenientes de redes sociais para gerar relatórios e análises de marcas de produtos para seus clientes. Entretanto, tendo em vista o grande volume de dados existentes nas redes sociais, a classificação manual das publicações se torna um trabalho maçante e passível de erro. Portanto, é evidente a necessidade de auxílio computacional para automatizar esse processo.

A coleção de dados utilizada corresponde a uma amostra composta por 600 publicações relacionadas ao tema "cerveja" postadas no Twitter nos anos de 2013, 2014 e 2015. As classes associadas a essas publicações são: saúde e bem-estar, conhecimento cervejeiro, economia, meio ambiente, cultura/comportamento e consumo responsável. A Tabela 5.1 apresenta cada uma dessas classes e exemplos de publicações das mesmas.

Tabela 5.1: Exemplos de publicações relacionadas ao tema "cerveja"

Classe	**Número de posts**	**Exemplo**
Saúde e Bem-estar	100	Cerveja hidrata igual à água após prática esportiva. Estudo feito na espanha comprovou que o consumo moderado da bebida após exercícios é benéfico para a saúde.
Conhecimento Cervejeiro	100	Amanhã mais uma vez darei a aula inaugural do Curso de Sommelier de Cerveja do Science of beer
Economia	100	RT @jornalextra: Cerveja e refrigerantes devem ficar mais caros neste Natal - A Associação Brasileira de Supermercados...
Meio Ambiente	100	Reciclagem de Caixas de Cerveja viram Biblioteca...
Cultura / Comportamento	100	Curiosidades YES! Alguém já ouviu falar na cerveja irlandesa Guinnes?...
Consumo Responsável	100	A cultura do Álcool - Diga NÃO as drogas! Hoje podemos ver nitidamente uma cultura de quase adoração sobre o álcool...

A amostra de publicações que compõem a coleção de dados utilizada neste trabalho foi selecionada utilizando-se os seguintes critérios: a) somente publicações igualmente rotuladas manualmente por dois anotadores foram consideradas; b) buscaram-se publicações com diversidade de conteúdo; c) as publicações foram escolhidas visando-se o balanceamento de classes nas bases de dados.

5.1.2 Configuração Experimental

A Tabela 5.2 apresenta os diferentes cenários que foram avaliados a fim de comprovar a hipótese de que as etapas presentes no arcabouço proposto são importantes para alcançar resultados melhores do que aqueles obtidos a partir de diferentes estratégias apresentadas na literatura.

Sendo assim, três cenários (1 a 3) envolvendo estratégias comumente adotadas na literatura são comparados com outros três cenários (4 a 6) relacionados com o arcabouço.

Tabela 5.2: Cenários propostos a fim de avaliar o arcabouço

Cenário	**Definição**
1	Textos não são pré-processados.
2	Padronização dos termos: remoção de acentos, caracteres especiais e redução das letras para caixa baixa.
3	Cenário 2, incluindo a técnica de *stemming*.
4	Implementação de todos os procedimentos do arcabouço (com exceção da Contextualização).
5	Implementação de todos os procedimentos do arcabouço (com exceção da Contextualização), utilizando uma heurística que seleciona atributos baseada na saída do procedimento de Extração de Características.
6	Cenário 5 com a adição dos resultados do procedimento de Contextualização.

A heurística utilizada nos cenários 5 e 6 seleciona um subconjunto de termos (baseado em suas características) para compor o vetor de atributos. Para a tarefa de CT, o vetor de atributos é composto por termos classificados como substantivos, nomes próprios, adjetivos, advérbios, verbos e hashtags.

O classificador utilizado na avaliação dos cenários foi o SVM com *kernel* linear. Ele foi escolhido por ser uma técnica que frequentemente apresenta bons desempenhos em trabalhos relacionados com mineração de textos (Irfan et al., 2015; Joachims, 1998). Objetivando encontrar a melhor configuração de parâmetros para cada base de dados resultante dos cenários de pré-processamento (ver Tabela 5.2), testes preliminares foram conduzidos empregando a técnica de validação cruzada e variando o parâmetro $C = 10^Z$,

com $Z \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ (Genkin et al., 2007). A implementação do SVM utilizado nos experimentos foi obtida na ferramenta Smile[1].

5.1.3 Resultados

A Tabela 5.3 apresenta os resultados experimentais obtidos para cada um dos cenários descritos anteriormente. A primeira coluna indica o cenário de pré-processamento utilizado. Da segunda até a quarta coluna são apresentados os resultados médios de desempenho preditivo em termos percentuais, a saber: precisão, revocação, e medida-F (F1) (Manning and Schütze, 1999), todos eles incluindo o intervalo de confiança. Na quinta coluna, tem-se o tamanho do vetor de atributos resultante do pré-processamento realizado no cenário em questão. Os resultados de desempenho preditivo apresentados nesta tabela foram obtidos a partir de 500 repetições do método 10-validação cruzada, sendo que, para cada repetição, uma semente diferente foi utilizada pelo método 10-validação cruzada. Desse modo, os resultados aqui apresentados correspondem a médias de 5000 execuções (500 x 10-validação cruzada).

Tabela 5.3: CT: Resultados para diferentes cenários utilizando o classificador SVM

Cenário	**Precisão**	**Revocação**	**F1**	**Vetor**
1	73,57 ± 0,07	70,23 ± 0,07	71,21 ± 0,08	3718
2	81,08 ± 0,06	79,41 ± 0,06	79,94 ± 0,06	2933
3	80,98 ± 0,05	79,46 ± 0,06	79,99 ± 0,06	2358
4	81,23 ± 0,06	79,38 ± 0,06	79,87 ± 0,06	2497
5	82,47 ± 0,06	80,31 ± 0,05	80,77 ± 0,05	2461
6	**82,64 ± 0,05**	**80,95 ± 0,06**	**81,40 ± 0,05**	2828

A partir dos resultados mostrados na Tabela 5.3, pode-se verificar que todos os cenários obtiveram desempenhos melhores do que aquele que não realiza nenhum pré-processamento do texto (cenário 1). Além disso, observa-se que todos os cenários relacionados com o arcabouço proposto (4 a 6) alcançaram desempenho preditivo superior àqueles obtidos por outras estratégias apresentadas na literatura (cenários 1 a 3). A Figura 5.1 apresenta a distribuição dos resultados por cada classe do cenário vencedor.

A análise de erros se deu ao percorrer textos que foram classificados incorretamente no cenário 1 e que foram classificados de forma correta no cenário 6. Notam-se características (vide exemplos a seguir) de informalidade textual proveniente do costume de canais sociais, o que remete um indicativo de que as técnicas de normalização foram precisas em

[1]https://github.com/haifengl/smile

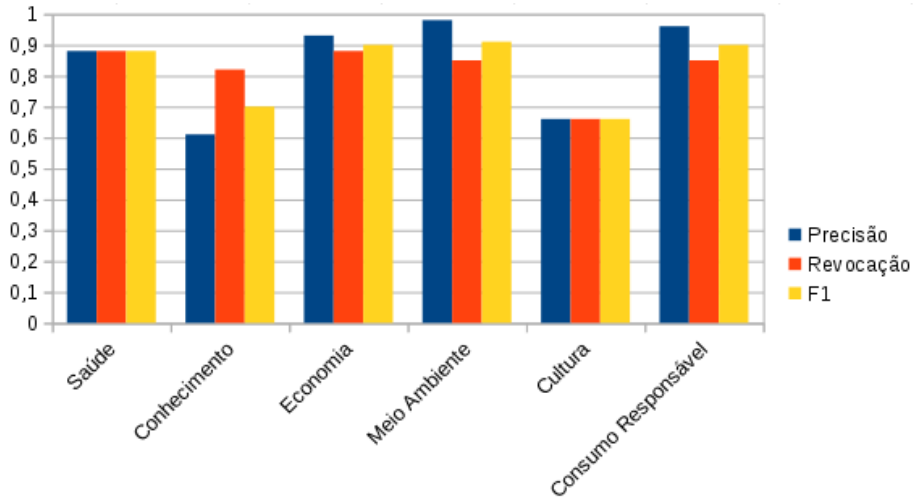

Figura 5.1: CT: Resultados do melhor cenário segmentado por classes

melhorar o desempenho do modelo preditivo. Ademais, duas classes tiveram publicações de caráter ambíguo entre si (Conhecimento Cervejeiro e Cultura/Comportamento), o que fez o desempenho da medida F1 piorar consideravelmente.

> "História da cerveja muito legalzzzzz"

> "@itsneco Vc+cerveja+copa Não rima com estudo..kkkkkkkkkkk"

> "RT @ORaul: sabe nada, Bri RT @entojo: pra ser sommelier de cerveja basta falar 3 vezes bem rápido papilas gustativas retrogosto aftertaste"

Além disso, de acordo com o teste estatístico *t-Student*, é possível afirmar, com um nível de confiança de 95%, que o melhor cenário envolvendo o arcabouço proposto (cenário 6) alcançou um valor significativamente maior do que aquele obtido pelo melhor cenário que não contempla o arcabouço (cenário 3).

5.2 Mineração de Opinião

O segundo experimento utilizando o arcabouço proposto envolve a tarefa de Mineração de Opinião. De modo conciso, essa tarefa consiste em classificar uma opinião expressa em um texto, por exemplo, como positiva ou negativa.

5.2.1 Coleção de Dados

A coleção de dados utilizada neste experimento foi obtida a partir do trabalho (Santos and Ladeira, 2014), em que os autores realizaram uma coleta de avaliações textuais de aplicativos móveis na loja virtual Google Play. Nesse trabalho, 10.000 avaliações foram rotuladas manualmente pelos autores nas classes positiva, neutra e negativa.

Inspecionando as avaliações e suas respectivas classes na base dados do trabalho (Santos and Ladeira, 2014), notou-se que a classe neutra possuía diversas avaliações cuja anotação manual era questionável. A título ilustrativo, o texto "Extensão paga não funciona O Melhor" parece ser uma avaliação negativa, mas foi rotulado como neutra. Devido a esse fato, utilizou-se apenas uma amostra das instâncias das classes positiva e negativa da base de dados adotada no trabalho (Santos and Ladeira, 2014). De modo a trabalhar com uma base de dados balanceada, 815 instâncias de cada uma das classes (positiva e negativa) foram aleatoriamente selecionadas. A Tabela 5.4 apresenta cada uma dessas classes e exemplos de publicações das mesmas.

Tabela 5.4: Exemplos de avaliações de aplicativos móveis

Classe	**Número de Avaliações**	**Exemplo**
Positiva	815	E muito legal o jogo angri birds do espace...
Negativa	815	Essa porcaria nao grava o jgo de onde agente para perdi 5horas de jogo no chrono trigger essa merda...

5.2.2 Configuração Experimental

Os cenários avaliados são exatamente os mesmos adotados no experimento apresentado na Seção 5.1 (ver Tabela 5.2). Do mesmo modo, a heurística utilizada nos cenários 5 e 6 seleciona um subconjunto de termos (baseado em suas características) para compor o vetor de atributos. No entanto, para a tarefa de OM, o vetor de atributos é composto por termos classificados como substantivos, nomes próprios, adjetivos, pronomes, verbos, advérbios, conjunções e interjeições.

Novamente, o classificador utilizado na avaliação dos cenários foi o SVM com kernel linear. Visando encontrar a melhor configuração de parâmetros para cada base de dados resultante dos cenários de pré-processamento, testes preliminares foram conduzidos empregando a técnica de validação cruzada e variando o parâmetro $C = 10^Z$, com $Z \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ (Genkin et al., 2007). A implementação do SVM

utilizado nos experimentos foi obtida na ferramenta Smile.

5.2.3 Resultados

A Tabela 5.5 apresenta os resultados experimentais. A primeira coluna indica o cenário de pré-processamento utilizado. Da segunda até a quarta coluna, são apresentados os resultados de desempenho preditivo médio em termos percentuais, a saber: precisão, revocação e medida-F (F1) Manning and Schütze (1999), todos eles incluindo o intervalo de confiança. Na quinta coluna, tem-se o tamanho do vetor de atributos resultante do pré-processamento realizado no cenário em questão. Os resultados de desempenho preditivo apresentados nessa tabela foram obtidos a partir de 500 repetições do método 10-validação cruzada, sendo que, para cada repetição, uma semente diferente foi utilizada pelo método 10-validação cruzada. Desse modo, os resultados aqui apresentados correspondem a médias de 5000 execuções (500 x 10-validação cruzada).

Tabela 5.5: MO: Resultados para diferentes cenários utilizando o classificador SVM

Cenário	**Precisão**	**Revocação**	**F1**	**Vetor**
1	92,05 ± 0,03	92,04 ± 0,03	92,04 ± 0,03	5713
2	92,26 ± 0,02	92,25 ± 0,02	92,25 ± 0,02	3302
3	93,18 ± 0,02	93,17 ± 0,02	93,17 ± 0,02	2533
4	93,53 ± 0,02	93,52 ± 0,02	93,52 ± 0,02	3463
5	94,53 ± 0,02	94,52 ± 0,02	94,52 ± 0,02	3434
6	**95,52 ± 0,02**	**95,51 ± 0,02**	**95,51 ± 0,02**	4399

O resultado desse experimento, mostrado na Tabela 5.5, foi semelhante àquele apresentado na Seção 5.1, ou seja, o pré-processamento do texto permitiu um aumento do F1 quando comparado ao cenário sem um pré-processamento textual (cenário 1). Pode-se observar que o melhor resultado (cenário 6) alcançou F1 igual a 95,51%, enquanto o cenário sem pré-processamento obteve F1 igual a 92,04%. Essa melhoria significa que, em média, 58 avaliações a mais foram classificadas corretamente quando comparado com o cenário sem os procedimentos do arcabouço proposto. Vale ressaltar que os cenários 4 e 5 também obtiveram desempenho preditivo superior, se comparados com os cenários 1, 2 e 3. A Figura 5.2 apresenta a distribuição dos resultados por cada classe, considerando o cenário vencedor.

A análise de erros se deu ao percorrer textos que foram classificados incorretamente no cenário 1 e que foram classificados de forma correta no cenário 6. Notam-se características (vide exemplos a seguir), de informalidade textual (principalmente erros de digitação) e

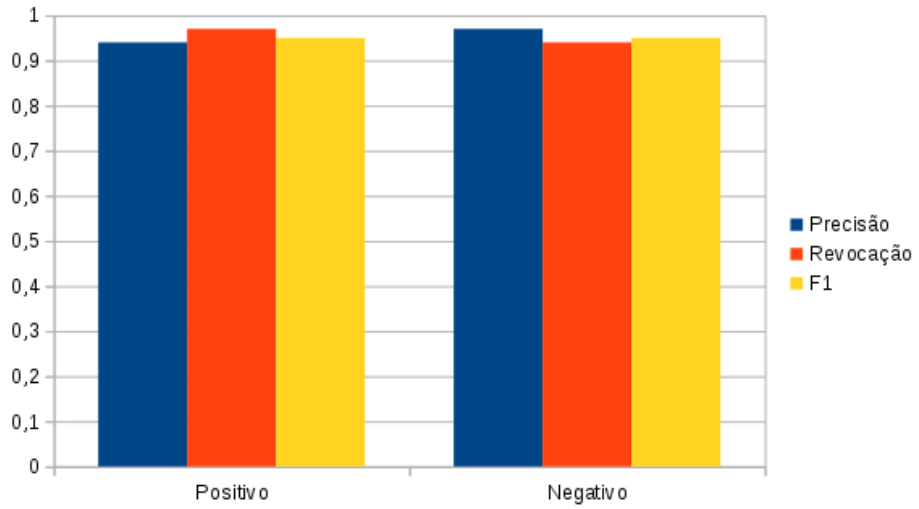

Figura 5.2: MO: Resultados do melhor cenário segmentado por classes

de caráter ambíguo, o que faz com que as tarefas de correção e contextualização do texto contribuam para se alcançar um desempenho preditivo superior.

> "Mto bm É mto bm mas n da d baixar o q faço?"

> "Bonsin É legau de joga com duas pessoas „"

> "Naun Baixeii Ainda Mais Meu Namorado Tem No Android Dele, i eh Super Divertidoo."

Por fim, de acordo com o teste estatístico *t-Student* (com nível de confiança de 95%), o melhor cenário envolvendo o arcabouço proposto (cenário 6) alcançou um valor de F1 significativamente superior àquele obtido pelo melhor cenário que não contempla a arcabouço proposto (cenário 3).

Capítulo 6

Conclusão

Como mencionado em Derczynski et al. (2015), o problema de PLN de textos de redes sociais em língua diversa da inglesa é desafiador, dado que há poucas ferramentas linguísticas desenvolvidas para esse caso. Não bastasse isso, a estrutura textual em redes sociais é hostil, isto é, há muito informalismo e uma falta de contexto nas publicações das principais plataformas *online*.

Neste trabalho, propôs-se um arcabouço que corresponde a uma adaptação de todo o processo de PLN utilizado em textos formais na língua portuguesa, reunindo diferentes técnicas de PLN espalhados por diferentes trabalhos reportados na literatura relacionada. A proposta contribui com conhecimento e especialidade na área de processamento de textos curtos e informais publicados em português brasileiro.

O arcabouço proposto foi avaliado para as aplicações de Categorização Textual (CT) e Mineração de Opinião (MO). Em CT, uma coleção de *tweets* relacionados ao tema "cerveja" formou as bases de dados utilizadas nos experimentos. Em OM, os experimentos foram elaborados com bases de dados de contendo avaliações de produtos provenientes da loja virtual Google Play.

Em ambas as tarefas, o arcabouço proposto mostrou que a realização do pré-processamento dos textos possibilita a melhoria do desempenho preditivo do classificador. Ademais, os resultados obtidos em cada cenário demonstrou que todos os módulos propostos pelo arcabouço contribuíram para o aumento da performance preditiva do classificador.

Como trabalhos futuros, pretende-se avaliar o arcabouço a partir de publicações associadas a diferentes domínios e em outras tarefas, tal como reconhecimento de entidades nomeadas. Além disso, uma avaliação extrínseca mais detalhada do módulo de Normalização Textual, aplicado a diferentes tipos de tarefas, faz-se importante para mensurar a

relevância de cada tarefa realizada nesse módulo.

Referências Bibliográficas

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer International Publishing, 1 edition.

Avanço, L. V., Duran, M. S., and Nunes, M. d. G. V. (2014). Towards a phonetic brazilian portuguese spell checker. pages 24–31, São Carlos, Brasil.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., and Aswani, N. (2013). Twitie: An open-source information extraction pipeline for microblog text. In *Recent Advances in Natural Language Processing*, pages 83–90, Hissar, Bulgaria.

Bontcheva, K. and Rout, D. (2014). Making sense of social media streams through semantics: a survey. *Semantic Web*, 5(5):373–403.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., and Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Dey, L. and Haque, S. M. (2009). Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):205–226.

Duran, M. S., Avanço, L. V., Aluísio, S. M., Pardo, T. A., and Nunes, M. G. (2014). Some issues on the normalization of a corpus of products reviews in portuguese. *European Chapter of the ACL*, page 22.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996). *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, 1 edition.

Genkin, A., Lewis, D. D., and Madigan, D. (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49:291–304(14).

Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27–38.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 368–378, Portland, Oregon.

Hartmann, N. S., Avanço, L. V., Balage, P. P., Duran, M. S., Nunes, M. G., Pardo, T. A., and Aluísio, S. M. (2014). A large corpus of product reviews in portuguese: Tackling out-of-vocabulary words. pages 3865–3871, Reykjavik, Iceland.

Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., Kolodziej, J., Wang, L., Chen, D., Rayes, A., et al. (2015). A survey on text mining in social networks. *The Knowledge Engineering Review*, 30(02):157–170.

Joachims, T. (1998). *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142. Berlin, Heidelberg.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, 1st edition.

Liu, F., Weng, F., and Jiang, X. (2012). A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1035–1044.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Meij, E., Weerkamp, W., and de Rijke, M. (2012). Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 563–572, Seattle, WA, USA. ACM.

Monroe, R. T., Kompanek, A., Melton, R., and Garlan, D. B. (1996). Architectural styles, design patterns, and objects. *IEEE software*, 14:43–52.

Moura Silva, W. D. C. d. (2013). Aprimorando o corretor gramatical cogroo. Master's thesis, Universidade de São Paulo, São Paulo, Brasil.

Nebhi, K., Bontcheva, K., and Gorrell, G. (2015). Restoring capitalization in# tweets. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 1111–1115, Florence, Italy.

Nielsen (2014). The digital consumer. Technical report, Nielsen.

Oliveira, D. M., Laender, A. H., Veloso, A., and da Silva, A. S. (2013). Fs-ner: a lightweight filter-stream approach to named entity recognition on twitter data. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 597–604, Rio de Janeiro, Brazil.

Paiva, V., Rademaker, A., and de Melo, G. (2012). Openwordnet-pt: An open Brazilian Wordnet for reasoning. In *Proceedings of COLING 2012: Demonstration Papers*, pages 353–360, Mumbai, India.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count (liwc): A computerized text analysis program. *Mahwah (NJ)*, vol. 7.

Petz, G., Karpowicz, M., Fürschuß, H., Auinger, A., Stříteskỳ, V., and Holzinger, A. (2014). Computational approaches for mining user's opinions on the web 2.0. *Information Processing & Management*, 50(6):899–908.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, UK.

Santos, F. L. and Ladeira, M. (2014). The role of text pre-processing in opinion mining on a social media language dataset. In *Intelligent Systems (BRACIS)*, pages 50–54, São Carlos, Brasil.

Sholom M. Weiss, Nitin Indurkhya, T. Z. (2015). *Fundamentals of Predictive Text Mining*. Texts in Computer Science. Springer-Verlag London, 2 edition.

Shuyo, N. (2010). Language detection library for java. http://code.google.com/p/language-detection/. [Online; acessado 18-Junho-2015].

Soucy, P. and Mineau, G. W. (2005). Beyond tfidf weighting for text categorization in the vector space model. In *International Joint Conference on Artificial Intelligence*, volume 5, pages 1130–1135, Edinburgh, Scotland.

Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., and Ng, A. Y. (2016). Neural language correction with character-based attention. *Computing Research Repository (CoRR)*, abs/1603.09727.