CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

TRATAMENTO DE TERMOS AUSENTES EM LÉXICOS EM TAREFAS DE ANÁLISE DE SENTIMENTOS

Gabriel Nascimento dos Santos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientador:
Gustavo Paiva Guedes e Silva

Rio de Janeiro,

Agosto, 2018

Tratamento de Termos Ausentes em Léxicos em Tarefas de Análise de Sentimento

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Gabriel Nascimento dos Santos

Aprovada por:

Presidente, Prof. Gustavo Paiva Guedes e Silva, D.Sc. (orientador)

Prof. Eduardo Soares Ogasawara, D.Sc.

Prof. Eduardo Bezerra da Silva, D.Sc.

Profª. Lilian Ferrari, D.Sc. (Universidade Federal do Rio de Janeiro)

Rio de Janeiro,

Dezembro 2017

Agradecimentos

RESUMO

Tratamento de Termos Ausentes em Léxicos em Tarefas de Análise de Sentimento

Gabriel Nascimento dos Santos

Orientador:
Gustavo Paiva Guedes e Silva

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do título de mestre.

As traduções automáticas de texto surgiram durante a guerra fria, nos anos 50, motivadas por questões militares. Atualmente esse tipo de tradução faz parte do nosso cotidiano e representa uma importante ferramenta para a comunicação no mundo globalizado, especialmente com a utilização de ferramentas de tradução automática de textos disponíveis em ambiente *web*. No entanto, apesar de tratar-se de uma área com mais de 60 anos de estudos, ainda há diversos desafios a serem superados, o que faz com que esse tipo de processo continue dependente de revisão humana para garantir a qualidade e confiabilidade da tradução. Existem, atualmente, diversas métricas e modelos para avaliar traduções automáticas de textos, dentre as quais, a preferida e mais utilizada é a métrica BLEU. Essa métrica avalia a qualidade das traduções sem considerar aspectos linguísticos e psicológicos das sentenças avaliadas. Nesse cenário, o principal objetivo desse trabalho é criar novos modelos, capazes de identificar e quantificar, em números percentuais, divergências psicológicas e linguísticas em traduções do inglês para o português do Brasil (tradução direta).Para alcançar esse objetivo, este trabalho utiliza uma ferramenta denominada LIWC (*Linguistic Inquiry and Word Count*) para analisar e processar linguagem natural, classificando e contabilizando palavras em categorias psicológicas e linguísticas. Dois modelos foram desenvolvidos com o objetivo de avaliar traduções diretas.Os experimentos foram conduzidos para comparar os dois modelos propostos com a métrica BLEU. Os resultados foram considerados promissores e espera-se que esse estudo possa contribuir com novos trabalhos na área de traduções automáticas de textos.

Palavras-chave:
Traduções automáticas de textos;Divergências psicológicas e linguísticas; Tradução direta.

Rio de Janeiro,

Agosto 2018

ABSTRACT

Analysis of Psycholinguistic Changes in Automatic Texts Translations

Rafael Guimarães Rodrigues

Advisor:
Gustavo Paiva Guedes e Silva

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Automatic text translations emerged during the Cold War in the 1950s, motivated by military issues. Currently this type of translation is part of our everyday life and represents an important tool for communication in the globalized world, especially with the use of machine translation tools available in the web environment. However, despite being an area with more than 60 years of study, there are still several challenges to be overcome, which makes this process remain dependent on human revision to guarantee the quality and reliability of the translation. There are, currently, several metrics and models to evaluate automatic text translations, among which, the preferred and most used is the BLEU metric. This metric evaluates the quality of translations without considering linguistic and psychological aspects of the sentences. In this scenario, the main goal of present work is to create new models, capable of identifying and quantifying, in percentage numbers, psychological and linguistic divergences in translations from English to Brazilian Portuguese (direct translation).To achieve this goal, this work uses a tool named LIWC (Linguistic Inquiry and Word Count) to analyse and process natural language, classifying and counting words in psychological and linguistic categories. Two models were developed to evaluate direct translations.The experiments were conducted to compare the two proposed models with the BLEU metric. The results were considered promising and it is expected that this study can contribute with new works in automatic text translations area.

Key words:
Automatic translation of texts; Psychological and linguistic divergences; Direct translation.

Rio de Janeiro,

August, 2018

Nascimento, Gabriel.

Tratamento de Termos Ausentes em Léxicos em Tarefas de Análise de Sentimento / Gabriel Nascimento dos Santos – 2018.

x, 23 f; enc.

Dissertação (Mestrado), Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2018.

Bibliografia: f, 19–23

1. Divergências psicolinguísticas 2. Traduções automáticas de textos I. Título

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

Capítulo I Introdução

O uso e expansão da internet e redes sociais nos últimos anos tem gerado uma grande quantidade de dados textuais de usuários que expressam suas opiniões, posições políticas, sentimentos e experiências [Rosenthal et al., 2015]. Isto tem despertado o interesse de empresas e pesquisadores de Análise de Sentimentos (AS) interessados em extrair estas informações [Pang et al., 2008]. O que as pessoas pensam sempre foi uma peça importante de informação durante o processo de tomada de decisão [Pang et al., 2008].

A identificação de emoções em redes sociais envolve o processamento de dados em textos, como expressões escritas e transcrições de expressões vocais. Este processo geralmente é chamado de *affect detection* (AD) [Calvo and D'Mello, 2010, Ishizuka et al., 2012]. Os métodos de AD são técnicas de Processamento de Linguagem Natural (PLN) que se baseiam em abordagens dimensionais e categóricas [Calvo and Mac Kim, 2013]. As abordagens dimensionais dizem respeito a percepções de expressões de emoção em um espaço vetorial, onde cada dado textual é representado em dimensões como a valência, alerta e dominância [Calvo and Mac Kim, 2013]. As abordagens categóricas trabalham com corpus de textos rotulados com suas expressões emocionais (e.g, felicidade, tristeza, medo, raiva), e quando se baseiam apenas no nível da valência (i.e positiva, negativa e neutra), essa abordagem diz respeito à AS [Calvo and D'Mello, 2010].

A AS, também chamada de Mineração de Opiniões (MO) é um campo de pesquisa de mineração de textos [Medhat et al., 2014]. Ela pode ser definida como a tarefa de identificar sentimentos, opiniões e avaliações positivas e negativas [Wilson et al., 2005]. A AS pode trabalhar em três níveis contextuais: nível de documento, nível de sentença e nível de aspecto. No nível de documento, as tarefas de AS focam em analisar se um documento possui opiniões positivas ou negativas [Medhat et al., 2014]. No nível de sentença o primeiro passo é identificar se a sentença é subjetiva ou objetiva [Medhat et al., 2014]. Se a sentença é subjetiva, a AS em nível de sentença determinará se a sentença expressa opiniões positivas ou negativas [Medhat et al., 2014]. A AS em nível de aspectos visa classificar o sentimento em relação aos aspectos específicos das entidades envolvidas nos textos.

A AS também possui duas abordagens: A AS baseada em Aprendizado de Máquina (ASAM) e a baseada em Léxicos (ASL) [Medhat et al., 2014]. A ASAM pode ser dividida nas abordagens supervisionada e não-supervisionada. Na abordagem supervisionada é preciso de um conjunto

de dados contendo documentos já rotulados para que seja treinado um classificador, enquanto a abordagem não-supervisionada não necessita de documentos rotulados, portanto geralmente é utilizada quando existe a dificuldade em obtê-los [Medhat et al., 2014]. Existem ainda as abordagens híbridas que combinam *features* de métodos baseados em léxicos e métodos de aprendizado de máquina. Diversos clasificadores podem ser utilizados em cascata numa tentativa de alcançar um bom nível de precisão [Prabowo and Thelwall, 2009]. Alguns dos classificadores comumente aplicados para o aprendizado supervisionado são Árvores de Decisão (DT), SVM, Rede Neural (NN), Naïve Bayes e Máxima Entropia (ME) [Ravi and Ravi, 2015].

A abordagem baseada em Léxicos se divide em mais duas abordagens: As que dependem de dicionários de *opinion words* manualmente anotadas, e as baseadas em corpus. A abordagem que se baseia em dicionários possuem dificuldades de encontrar palavras do próprio domínio e de um contexto restrito [Medhat et al., 2014]. A abordagem baseada em corpus se fundamenta na probabilidade de ocorrência de uma *opinion word* um conjunto positivo ou negativo de palavras, realizando uma pesquisa em uma quantidade muito grande de textos em mecanismos de pesquisa [Ravi and Ravi, 2015]. Ela também ajuda a resolver o problema de reconhecer *opinion words* de domínio e seus métodos dependem de padrões sintáticos que ocorrem no corpus [Medhat et al., 2014].

Métodos baseados em léxicos de *opinion words* têm encontrado desafios em análise de textos informais, curtos e cacográficos, como os encontrados em serviços como o *Twitter* [Kiritchenko et al., 2014, Kouloumpis et al., 2011]. Redes sociais possuem muitos erros de ortografia e gírias, tornando o trabalho de AS mais difícil [Nguyen et al., 2015]. Palavras fora do vocabulário são mais comuns nas mídias sociais texto do que tipos de texto mais convencionais [Baldwin et al., 2013]. Além disso, muitos termos técnicos ou específicos de um domínio não estão incluídos em léxicos [Park et al., 2016]. Isto gera um problema comum chamado de *out-of-vocabulary (OOV) words*.

Esta dissertação concentra-se em um esforço de tratar OOV *words* em Léxicos baseados em dicionários de *opinion words* encontrados na literatura, como o LIWC [Pennebaker et al., 2001], ANEW [Bradley and Lang, 1999] e SentiWordNet [Esuli and Sebastiani, 2007]. Em contraste com abordagens que tratam OOV *words* com recursos externos, este trabalho visa resolver este problema considerando a palavra ausente como outra *opinion word* presente no dicionário. Desta forma, foi proposto um algoritmo chamado IKLex (*Input KNN Lexicon*), baseado no algoritmo k-Nearest Neighbor (KNN). O IKLex funciona de forma que dado uma ausência de uma palavra em um léxico, obtém as *features* da palavra mais próxima que esteja presente no léxico.

Para obter similaridades entre palavras são utilizados modelos de *word embeddings*. Os *word embeddings* são representações de palavras como vetores de números reais em um espaço

multidimensional [Turian et al., 2010]. Cada dimensão de um *word embedding* representa uma *feature* latente da palavra, capturando propriedades sintáticas e semânticas [Turian et al., 2010]. *Word embeddings* são tipicamente gerados por modelos de redes neurais [Turian et al., 2010], tais como o Word2Vec [Mikolov et al., 2013], GloVe [Pennington et al., 2014] e o FastText [Bojanowski et al., 2016].

Esta dissertação apresenta como contribuição: (i) o algoritmo IKLex. (ii) um novo conjunto de dados colhido de uma rede social brasileira denominada Meu Querido Diário (MQD). Nesta rede social, os usuários alimentam um diário como uma atividade cotidiana e interagem fazendo comentários. Este conjunto de dados possui como diferencial sua precisão em relação ao rótulo das emoções relacionados aos documentos, pois nesta rede social os próprios usuários podem informar a emoção relacionada a sua publicação no diário.

Os experimentos foram conduzidos utilizando dois conjuntos de dados, sendo um deles o conjunto de dados proveniente do MQD, em Português do Brasil, e outro conjunto de dados em inglês. Também foram utilizados o LIWC como léxico e o *Word2Vec* como modelo para gerar os *word embeddings*. Foram escolhidos três classificadores, que são o SVM com kernel linear (LSVC), o Multinomial Naïve Bayes (MNB) e o Random Forests (RF). É realizado o treinamento dos classificadores antes e depois de aplicar o algoritmo proposto. Os resultados indicam que o uso do IKLex melhora a qualidade dos resultados da classificação.

Esta dissertação está organizada em 7 capítulos. O capítulo II apresenta todo o referencial teórico necessário para o entendimento do assunto abordado nos próximos capítulos. O capítulo III descreve os trabalhos relacionados a esta dissertação. O capítulo **??** discorre sobre a metodologia aplicada neste trabalho. O capítulo **??** apresenta a avaliação dos experimentos. O capítulo V traz as considerações finais e por fim o capítulo VI apresenta o cronograma deste projeto de pesquisa.

Capítulo II Referencial Teórico

Neste capítulo é apresentado todo o embasamento necessário e os conceitos essenciais para o entendimento desta dissertação. O capítulo está organizado como segue. A seção II.1 apresenta o LIWC, que possui um léxico utilizado nesta dissertação. A seção II.3 apresenta o *Word2Vec*, que são arquiteturas de rede neural para gerar *word embeddings*. Por fim, a seção II.5 apresenta os três classificadores utilizados nesta dissertação: *Support Vector Machines*, *Multinomial Naive Bayes* e *Random Forest*.

II.1 LIWC

O LIWC [1] (Linguistic Inquiry and Word Count) é um software proposto por Pennebaker em 2001 que possui o objetivo de extrair e analisar componentes emocionais, cognitivos e estruturais presentes na fala e em textos [Pennebaker et al., 2003]. Segundo os autores, foi concebido com o objetivo inicial de acompanhar melhorias na saúde dos pacientes, que deveriam descrever suas experiências negativas. Em seguida, as características desses textos eram analisadas com base nas características extraídas pelo LIWC. Isso é feito com base em um léxico que classifica cada palavra entre 1 à 64 categorias. Estas categorias podem se referir à classes gramaticais (e.g., pronome, verbo) ou à aspectos psicológicos e/ou cognitivos (e.g., raiva, tristeza). O léxico do LIWC em sua versão 2007 possui 127.149 palavras e *word stems* para português do Brasil [Balage Filho et al., 2013], enquanto o léxico do LIWC em sua versão 2007 na língua inglesa possui cerca de 4500 palavras e *word stems*.

II.2 Soma de vetores de palavra do LIWC

Dado um documento $D = \{w_1, ..., w_n\}$ representado por um conjunto de n termos, e sendo L o conjunto de palavras do léxico do LIWC, $\delta = D \cap L$ representa os termos em comum entre o LIWC e o documento. Em outras palavras, $\delta = \{w_1, ..., w_k\}$ representa um conjunto de *opinion words* do LIWC existentes no documento D. Cada termo $w_i \in \delta$ é interpretado como um vetor $\vec{v}_i = (f_1, ..., f_{64})$ onde estão contidas as 64 *features* fornecidas pelo LIWC 2007. Sendo $TF = (tf_1, ..., tf_k)$ um vetor contendo as frequências dos termos (*term frequencies*) de δ, a soma dos vetores de palavra do

[1] http://liwc.wpengine.com

LIWC pode ser realizada conforme a equação II.1.

$$DLiwc = \sum_{i=0}^{k} \vec{v}_i \cdot tf_i \tag{II.1}$$

II.3 Word2Vec

Word2Vec são arquiteturas de redes neurais propostas por Mikolov et al. [2013] para realizar o aprendizado de *word embeddings*. A figura apresenta as duas arquiteturas fornecidas, as quais possuem diferentes tarefas de aprendizado. As duas se chamam Continuous Bag-Of-Words (CBOW) e Skip-Gram. A arquitetura CBOW recebe como entrada da rede neural um conjunto de palavras representando o contexto no qual a rede deve prever a palavra faltante. A arquitetura Skip-Gram recebe como entrada uma palavra e tenta prever as palavras ao redor dentro de um contexto de tamanho fixo. Mikolov et al. [2013] explica que a arquitetura Skip-Gram gera melhores *word embeddings*, embora a complexidade computacional seja maior. Nesta dissertação foi escolhida a arquitetura Skip-Gram devido a sua maior precisão.

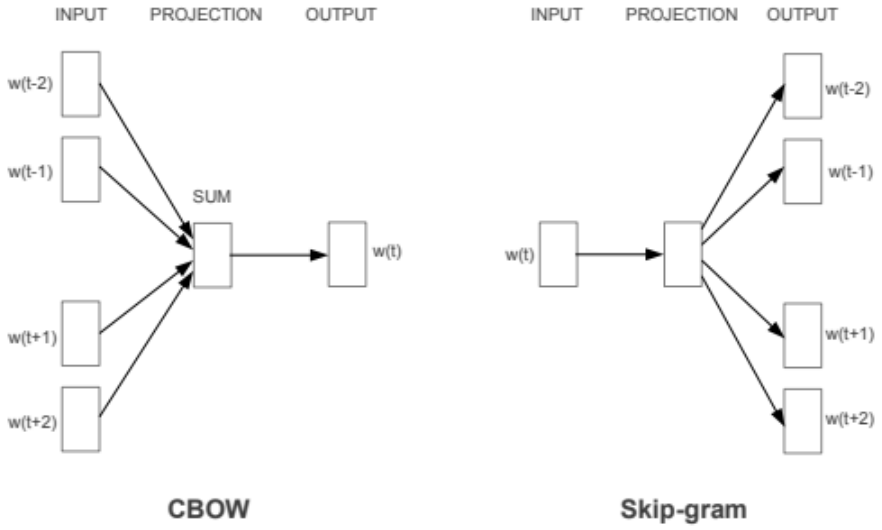

Figura II.1: Arquiteturas do Word2Vec. Retirado de [Mikolov et al., 2013].

II.4 Similaridade por Cosseno

A similaridade entre vetores do mesmo número de dimensões pode se medida através do cálculo do cosseno do ângulo formado entre entre eles. Conforme equação II.2, a similaridade por cosseno entre dois vetores é obtida pelo produto escalar dos vetores dividido pelo produto do módulo dos dois vetores. Diversos trabalhos em Mineração de Textos (Mineração de textos (MT)) costumam utilizar a medida de similaridade por cosseno para calcular a similaridade entre

representações vetoriais. Dentre estes trabalhos, podemos destacar [Ghosh et al., 2015, Maas et al., 2011].

$$cos(A,B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{II.2}$$

II.5 Classificadores

II.5.1 Support Vector Machines

O Support Vector Machines é uma técnica de aprendizado de máquina que consiste na teoria do aprendizado estatístico, desenvolvida por Vapnik [1995]. Este método tem como objetivo resolver um problema de classificação encontrando um hiperplano ótimo em um espaço vetorial de alta dimensionalidade [Cortes and Vapnik, 1995].

O SVM apresenta grande sucesso na tarefa de classificação em texto por ser efetivo em espaços de alta dimensionalidade e em tarefas que a dimensionalidade dos dados é maior que o número de amostras disponíveis [Forman, 2007]. Nesta dissertação será utilizado o classificador SVM com hiperplano linear(LSVC), devido seu menor tempo de treinamento e maior simplicidade.

II.5.2 Random Forest

Random Forest(RF) se trata de um método de aprendizado supervisionado baseado em um conjunto de árvores de decisão que recebem amostras populacionais aleatórias e independentes da mesma distribuição Breiman [2001]. Cada predição de uma árvore em uma floresta tem o poder de um voto [Breiman, 2001]. Após todo este conjunto de árvores realizarem um trabalho de classificação, é escolhida a classe que obteve o maior número de votos [Breiman, 2001].

A generalização da RF depende da força de cada árvore de decisão na floresta e da baixa correlação entre todas as árvores [Breiman, 2001]. Desta forma, além de cada árvore precisar ser um bom classificador, é importante a aleatoriedade de amostras e a independência dos dados de cada árvore.

II.5.3 Naive Bayes

O classificador Naive Bayes se baseia no teorema de Bayes, que presume a independência de variáveis aleatórias [McCallum et al., 1998]. Por isso, dado uma hipótese em um contexto de classificação de textos, todas as *features* ou eventos de um documento, como a ocorrência de palavras, são independentes. Em parte dos casos da vida real essa hipótese de independência não é verdadeira, no entanto, de forma surpreendente este classificador tende a ter um bom desempenho [Lewis, 1998, McCallum et al., 1998].

Existem dois tipos principais de classificadores Naive Bayes utilizados em classificação de textos, dependendo da forma de distribuição dos atributos ou variáveis aleatórias [Lewis, 1998, McCallum et al., 1998]. Em representações mais simples de documentos, como vetores binários indicando a presença ou não de termos no documento, utiliza-se uma variação chamada Bernoulli Naive Bayes, pelo fato dos atributos seguirem uma distribuição de Bernoulli [Lewis, 1998, McCallum et al., 1998]. Para a representação de documentos utilizando uma representação de atributos como uma distribuição discreta de números reais, se utiliza o *Multinomial Naive Bayes* (MNB) [Lewis, 1998, McCallum et al., 1998]. Isto é comum no cenário onde documentos são representados por um vetor contendo números de ocorrências de eventos. Estes eventos geralmente são ocorrências de termos em um documento.

Este trabalho utilizará o MNB, pois os documentos são representados por vetores de valores discretos. Sendo $V = \{w_1, ..., w_n\}$ um conjunto de palavras que forma o vocabulário de tamanho n, $\vec{D} = (x_1, ..., x_n)$ um vetor contendo valores discretos representando *features* de um documento, cada palavra w_i é representada por um valor discreto x_k no documento D. E sendo $C = \{c_1, ..., c_k\}$ as classes possíveis para qualquer documento, o classificador MNB pode ser representado pela equação II.3.

$$\hat{c} = \underset{c_j \in C}{\operatorname{argmax}} P(c_k|D) = P(c_k) \prod_{i}^{n} P(w_i|c_k) \tag{II.3}$$

Expandimos $P(w_i|c_k)$, que representa a probabilidade da palavra w_i pertencer à classe c_k na equação II.4. Sendo T_k os documentos no conjunto de treinamento T atribuídos à classe c_k, então $\sum_{x_i \in T_k} x_i$ é o somatório da *feature* x_i que ocorre nos documentos em T_k. O α representa um hiperparâmetro de suavização (*smoothing*) no classificador para evitar com que probabilidades iguais a zero ocorram. Quando $\alpha = 1$, a suavização é denominada *Laplace smoothing*, enquanto para demais valores onde $\alpha < 1$ a suavização é denominada *Lidstone smoothing* [Tan et al., 2014].

$$P(w_i|c_k) = \frac{x_i + \alpha}{\sum_{x_i \in T_k} x_i + n\alpha} \tag{II.4}$$

Capítulo III Trabalhos relacionados

Neste capítulo, serão descritos os trabalhos relacionados ao estudo desenvolvido nesta dissertação. Para tanto, na seção III.1 são descritos alguns trabalhos que utilizam léxicos para AS e classificação de polaridade. A seção III.2 discorre sobre trabalhos que lidam de alguma forma com o problema de *OOV words*.

III.1 Trabahos que utilizam Léxicos

No contexto de trabalhos que utilizam léxicos, destaca-se o trabalho de Singh et al. [2017]. Neste trabalho, foram analisados os impactos de sentimentos e opiniões e retweets de usuários do *Twitter* ao redor do mundo em um período de desmonetização ocorrida na Índia em 2016, quando o governo Indiano decidiu parar de utilizar as cédulas de 500 e 1000 Rupias, numa tentativa de combater a corrupção e melhorar a economia. Para isto, foi utilizado um conjunto de dados de *tweets* extraídos neste período. Os sentimentos de cada *tweet* foi determinado através do VADER, um léxico de análise de sentimento baseado em regras que é focado para o domínio de sentimentos expressos nas mídias sociais. Também foi realizada uma análise de *retweets*, onde foi feita uma abordagem supervisionada com o objetivo de analisar quais *tweets* seriam *retwitados* ou não. Foram utilizados sete classificadores para esta tarefa, que são: SVM, CART, Regressão Logística, *Multinomial Naive Bayes*, RF, *Bernoulli Naive Bayes* e *KNN*. Foram testadas *features* dos *tweets* como unigramas, bigramas e uma união de unigramas e bigramas. O maior $F1score$ obtido neste trabalho foi com o uso do CART em conjunto com bigramas. No entanto, o desempenho geral dos classificadores foi melhor com o uso de unigramas e bigramas em conjunto.

O trabalho de Keshavarz and Abadeh [2017] possui o objetivo de melhorar a classificação de polaridade de sentimentos em microblogs pela da construção de léxicos de sentimentos adaptativos. Ao invés da geração manual dos léxicos é proposto um algoritmo genético adaptativo denominado ALGA. O algoritmo realiza a tarefa de classificação de polaridade gerando léxicos na fase de treinamento, e após isto estes léxicos são utilizados na fase de teste de classificação de polaridade. Os experimentos foram conduzidos em seis conjuntos de dados do Twitter. Desses seis conjuntos de dados, o ALGA obteve melhor acurácia que o estado da arte em dois, e

melhor *F1 score* em quatro.

O trabalho de Rezapour et al. [2017] se baseia na ideia de que as *hashtags* do *Twitter* são termos importantes que contribuem para transmitir o sentimento de *tweets*. Neste estudo, foram testadas se a inclusão destas *hashtags* em um léxico de sentimento melhora a precisão da tarefa de análise de sentimento. Para validar esta hipótese foram analisados os *tweets* que mencionam os candidatos à presidência dos EUA na eleição de 2016 (Hillary Clinton, Bernie Sanders, Donald Trump, Ted Cruz e John Kasich) durante os 13 dias que antecederam as eleições primárias de Nova York. Após isto, é então proposto um método de análise de sentimentos que verifica a popularidade dos candidatos da eleição pelo número de *tweets* com valência positiva e então verificado se esse método corresponde com os resultados reais da eleição. São utilizados dois léxicos, sendo um deles o LIWC e o outro o proposto por Wilson et al. [2005]. Resultados apontam que o uso de *hashtags* melhoram em até 10% o *F1 score* do léxico proposto por Wilson et al. [2005] ao realizar os experimentos com um conjunto de dados anotado. Os resultados também apresentam que 48% dos *tweets* do conjunto de dados analisado onde mencionavam os candidatos republicanos continham um sentimento positivo em relação a Donald Trump, fazendo dele o vencedor mais provável. Os outros dois candidatos, Ted Cruz e John Kasich, receberam 29% e 23% dos *tweets* positivos. Isso se aproxima dos números de pesquisa reais liberados para a primária de Nova York. Entre as menções dos candidatos democratas, no entanto, Bernie Sanders obteve a maior taxa de tweets com sentimento positivo, contrariando os resultados das eleições primárias.

O trabalho de Tavares and Guedes [2017] utiliza o LIWC em uma tarefa de classificação de filmes com base em suas legendas e informações extraídas de redes sociais. Para isto, foi escolhido o conjunto de dados derivado do IMDB, contendo 5.043 filmes. As classes deste conjunto de dados foram definidas através de um *data binning* das avaliações dos filmes, sendo: excelente ($7,5$ a $10,0$), bom ($5,0$ a $7,4$), ruim ($2,5$ a $4,9$) e péssimo ($1,0$ a $2,4$). Foram selecionados apenas os filmes que tiveram avaliação excelente e ruim, totalizando 172 filmes. Em seguida, foram realizadas duas seleções de atributo distintas no conjunto de dados, sendo uma delas denominada *FILM-172-76*, levando em conta todos os atributos gerados pela contagem dos vetores de palavra do LIWC. A outra foi denominada *FILM-172-15*, levando em conta apenas 20% dos atributos mais relevantes segundo o *Information Gain*, seguindo o princípio de Pareto. Os experimentos foram conduzidos utilizando os classificadores *ZeroR*, *Random Forest*, *Naive Bayes*, *Multinomial Naive Bayes* e *Sequential Minimal Optimization*. Os classificadores que obtiveram melhor resultado com base no *F1 score* foram o *Random Forest* com o *FILM-172-76*, e o *Sequential Minimal Optimization* com o *FILM-172-15*.

III.2 Trabalhos que lidam com *out-of-vocabulary words*

No contexto de trabalhos que lidam com *OOV words* se destaca o de Maity et al. [2016]. Nele, são recolhidos cerca de 1 bilhão de *tweets* para formar um corpus e é proposto um modelo de classificação de *OOV words* em seis categorias (e.g emoticons, alongamento de palavras, expressões, encurtamento de palavras/abreviações, nomes próprios e fusões de palavras). É utilizado o dicionário denominado GNU Aspell[1] como base para detectar as *OOV words*. São então definidas 3.500 *OOV words*, sendo cada uma delas manualmente anotadas por cinco autores em uma das seis categorias, alcançando um *Fleiss' Kappa* de $0,96$ de concordância.

São propostos métodos simples para classificar *emoticons* e alongamentos de palavras. Para classificar *emoticons* são utilizadas simples expressões regulares, alcançando uma acurácia de $98,1\%$, com precisão de $87,7\%$ e *recall* de $97,6\%$. Para detecção de alongamento de palavras, as letras repetitivas são removidas de uma a uma, e é verificado no dicionário GNU Aspell a existência da palavra. Este método alcançou uma acurácia de $93,1\%$, no entanto, a precisão e o *recall* foram de $43,2\%$ e $67,7\%$, respectivamente.

Para classificar as demais categorias é definido um método mais complexo, que se baseia em três tipos de *features*: features lexicais, features de conteúdo e features de contexto. As features lexicais se relacionam com as propriedades lexicais das palavras em torno das OOV words. As *features* de conteúdo se relacionam com o conteúdo dos *tweets* que as OOV words aparecem. Enfim, as *features* de contexto levam em conta informações de posicionamento e localização de várias entidades nos *tweets*.

Os experimentos para classificar as quatro categorias restantes são conduzidos utilizando o classificador SVM e *Logistic Regression*, sob o método *10-fold cross validation*. Também foi adotado o LDA (*Latent Dirichlet Allocation*) para reduzir a dimensionalidade, com diversos números de tópicos, sem grandes impactos nos resultados dos classificadores. Ambos os classificadores tiveram desempenho de classificação muito semelhante (*F1 score* de $79,6\%$, com número de tópicos = 50), porém o *Logistic Regression* obteve melhor área sob a curva ROC em relação ao SVM. Foi observado que as *features* de conteúdo foram as mais significativas para o alcance do resultado, logo há forte diferença semântica entre as seis categorias de *OOV words*.

O trabalho de Hartmann et al. [2014] descreve os procedimentos de pré-processamento realizados para tratar *OOV words* presentes em um conjunto de dados de avaliações de produtos em Português do Brasil. Para construir o conjunto de dados a ser analisado foi efetuado o *crawling* do site Buscapé[2] em Setembro de 2013. Posteriormente, as *OOV words* foram detectadas com o uso do vocabulário *Unitex-PB* [Muniz et al., 2005] e parte delas corrigidas com o uso do GNU As-

[1] http://aspell.net/
[2] https://www.buscape.com.br

pell. Também foram comparadas a quantidade de correções realizadas pelo REGRA, o corretor ortográfico do MS-Office, com a quantidade de correções realizadas pelo GNU Aspell. O REGRA corrigiu $11,51\%$ tokens a menos em relação ao GNU Aspell. A seguir foi medida a precisão de 369 *tags* sintáticas e morfosintáticas informadas pelo analisador sintático Palavras [Bick, 2000], melhorando de $83,73\%$ para $84,28\%$ após aplicar o pré-processamento com o Aspell. Foi realizada uma investigação mais profunda a respeito dos tipos de *OOV words* presentes no corpus. Com este fim, quatro pares de juízes anotaram manualmente 5.575 *tokens*, correspondentes às *OOV words*, indicando quais de 8 categorias (e.g acrônimos, nomes prôprios, gírias de internet) eles se enquadram. Foi alcançado um nível médio de concordância $Kappa$ de 0.752, considerando a categoria de erros de ortografia e um nível médio de concordância $Kappa$ moderado de 0.589, desconsiderando a categoria de erros de ortografia, por ser a mais comum. Por fim, foi desenvolvido um *workbench* para normalização denominado *Lexical Normalization of Product Reviews from the Web*, que utiliza os recursos léxicos produzidos neste trabalho.

Huang et al. [2014] propuseram um método não-supervisionado para detecção de novas *opinion words*. O método proposto extrai padrões lexicais que possuem forte associação estatística com um conjunto de *seed words* contidas em um léxico. Os padrões lexicais extraídos são usados para encontrar outras palavras prováveis ordenadas de forma decrescente (da mais provável para a menos provável). Um conjunto de k novas palavras mais prováveis podem ser adicionadas ao conjunto de *seed words* definido para a próxima iteração. O processo pode ser executado iterativamente até que uma condição de parada seja atendida. Estas novas *opinion words* foram adicionados ao léxico *Hownet*. Os experimentos de classificação de polaridade foram conduzidos em um conjunto de dados manualmente anotado de 4.500 posts do *Weibo* que possuem no mínimo uma *opinion word* contida no léxico *Hownet*. Ao treinar o classificador SVM, o método proposto mostrou um ganho entre 2% a 3% na acurácia. No entanto, este método possui algumas limitações. Primeiramente, é necessário que os termos do corpus possuam *POS (Part-of-Speech) tags*, o que pode gerar dificuldades, uma vez que os *POS-Taggers* também possuem dificuldades em realizar suas tarefas com precisão em corpus originados de redes sociais devido ao vocabulário utilizado [Gimpel et al., 2011]. Outro problema é que o método proposto apenas detecta novos adjetivos, ignorando outras classes gramaticais que seriam importantes para a tarefa de análise de sentimentos.

Este trabalho difere dos demais trabalhos supracitados. Este visa tratar do problema de *OOV words* em léxicos para AS semelhantes ao LIWC utilizando a informação semântica de *word embeddings* de forma completamente não-supervisionada. Diferente do trabalho de Huang et al. [2014], este trabalho não está limitado ao uso de POS-Taggers, ou de uma classe gramatical específica.

Capítulo IV Metodologia

Este capítulo aborda a metodologia aplicada e o passo a passo seguido para a condução dos experimentos. A seção IV.1 descreve os conjuntos de dados utilizados e o pré-processamento realizado. A seção IV.2 aborda a forma de treinamento dos modelos *Word2Vec*. A seção IV.3 expõe os passos do algoritmo proposto para esta dissertação. A seção IV.4 descreve o fluxo de preparação dos dados para os classificadores. Enfim, a seção IV.5 descreve como foi realizado o treinamento dos classificadores.

IV.1 Conjuntos de Dados

Foram selecionados dois conjuntos de dados para realizar os experimentos desta dissertação. Um dos conjuntos de dados está no idioma Português do Brasil e o outro no idioma Inglês. O conjunto de dados em Português do Brasil foi extraído de uma rede social brasileira denominada Meu Querido Diário[1]. Nesta rede social usuários possuem um diário virtual, onde descrevem seus dias, podendo interagir com outros usuários através de comentários. É importante destacar que quando um usuário realiza uma entrada no diário ele possui opção de informar o sentimento referente àquela entrada. Os sentimentos possíveis para inserção são felicidade, tristeza, medo, raiva, nojo e surpresa. Para realizar a classificação de polaridade foram escolhidas apenas o subconjunto contendo as emoções de felicidade e tristeza.

Este conjunto de dados possui 60.314 entradas com as emoções *felicidade* e *tristeza* associadas. Todas as letras maiúsculas foram convertidas em letras minúsculas. Conforme adotado em Chen et al. [2013], os termos com menos de cinco ocorrências foram removidos, o que totalizou um vocabulário com 45.067 termos únicos e $13.744.318$ palavras nestas 60.314 entradas. O conjunto de dados já pré-processado é denominado MQDFT2018. É importante destacar que dos 45.067 termos, apenas 17.444 existem no léxico do LIWC, o que corresponde a $3.577.055$ palavras que não são categorizadas. A partir desse momento, nos referimos às entradas como documentos.

O conjunto de dados em Inglês foi proposto por Pang and Lee [2004] para classificação de polaridades. Este conjunto de dados possui 1000 documentos possuindo valência positiva e 1000

[1] http://www.mqd.com.br

documentos com valência negativa. Todas as letras maiúsculas foram convertidas em letras minúsculas. Os termos com menos de cinco ocorrências foram removidos, fazendo com que o conjunto de dados possua 14.890 termos únicos e $1.472.241$ palavras após o pré-processamento. Dos 14.890 termos únicos o LIWC não possui 13.411, correspondendo a 655.901 palavras não categorizadas. O conjunto de dados já pré-processado é denominado POLARITY-EN.

IV.2 Treinamento do modelo *Word2Vec*

Para cada conjunto de dados foi treinado um modelo *Word2Vec* com a arquitetura *Skip-Gram* para gerar os *word embeddings*. Para realizar o treinamento, alguns hiperparâmetros foram definidos. Os vetores gerados possuem 300 dimensões, conforme utilizado em diversos trabalhos na literatura [Mikolov et al., 2013, Ombabi et al., 2017, Ouyang et al., 2015]. O tamanho do contexto (*window size*) utilizado corresponde a 5, conforme adotado em [Mihaylov and Nakov, 2016].

Termos com menos de cinco ocorrências foram desconsiderados, conforme a seção IV.1. O modelo de *word embeddings* em Português do Brasil, treinado com o MQDFT2018, foi denominado W2VMQD. Por outro lado, o modelo de *word embeddings* em Inglês, treinado com o conjunto de dados *POLARITY-EN*, foi denominado W2VPOL.

IV.3 Input KNN Lexicon

Esta dissertação propõe o algoritmo IV.1, denominado IKLex (*Input KNN Lexicon*). Ele trata as ausências de palavras em um léxico considerando os vetores gerados pelo modelo de *word embeddings*. Dessa maneira, dada uma palavra desconhecida pelo léxico, são obtidas as k palavras com maior similaridade por cosseno (definida na seção II.4) no modelo *Word2Vec* recebido como parâmetro.

Portanto, as *features* da palavra ausente no léxico serão as mesmas *features* da palavra mais próxima que faça parte do léxico. Pode se considerar que na ausência de uma palavra, ela é considerada como a mais próxima semanticamente.

O IKLex funciona da seguinte forma. O passo 1 obtém o vocabulário de todos os documentos do conjunto de dados. Para cada termo w do vocabulário, o passo 3 verifica se o léxico contém a palavra. Caso contenha, o passo 4 atribui as *features* correspondentes no léxico a essa palavra. Caso o léxico não contenha a palavra, o passo 6 obtém os k termos mais próximos (ordenado do mais próximo para o mais distante por meio da similaridade por cosseno) com base no modelo de *word embeddings* WEmb.

Em seguida, para cada termo v contido nos k vizinhos mais próximos, o passo 9 verifica se o termo v está contido no léxico. Caso esteja, o passo 10 atribui as *features* da palavra v no léxico ao termo w e descarta os demais vizinhos mais próximos. Caso nenhum dos termos vizinhos mais

Algorithm IV.1 IKLex(D, lex, $WEmb$, k)
Input:

- D = conjunto de documentos
- lex = léxico de palavras associadas a categorias
- $WEmb$ = Modelo de Word Embeddings
- k = Número de vizinhos mais próximos

Output: $nLex$, léxico modificado

```
1: voc ← obterVocabulario(D)
2: for all w ∈ voc do
3:    if w ∈ lex then
4:       nLex[w] ← lex[w]
5:    else
6:       kVizinhos ← obterKNNOrd(w, k, WEmb)
7:       nLex[w] ← 0⃗
8:       for all v ∈ kVizinhos do
9:          if v ∈ lex then
10:            nLex[w] ← nLex[v]
11:            break
12:         end if
13:      end for
14:   end if
15: end for
16: return nLex
```

próximos esteja contido no léxico, é atribuído um vetor nulo $\vec{0}$ às *features*, conforme inicialização da variável efetuada no passo 7.

IV.4 Preparação dos dados para classificação

A figura descreve como funciona de forma geral o fluxo de preparação dos dados antes de realizar o treinamento dos classificadores para os experimentos. Primeiro será realizado o pré-processamento que foi descrito na seção IV.1, sob os conjuntos de dados utilizados. Em seguida, existe uma decisão no fluxo a se tomar com o conjunto de dados já pré-processado. Caso o IKLEX seja executado, é realizado o treinamento do modelo de *word embeddings Word2Vec* sob o conjunto de dados pré-processado. Este é então utilizado como entrada, junto com o LIWC, para o algoritmo IKLEX. Após a execução do algoritmo IKLex, então a soma de vetores de palavra do LIWC é executada, gerando o conjunto de dados preparado para treinamento nos classificadores. Caso não seja executado o algoritmo IKLEX, então é feita somente a soma de vetores de palavra do LIWC.

Foi determinado para este trabalho que a execução do algoritmo IKLEX será com o hiper-parâmetro $k = 20$, para encontrar os 10 *word embeddings* mais próximos para cada termo ausente. Ao executar o fluxo utilizando os conjuntos de dados MQDFT2018 e POLARITY-EN e

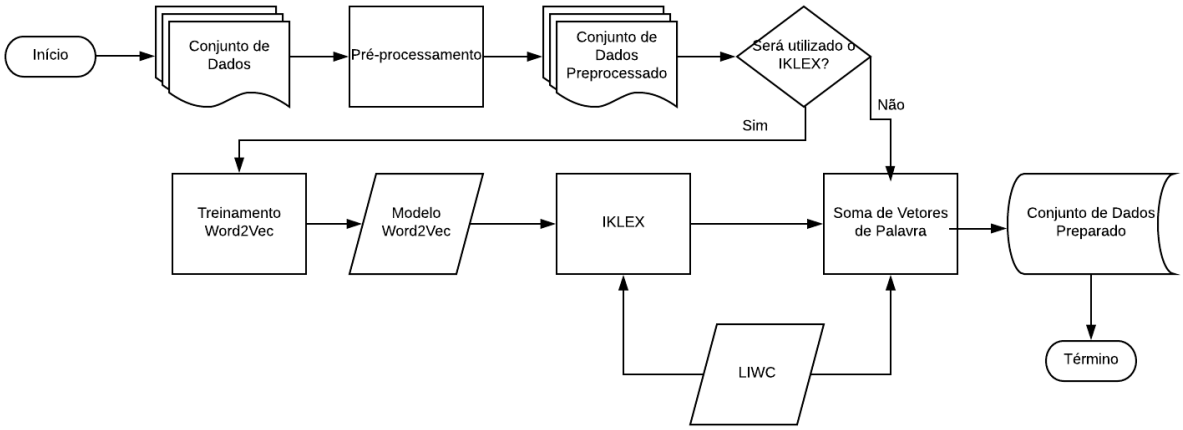

Figura IV.1: Fluxo de preparação dos dados para os classificadores

optando pela execução do IKLex, irá gerar os conjuntos de dados MQD-IKLEX e POLARITY-IKLEX, respectivamente. Ao executar o fluxo utilizando os conjuntos de dados MQDFT2018 e POLARITY-EN, sem optar pela execução do IKLex, irá gerar os conjuntos de dados MQD-LIWC e POLARITY-LIWC, respectivamente.

IV.5 Treinamento dos classificadores

Após preparar os conjuntos de dados pra classificação é necessário definir alguns aspectos relacionados a forma de treinamento e hiperparâmetros dos classificadores MNB, RF e LSVC. Para treinar os classificadores foi escolhida a biblioteca scikit-learn[2] da linguagem Python. Todos os classificadores foram treinados utilizando o método *10-fold cross validation*, conforme feito por Mullen and Collier [2004] e Wilson et al. [2005] e outros trabalhos na literatura.

O classificador MNB foi inicializado com o *Laplace smoothing* para evitar probabilidades iguais a zero [Saif et al., 2012]. O RF foi treinado com 128 árvores, levando em consideração o trabalho de Oshiro et al. [2012], onde os autores sugerem um número de árvores em uma RF entre 64 e 128. O tipo de SVM estabelecido para esse trabalho é o de *kernel* linear (LSVC), conforme descrito na seção II.5.1, devido a sua simplicidade e menor tempo de treinamento. O classificador LSVC foi configurado para resolver um problema de otimização primal, pois o número de features é menor que o conjunto de treinamento. Demais hiperparâmetros do LSVC foram definidos como os padrões para o algoritmo nesta biblioteca.

[2]http://scikit-learn.org/stable/

IV.6 Avaliação Experimental e Discussão

A tabela IV.1 mostra os *F1 scores* obtidos com os classificadores MNB, LSVC e RF sob os conjuntos de dados MQD-IKLEX, MQD-LIWC, POLARITY-LIWC e POLARITY-IKLEX. É possível observar uma melhora de ao menos 1% em cada classificador ao aplicar o algoritmo IKLEX, tanto no conjunto de dados em Português do Brasil, quanto no conjunto de dados em Inglês. O classificador que melhor se comportou para o idioma Inglês foi o LSVC, enquanto que para o idioma Português do Brasil foi o classificador RF,

Tabela IV.1: F1 score alcançado com a execução do MNB, LSVC e RF no conjunto de dados MQDFT2018

	MNB	LSVC	RF
MQD-LIWC	0.68	0.65	0.70
MQD-IKLEX	0.70	0.66	**0.72**
POLARITY-LIWC	0.65	0.72	0.68
POLARITY-IKLEN	0.66	**0.74**	0.69

Capítulo V Considerações

O presente estudo teve foco em tratar *OOV words* em léxicos para AS semelhantes ao LIWC. Foi proposto um algoritmo denominado IKLEX como solução para este problema. Também foi apresentado um novo conjunto de dados no idioma Português do Brasil proveniente de uma rede social brasileira chamada Meu Querido Diário, criando mais um recurso para a pesquisa de AS neste idioma.

A eficincia do algoritmo foi indicada pelos experimentos que foram conduzidos com o uso três classificadores em dois conjuntos de dados. Um dos conjuntos de dados está no idioma Português do Brasil e outro conjunto de dados no idioma Inglês. Os classificadores que foram treinados, o MNB, RF e LSVC mostraram uma melhora no *F1 score* de ao menos 1%. Em trabalhos futuros é preciso analisar o efeito do hiperparâmetro k no algoritmo IKLEX e sua relação com a quantidade de palavras que foram consideradas pelo algoritmo. Em trabalhos futuros, serão realizados experimentos com outros modelos de aprendizado de *word embeddings*, em outros léxicos e aplicá-los em outras tarefas.

Capítulo VI Cronograma

Segue abaixo as atividades pendentes até a defesa da dissertação.

1. Implementação de sugestões da banca de qualificação;
2. Trabalho de refinamento do texto;
3. Pesquisa sobre outras técnicas de avaliação de Tradução Automática de Textos (TAT)s

Tabela VI.1: Cronograma previsto até a defesa da dissertação

Atividades	2017	2018					
	12	01	02	03	04	05	06
1	X	X	X				
2	X	X	X	X	X		
3		X	X	X	X	X	
4		X	X	X			
5			X	X	X		
6			X	X	X		
7		X	X	X	X		
8			X	X	X	X	
9		X	X	X	X	X	
10					X	X	
11							X

Referências Bibliográficas

Balage Filho, P. P., Pardo, T. A. S., and Aluísio, S. M. (2013). An evaluation of the brazilian portuguese liwc dictionary for sentiment analysis. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364.

Bick, E. (2000). *The Parsing System"Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus Universitetsforlag.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bradley, M. M. and Lang, P. J. (1999). Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Citeseer.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Calvo, R. A. and D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on affective computing*, 1(1):18–37.

Calvo, R. A. and Mac Kim, S. (2013). Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543.

Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., and Ghosh, R. (2013). Discovering coherent topics using general knowledge. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 209–218. ACM.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Esuli, A. and Sebastiani, F. (2007). Sentiwordnet: a high-coverage lexical resource for opinion mining. *Evaluation*, pages 1–26.

Forman, G. (2007). Feature selection for text classification. *Computational methods of feature selection*, 1944355797.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., and Reyes, A. (2015). Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Hartmann, N. S., Avanço, L. V., Balage Filho, P. P., Duran, M. S., Nunes, M. D. G. V., Pardo, T. A. S., Aluisio, S. M., et al. (2014). A large corpus of product reviews in portuguese: Tackling out-of-vocabulary words. In *International Conference on Language Resources and Evaluation, 9th.* European Language Resources Association-ELRA.

Huang, M., Ye, B., Wang, Y., Chen, H., Cheng, J., and Zhu, X. (2014). New word detection for sentiment analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 531–541.

Ishizuka, M., Neviarouskaya, A., and Shaikh, M. A. M. (2012). Textual affect sensing and affective communication. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 6(4):81–102.

Keshavarz, H. and Abadeh, M. S. (2017). Alga: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. *Knowledge-Based Systems*, 122:1–16.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Kouloumpis, E., Wilson, T., and Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Maity, S., Chaudhary, A., Kumar, S., Mukherjee, A., Sarda, C., Patil, A., and Mondal, A. (2016). Wassup? lol: Characterizing out-of-vocabulary words in twitter. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*, pages 341–344. ACM.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

Mihaylov, T. and Nakov, P. (2016). Semanticz at semeval-2016 task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 879–886.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Muniz, M. C., Nunes, M. D. G. V., and Laporte, E. (2005). Unitex-pb, a set of flexible language resources for brazilian portuguese. In *Workshop on Technology on Information and Human Language (TIL)*, pages 2059–2068.

Nguyen, T. H., Shirai, K., and Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24):9603–9611.

Ombabi, A. H., Lazzez, O., Ouarda, W., and Alimi, A. M. (2017). Deep learning framework based on word2vec and cnnfor users interests classification. In *Computer Science and Information Technology (SCCSIT), 2017 Sudan Conference on*, pages 1–7. IEEE.

Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168. Springer.

Ouyang, X., Zhou, P., Li, C. H., and Liu, L. (2015). Sentiment analysis using convolutional neural network. In *Computer and Information Technology; Ubiquitous Computing and Communica-*

tions; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, pages 2359–2364. IEEE.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*.

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Park, S., Fazly, A., Lee, A., Seibel, B., Zi, W., and Cook, P. (2016). Classifying out-of-vocabulary terms in a domain-specific social media corpus. In *LREC*.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Rezapour, R., Wang, L., Abdar, O., and Diesner, J. (2017). Identifying the overlap between election result and candidates' ranking based on hashtag-enhanced, lexicon-based sentiment analysis. In *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*, pages 93–96. IEEE.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., and Stoyanov, V. (2015). Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 451–463.

Saif, H., He, Y., and Alani, H. (2012). Alleviating data sparsity for twitter sentiment analysis. CEUR Workshop Proceedings (CEUR-WS. org).

Singh, P., Dave, A., and Dar, K. (2017). Demonetization: Sentiment and retweet analysis. In *Inventive Computing and Informatics (ICICI), International Conference on*, pages 891–896. IEEE.

Tan, L., Zampieri, M., Ljubešic, N., and Tiedemann, J. (2014). Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 11–15.

Tavares, R. T. and Guedes, G. P. G. (2017). Classificação de filmes: uma abordagem utilizando o liwc. In *Congresso da Sociedade Brasileira de Computação-CSBC*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.