CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

TRATAMENTO DE PALAVRAS FORA DO VOCABULÁRIO EM TAREFAS DE ANÁLISE DE SENTIMENTOS COM LÉXICOS

Gabriel Nascimento dos Santos

Dissertação apresentada 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.

Orientador(a): Gustavo Paiva Guedes e Silva

Rio de Janeiro,

Julho 2019

TRATAMENTO DE PALAVRAS FORA DO VOCABULÁRIO EM TAREFAS DE ANÁLISE DE SENTIMENTOS COM LÉXICOS

Dissertação apresentada 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.

Gabriel Nascimento dos Santos

Banca Examinadora:

Presidente, Professor D.Sc. Gustavo Paiva Guedes e Silva (CEFET/RJ) (Orientador(a))

Professor D.Sc. Fellipe Ribeiro Duarte (UFRRJ)

Professor Ph.D Eduardo Bezerra da Silva (CEFET/RJ)

Professor D.Sc. Ronaldo Ribeiro Goldschmidt (IME)

Rio de Janeiro,

Julho 2019

CEFET/RJ – Sistema de Bibliotecas / Biblioteca Central

S237 Santos, Gabriel Nascimento dos
Tratamento de palavras fora de vocabulário em tarefas de análise de sentimentos com léxicos / Gabriel Nascimento dos Santos.—2019.
97f. : il. (algumas color.) , grafs. , tabs. ; enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca , 2019.
Bibliografia : f. 88-97
Orientador : Gustavo Paiva Guedes e Silva

1. Ciência da computação. 2. Mineração de uso da Web. 3. Processamento de linguagem natural. I. Guedes e Silva, Gustavo Paiva (Orient.). II. Título.

CDD 004

Elaborada pela bibliotecária Mariana Oliveira CRB-7/5929

DEDICATÓRIA

Dedico esta dissertação à minha família.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Agradeço ao apoio do meu orientador Gustavo Guedes por possibilitar este trabalho em tão pouco tempo e por estar disponível em diversos finais de semana e feriados.

Agradeço ao Fellipe Duarte pelo apoio, pelas duras críticas e por tirar minhas dúvidas e diversas reuniões que realizamos.

Agradeço ao meu amigo Jorge Soares por ter me iniciado na pesquisa no início do mestrado.

Agradeço à minha esposa Renée pelo apoio e paciência devido a minha menor atenção no período de pesquisa.

Agradeço à minha mãe Kathia por ter me criado com esforço e orientado a estudar em toda minha vida.

Agradeço à minha segunda mãe e avó Adelmira por ter me mimado.

Agradeço ao meu amigo Gustavo Castro que como amigo e diretor da DINFO/UERJ me apoiou nessa empreitada.

Agradeço ao meu amigo Alexandre Jammel pelos diversos conselhos.

Agradeço a Rafaella Galdino por ter me ajudado a revisar essa dissertação.

Agradeço aos meus amigos Alexandre Cunha, Flávio Damasceno, Rafael Guimarães, Carlos Vianna, Carlos Teles, Fernanda Britto e Rodolpho Nascimento pela amizade, inúmeras brincadeiras e discussões de cunho filosófico de botequim.

Agradeço aos diversos amigos que fiz no CEFET durante o mestrado.

É impossível enumerar todos que participaram e apoiaram indiretamente esta dissertação.

RESUMO

TRATAMENTO DE PALAVRAS FORA DO VOCABULÁRIO EM TAREFAS DE ANÁLISE DE SENTIMENTOS COM LÉXICOS

O número de usuários da internet que utilizam redes sociais, *microblogs* e sites de avaliação vem aumentando significantemente nos últimos anos. Com isso, usuários tendem a expor suas opiniões e transmitir o que sentem sobre determinado serviço, produto, e os mais diversos assuntos. Isso tem despertado o interesse de pesquisadores de processamento de linguagem natural, especialmente os de Análise de Sentimentos, que se interessam em explorar técnicas de extrair e entender as opiniões fornecidas pelos usuários que utilizam serviços orientados a opiniões. A Análise de Sentimentos possui três abordagens: a abordagem baseada em aprendizado de máquina, a abordagem baseada em léxicos e a abordagem híbrida. A abordagem baseada em léxicos e a abordagem híbrida sofrem com o problema de palavras fora do vocabulário ao lidar com a natureza dos textos de redes sociais. Lidar com textos provenientes de redes sociais é um grande desafio, pois eles variam de textos bem escritos a sentenças completamente sem sentido. Isso ocorre por diversos motivos, como a limitação do número de caracteres (como no *Twitter*) e até mesmo por erros ortográficos intencionais. Este trabalho propõe um algoritmo que utiliza *word embeddings* para tratar palavras fora do vocabulário em tarefas de Análise de Sentimentos com abordagens baseadas em léxico ou abordagens híbridas. A estratégia do algoritmo proposto é baseada na hipótese que palavras que tenham contextos parecidos, possuem significados semelhantes. O algoritmo consiste em eleger as palavras mais similares semanticamente e utilizar as categorias da mais próxima que esteja contida no léxico utilizado. Os experimentos foram conduzidos em três conjuntos de dados em Português do Brasil. Foram utilizados três classificadores e foram observadas melhorias de até $3,3$ pontos percentuais no *F1 score* após o uso do algoritmo proposto.

Palavras-chave: Palavras fora do vocabulário;Word embeddings; Análise de Sentimentos

ABSTRACT

HANDLING OUT-OF-VOCABULARY WORDS IN LEXICON-BASED SENTIMENT ANALYSIS TASKS

The number of Internet users who use social networks, microblogs and review sites has been increasing significantly in recent years. Therefore, users tend to express their opinions and convey what they feel about a given service, product, and the most diverse issues. This has attracted the interest of natural language processing researchers, especially those of Sentiment Analysis, who are interested in exploring techniques to extract and understand the opinions provided by users who use opinions-oriented services. The Sentiment Analysis has three approaches: machine-learning based approach, lexical-based approach and hybrid approach. The lexical and hybrid approaches suffers from the problem of out-of-vocabulary words in dealing with the nature of social network texts. Dealing with texts from social networks is a big challenge because they vary from well written texts to completely meaningless sentences. This occurs for a number of reasons, such as limiting the number of characters (such as Twitter) and even intentional misspellings. This work proposes a algorithm that uses word embeddings to treat words out of vocabulary in Analysis taskswith approaches based on lexical or hybrid approaches. The strategy of the proposed algorithm is based on the hypothesis that words that occurs in similar context tend to have similar meanings. The algorithm consists of choosing the most semantically similar words and using the features of the closest one that is contained in the lexicon used. The experiments were conducted in three datasets in Brazilian Portuguese.Three classifiers were used and improvements of up to 3.3 percent points in the F1 score were observed after the use of the proposed algorithm.

Keywords: Out-of-vocabulary words; Word embeddings; Sentiment Analysis

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

AS	Análise De Sentimentos
LSVC	*Linear Support Vector Classifier*
MNB	Multinomial Naive Bayes
MO	Mineração De Opiniões
MQD	Meu Querido Diário
MT	Mineração De Textos
OOV	Out-of-vocabulary Words
PLN	Processamento De Linguagem Natural
POS	Part-Of-Speech
RF	Random Forests
SMS	Short Message Service
SVM	Support Vector Machine

SUMÁRIO

Introdução

O uso e expansão da internet 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]. Essa expansão se deve ao aumento da popularidade de serviços orientados a opiniões de usuários, que surgiram nos últimos anos, como redes sociais, sites de críticas (reviews), *microblogs* [Pang et al., 2008; Rosenthal et al., 2015]. Isso tem despertado o interesse de empresas e pesquisadores de Análise de Sentimentos (AS) interessados em extrair opiniões e sentimentos desses serviços [Pang et al., 2008].

A Análise de Sentimentos (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 lida com três níveis contextuais: nível de documento, nível de sentença e nível de aspecto [Medhat et al., 2014].

No nível de documento e de sentenças, as tarefas de AS se concentram em analisar se um documento ou sentença possui opiniões positivas ou negativas [Medhat et al., 2014]. A única diferença entre o nível de documento e de sentenças é que as sentenças são apenas documentos curtos Medhat et al. [2014]. No geral, não há outras diferenças além desta.

O nível de aspectos possui o objetivo de classificar o sentimento em relação aos aspectos específicos das entidades envolvidas nos textos [Medhat et al., 2014]. O primeiro passo é identificar as entidades e seus aspectos [Medhat et al., 2014]. Os detentores de opinião podem dar opções diferentes para diferentes aspectos da mesma entidade [Medhat et al., 2014].

Na AS, o processo de detectar opiniões positivas e negativas também é chamado de classificação de polaridade [Noferesti and Shamsfard, 2015]. A classificação de polaridade é a principal tarefa da AS [Noferesti and Shamsfard, 2015]. Ela se concentra em detectar se um documento, sentença ou aspecto possui sentimento positivo ou negativo, conforme realizado nos trabalhos de Farra et al. [2010]; Tan et al. [2011]; Wagner et al. [2014].

A AS também possui três abordagens: a AS baseada em aprendizado de máquina, a abordagem baseada em léxicos e a abordagem híbrida [Medhat et al., 2014]. A abordagem baseada em aprendizado de máquina pode ser dividida nas abordagens supervisionada e não-supervisionada [Medhat et al., 2014]. 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 [Medhat et al., 2014].

A abordagem baseada em léxicos se divide em mais duas abordagens: a baseada em dicionários e a baseada em corpus [Ravi and Ravi, 2015; Medhat et al., 2014]. A abordagem baseada em dicionários depende de um dicionário para fornecer a polaridade, ou orientação semântica, para cada palavra [Balage Filho et al., 2013; Ravi and Ravi, 2015; Medhat et al., 2014]. Essa abordagem não requer um conjunto de dados anotado, no entanto, o dicionário precisa ter um bom conjunto de palavras para ter uma boa cobertura de vocabulário [Balage Filho et al., 2013].

A abordagem baseada em corpus se baseia em padrões sintáticos e na probabilidade de ocorrência de uma palavra em 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]. A abordagem baseada em corpus lida bem com palavras de um domínio ou contexto específico, pois seus métodos dependem de padrões sintáticos que ocorrem no corpus [Medhat et al., 2014], enquanto a abordagem que se baseia em dicionários possui dificuldades de lidar com essas palavras [Medhat et al., 2014; Park et al., 2016].

Por fim, a abordagem híbrida se trata do uso conjunto das abordagens baseadas em aprendizado de máquina e abordagem baseada em léxicos [Medhat et al., 2014]. É muito comum que em abordagens híbridas os léxicos adotados possuam um papel importante [Medhat et al., 2014]. Por fazer uso de léxicos, o problema de palavras fora do vocabulário também pode afetar as abordagens híbridas.

É conhecido que a qualidade dos textos encontrados em conteúdos gerados por usuários em redes sociais podem ser textos bem escritos, como conteúdos de notícias, ou podem ser textos sem sentido algum [Agichtein et al., 2008; Han et al., 2013]. Em geral, redes sociais possuem muitos erros de ortografia, gírias e gramática incomum, gerando palavras fora do vocabulário, também chamadas de *Out-of-vocabulary words (OOV)*, em inglês [Nguyen et al., 2015; Agichtein et al., 2008]. Com isso, as palavras fora

do vocabulário dificultam especialmente as tarefas de AS que usam léxicos.

Embora métodos baseados em léxicos geralmente tenham desempenho inferior em relação aos métodos baseados em aprendizado de máquina, eles continuam sendo competitivos, pois os métodos baseados em aprendizado de máquina supervisionados necessitam de amostras manualmente anotadas que nem sempre estão disponíveis [Hailong et al., 2014]. Isso justifica o estudo de formas de melhorar o desempenho de métodos baseados em léxicos. Uma das formas de melhorar o desempenho é de alguma maneira expandir a quantidade de palavras abrangidas por estes léxicos, através da expansão de um léxico, como proposto por Huang et al. [2014] e Rezapour et al. [2017]. Também é possível tratar esse problema treinando e utilizando algum classificador que categorize as palavras fora do vocabulário, conforme proposto por Maity et al. [2016]. Outra forma de tratar o problema é normalizar os textos, tentando retornar as palavras a sua forma canônica, conforme proposto por Han et al. [2013] e Hartmann et al. [2014].

No entanto, a normalização realizada em mensagens de texto (e redes sociais) enfrenta o problema no qual as variantes lexicais presentes nos textos são frequentemente geradas intencionalmente, seja devido ao desejo de poupar caracteres, por identidade social ou devido a convenções neste subgênero de texto [Han et al., 2013]. A solução proposta neste trabalho não enfrenta este problema, pois se trata de uma forma de expansão de um léxico já existente, tratando palavras fora do vocabulário como outras palavras semanticamente mais próximas.

O objetivo desta dissertação concentra-se em um esforço para tratar palavras fora do vocabulário ao utilizar métodos de AS baseados em léxicos e métodos híbridos, no nível de documentos e sentenças. Em contraste com abordagens que tratam palavras fora do vocabulário com recursos externos, este trabalho visa resolver esse problema considerando a palavra ausente como outra palavra presente no léxico. Desta forma, foi proposto um algoritmo chamado *IKLex 2* (*Input KNN Lexicon 2*), uma variante do algoritmo *IKLex* [Nascimento et al., 2018b], por sua vez baseado no algoritmo k-Nearest Neighbor (KNN). O IKLEX 2 funciona de maneira que, dada a ausência de uma palavra em um léxico, as categorias da palavra mais próxima semanticamente são utilizadas.

O *IKLex 2* se baseia na hipótese distribucional [Harris, 1954], cuja a ideia é que palavras que possuem o mesmo contexto, tendem a ter o mesmo significado. Porquanto, é necessário utilizar uma forma de Processamento de Linguagem Natural (PLN) que seja compatível com a hipótese distribucional e que possua características semânticas. Com

base na hipótese distribucional, muitos métodos de representações de palavras foram explorados na comunidade de PLN [Levy and Goldberg, 2014]. No estado da arte da área de PLN as palavras são representadas como vetores densos que são aprendidos por meio do treinamento de redes neurais [Levy and Goldberg, 2014]. Esses vetores são denominados *word embeddings* [Levy and Goldberg, 2014].

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 característica 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., 2013a], GloVe [Pennington et al., 2014] e o FastText [Bojanowski et al., 2016]. Por *word embeddings* conterem características semânticas, o *IKLex 2* os utiliza para obter a similaridade entre palavras.

Esta dissertação apresenta como contribuição: (i) o algoritmo *IKLex 2*. (ii) um novo conjunto de dados, denominado $MQDEmotion2018$, colhido de uma rede social brasileira chamada Meu Querido Diário (MQD). Nessa rede social, os usuários alimentam um diário como uma atividade cotidiana e interagem fazendo comentários. Esse conjunto de dados possui como diferencial sua precisão em relação ao rótulo das emoções relacionados aos documentos, pois nessa rede social os próprios usuários podem informar a emoção relacionada a sua publicação no diário.

As contribuições encontram-se desmembradas nas seguintes publicações realizadas durante o estudo deste trabalho:

- NASCIMENTO, G; DUARTE, F.; GUEDES, G. P.; Emoções em português do Brasil: um conjunto de dados e resultados de base. VII Brazilian Workshop on Social Network Analysis and Mining (BraSNAM), 2018, Natal. Anais do XXXVIII Congresso da SBC, 2018.

- NASCIMENTO, G; DUARTE, F.; GUEDES, G. P.; Handling out-of-vocabulary words in lexicons to polarity classification. IHC, 2018, Belém. Anais do XVII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, 2018.

Os experimentos foram conduzidos utilizando três conjuntos de dados em Português do Brasil, nos quais um dos conjuntos de dados é proveniente do MQD e dois conjuntos de dados provenientes do *Twitter*. Também foram utilizados o LIWC como léxico

e o *Word2Vec* para treinar os *word embeddings*.

Os experimentos possuem o objetivo de responder se o uso do *IKLex 2* para tratar palavras fora do vocabulário melhora a classificação de polaridade. A forma de avaliar os experimentos neste trabalho consiste em treinar diversos classificadores e avaliar as métricas de cada modelo. Dessa forma, foram treinados classificadores que utilizam o LIWC e classificadores que utilizaram léxicos expandidos com o *IKLex 2*. Posteriormente, esses modelos foram comparados e e foi realizado um teste de hipótese com o melhor resultado obtido.

Foram escolhidos três classificadores para a condução dos experimentos, sendo eles: o *Multinomial Naive Bayes (MNB)*, *Linear Support Vector Classifier (LSVC)* e *Random Forests (RF)*. Esta escolha foi baseada no fato desses algoritmos serem amplamente utilizados em tarefas de AS [Feldman, 2013; Gupte et al., 2014; Ali et al., 2012]. Os resultados obtidos com os testes dos três classificadores indicam que o usar o IKLex é melhor que desconsiderar as palavras fora do vocabulário.

Os resultados indicam que houve uma melhora no desempenho da classificação de polaridade após o tratamento das palavras fora do vocabulário com o *IKLex 2*. Isso indica que tratar as palavras fora do vocabulário é importante para as tarefas AS baseadas em léxicos ou híbridas. Também indica que os léxicos gerados pelo *IKLex 2* são superiores ao léxico original.

Esta dissertação está organizada em 6 capítulos. O capítulo 1 apresenta todo o referencial teórico necessário para o entendimento do assunto abordado nos próximos capítulos. O capítulo 2 descreve os trabalhos relacionados a esta dissertação. O capítulo 3 discorre sobre a proposta desta dissertação. O capítulo 4 apresenta a avaliação dos experimentos. Por fim, o trabalho segue com as considerações finais.

1- 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 1.1 disserta sobre a Análise de Sentimentos e sua tipologia. A seção 1.2 apresenta a hipótese distribucional e seu uso na área de Processamento de Linguagem Natural. A seção 1.3 apresenta o LIWC, que possui um léxico utilizado nesta dissertação. A seção 1.4 apresenta o *Word2Vec*. A seção 1.5 explica sobre a similaridade por cosseno. A seção 1.6 apresenta os três classificadores utilizados nesta dissertação: *Support Vector Machines*, *Multinomial Naive Bayes* e *Random Forest*. A seção 1.7 apresenta a métrica de avaliação de classificação *F1 score*. Por fim, a seção 1.8 explica a técnica de validação cruzada.

1.1- Análise de Sentimentos

A Análise de Sentimentos (AS), também chamada de Mineração de Opiniões (MO) é um campo de pesquisa na área de Mineração de Textos (MT) que estuda opiniões, sentimentos e subjetividades em textos [Medhat et al., 2014]. Os sentimentos e opiniões em textos podem se referir, por exemplo, a indivíduos, eventos, produtos, ou outros assuntos. Alguns pesquisadores, no entanto, apontam que AS e MO possuem leves diferenças em seus conceitos [Medhat et al., 2014]. Eles apontam que a MO é originária da área de pesquisa de Recuperação da Informação, e que a MO extrai e analisa as opiniões presentes em um texto a respeito de uma entidade (indivíduos, eventos ou tópicos), enquanto a AS é originária da área de pesquisa de Processamento de Linguagem Natural e possui o objetivo de obter os sentimentos expressos em textos [Tsytsarau and Palpanas, 2012]. No entanto, esses dois problemas são semelhantes em sua essência [Tsytsarau and Palpanas, 2012].

A AS pode ser considerada um processo de classificação de polaridade [Medhat et al., 2014]. A classificação de polaridade possui o objetivo de detectar se uma opinião

ou sentimento são positivos ou negativos [Noferesti and Shamsfard, 2015]. Esse processo de classificação pode ocorrer em diferentes níveis de granularidade. A AS atua em três níveis de granularidade, sendo eles: o nível de documento, nível de sentenças e nível de aspectos [Medhat et al., 2014].

O nível de documento possui o objetivo de classificar se um documento expressa um sentimento ou opinião positiva ou negativa, tratando cada documento como uma unidade básica e atômica de informação [Medhat et al., 2014]. O nível de sentenças determina se cada sentença expressa um sentimento ou opinião positiva ou negativa [Medhat et al., 2014]. No entanto, não há diferença fundamental entre nível de documento e sentença porque sentenças são apenas documentos curtos [Liu, 2012]. Porquanto, o nível de aspectos possui o objetivo de identificar sentimentos a respeito de aspectos de entidades [Medhat et al., 2014], ou seja, uma entidade pode possuir um ou mais aspectos ou tópicos, e esses aspectos podem ter sentimentos positivos e negativos. Como exemplo, podemos considerar a seguinte sentença a respeito de um produto: “O **smartphone** possui um bom **desempenho**, mas a bateria não **dura**”. O *smartphone* é a entidade, que possui opiniões diferentes a respeito de diferentes aspectos, no caso em questão desempenho e a duração da bateria.

Segundo Medhat et al. [2014], a AS possui três abordagens: a abordagem com aprendizado de máquina, a abordagem baseada em léxicos e a abordagem híbrida. A abordagem com aprendizado de máquina utiliza algoritmos de aprendizado de máquina para deduzir o sentimento dos textos utilizando suas características sintáticas e linguísticas [Medhat et al., 2014]. Essa abordagem faz uso de algoritmos de aprendizagem 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 precisa de documentos rotulados [Medhat et al., 2014].

A segunda abordagem é a abordagem baseada em léxicos, que se divide em duas: a baseadas em dicionários e a baseada em corpus [Ravi and Ravi, 2015; Medhat et al., 2014]. A baseada em dicionários faz uso de um conjunto bem definido de palavras manualmente anotadas, com orientações bem definidas [Medhat et al., 2014]. A abordagem baseada em corpus faz uso do próprio *corpus* de texto para encontrar palavras que expressam opinião com orientações específicas de contexto [Medhat et al., 2014]. Essa abordagem se baseia em padrões sintáticos e na probabilidade de ocorrência de

uma palavra em 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].

Segundo Medhat et al. [2014], existem as abordagens híbridas, que utilizam aprendizado de máquina e léxicos. Em abordagens híbridas os léxicos possuem um papel chave, em conjunto com os algoritmos de aprendizado de máquina [Medhat et al., 2014]. Dessa forma, é normal que léxicos sejam usados em conjunto com um classificador.

A figura 1 ilustra as subdivisões das abordagens de AS. A proposta dessa dissertação se concentra nas abordagens de AS que utilizam léxicos baseadas em dicionários e híbridas. As abordagens utilizadas neste trabalho estão destacadas em azul no diagrama de *Venn* da figura 1.

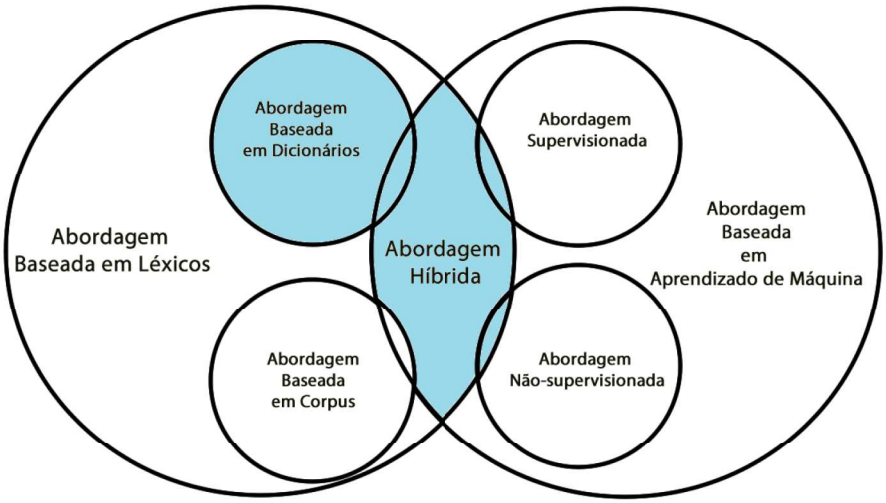

Figura 1 – Abordagens de Análise de Sentimentos. As abordagens destacadas em azul são as abordagens que essa dissertação se concentra.

1.2- Hipótese Distribucional no Processamento de Linguagem Natural

Wittgenstein [2009] destaca que o significado de uma palavra é o seu uso na linguagem. O uso de uma palavra em um idioma ou linguagem também envolve qual contexto a palavra está inserida. No entanto, existem diversas definições sobre o que o que seria esse “contexto” [Erk, 2012]. Na definição mais simples, um contexto seria

apenas um conjunto de palavras (*bag of words*) que ocorreram nas proximidades da palavra alvo [Erk, 2012].

Nesse cenário, Harris [1954] apresentou a hipótese distribucional, que se baseia no princípio de que partes de uma linguagem não ocorrem arbitrariamente em relação uma à outra: cada elemento ocorre em certas posições relativas a certos outros elementos. A ideia geral por trás da hipótese distribucional parece bastante clara: existe uma correlação entre semelhança distribucional e similaridade do significado de uma palavra, o que nos permite utilizar a primeira para estimar a segunda [Sahlgren, 2008].

Embora façam muitos anos que o estudo da semântica tenha surgido no campo da línguística, até certo tempo a área de Processamento de Linguagem Natural (PLN) e Recuperação da Informação (RI) ainda tratavam palavras apenas como unidades quase atômicas. Existem diversas formas de representar um documento de texto [Huang]. É muito comum que os documentos sejam representados como *bag of words* [Harris, 1954].

Na representação *bag of words* cada documento é considerado como um "saco" de palavras que são independentes e sem ordem [Huang]. Cada documento torna-se um vetor que consiste em valores não negativos em cada dimensão [Huang]. Cada dimensão ou palavra do documento é pontuada de acordo com sua frequência, o que significa que os termos que aparecem com mais frequência são mais importantes e descritivos para o documento [Huang].

É válido ressaltar que a principal desvantagem da representação em *bag of words* é a perda de informação semântica [Bekkerman and Allan, 2004]. Nesse sentido, o campo de pesquisa de PLN começou a buscar soluções que levam em conta a semântica e o contexto de cada elemento de linguagem. Um exemplo disso é o surgimento do *bag of n-grams* [Lewis, 1992], que considera combinações de até n palavras para obter alguma informação semântica dos documentos.

Havendo ainda a deficiência da representação *bag of words* e *bag of n-grams* em relação a semântica, pesquisadores propuseram modelos para representar palavras em um espaço vetorial que agrupa palavras semelhantes, alcançando melhor desempenho em tarefas de PLN [Mikolov et al., 2013b]. Esses modelos são conhecidos como *word embeddings* e possuem um grande poder de generalização [Camacho-Collados and Pilehvar, 2018]. No estado da arte da área de PLN é comum o uso de *word embeddings* gerados por redes neurais [Camacho-Collados and Pilehvar, 2018].

A figura 2 apresenta um exemplo de modelo fictício de *word embeddings* gerado

em um espaço vetorial bidimensional. Observe que no modelo fictício apresentado, o vetor que se refere à palavra “casa” é próximo de “apartamento” e “homem” é um vetor próximo de “mulher”. Dependendo do contexto apresentado, a semântica da palavra “casa” se aproxima da palavra “apartamento”. Se levarmos em conta as características do mundo real, as casas possuem janelas, portas e são de fato uma forma de moradia, assim como os apartamentos.

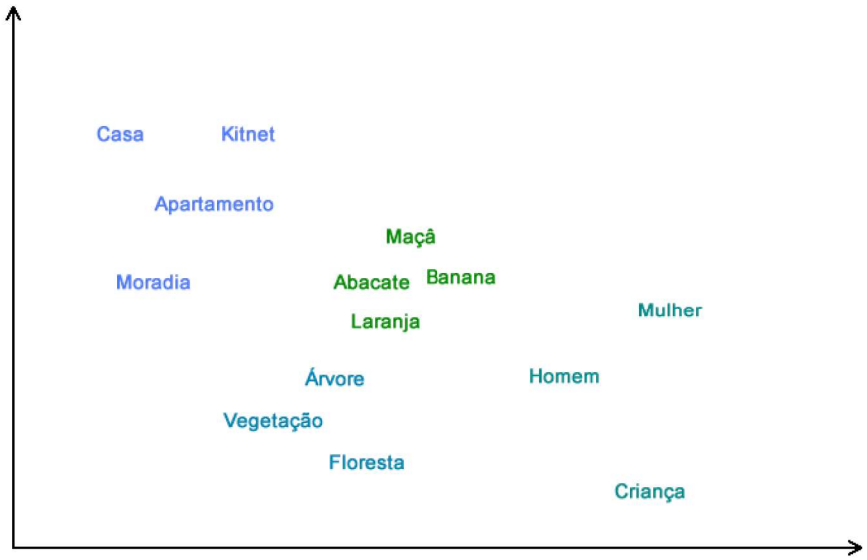

Figura 2 – Exemplo de word embeddings em um espaço vetorial bidimensional

Ao longo da pesquisa foi observado que alguns autores consideram importante levar em conta a semântica na Análise de Sentimentos [Saif et al., 2016, 2012], especialmente quando se trata de conjuntos de dados grandes e quais os documentos presentes não tratam apenas de um determinado assunto. Assim como tais autores esta dissertação também considera a semântica algo importante para a Análise de Sentimentos. Destarte, a proposta desta dissertação se baseia na hipótese distribucional, ou seja, que palavras que ocorrem em contextos similares tendem a ter significados similares [Malcolm, 1954; Turney and Pantel, 2010].

1.3- 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, o LIWC 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. Essas categorias podem se referir à classes gramaticais (e.g., pronome, verbo) ou à aspectos psicológicos e/ou cognitivos (e.g., raiva, tristeza). A versão do LIWC utilizada neste trabalho é o LIWC 2007. O léxico do LIWC em sua versão 2007 possui 127.149 palavras e *word stems* para português do Brasil [Balage Filho et al., 2013].

1.4- Word2Vec

Word2Vec são arquiteturas de redes neurais, propostas por Mikolov et al. [2013a] para realizar o aprendizado de *word embeddings*. Mikolov et al. [2013a] propõem duas arquiteturas para gerar *word embeddings*. A primeira arquitetura é denominada Continuous Bag-Of-Words (CBOW) e a segunda é denominada Skip-Gram. A diferença entre essas duas arquiteturas é em relação a tarefa de aprendizado proposta, mas ambas possuem apenas uma camada oculta.

A arquitetura CBOW aprende uma palavra por outras palavras dentro de um contexto de tamanho fixo. Dado um vocabulário de tamanho V, cada palavra é representada como um vetor *one-hot encoded* de tamanho V, ou seja, cada palavra é representada por um índice $w \in \{1, ..., V\}$ de um vetor $\vec{v}$ de tamanho V, qual $\vec{v}_w = 1$ e todos os outros índices $\vec{v}_{w'} = 0$. Seja C o tamanho do contexto de palavras, a camada de entrada corresponderá a cada um desses vetores *one-hot encoded* de cada palavra.

A camada oculta possui N neurônios, que correspondem ao número de dimensões

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

escolhidas para os *word embeddings*. Então, os pesos correspondentes à camada oculta da rede neural são representados por uma matriz W de dimensões $V \times N$. Em seguida, caso $C = 1$, a camada oculta realiza uma transformação linear, conforme apresentado na equação 1. Em cenários que $C > 1$, ou seja, há mais de uma palavra no contexto, a ativação é correspondente a equação 2, que se trata de uma média das ativações realizadas com os vetores das palavras da camada de entrada:

$$h = W^T x = W^T_{(k,.)} := v_{wI} \tag{1}$$

$$h = \frac{1}{C} \sum_{i}^{C} W^T x_i := v_{wI} \tag{2}$$

Note que na equação 1 o que ocorre é apenas uma cópia da k-ésima linha de W para h. O vetor v_{wI} trata-se da representação vetorial em N dimensões das palavras de entrada. Não existe uma função de ativação não-linear para a camada oculta. Por conseguinte, os pesos da camada de saída são representados em uma matriz W' com dimensões $N \times V$. Com isso, é calculado um escalar u_j para cada palavra no vocabulário na equação 3, qual v'_{wj} é referente a j-ésima coluna da matriz W'. Por fim, pode ser calculada a função *Softmax* para obter a distribuição de probabilidade posterior das palavras, conforme a equação 4.

$$u_j = {v'_{wj}}^T h \tag{3}$$

$$\hat{y} = \frac{e^{u_j}}{\sum_{m=1}^{V} e^{u_m}} \tag{4}$$

A equação 4 pode ser expandida para melhor entendimento, conforme a equação 5. Note que v'_{wj} e v_{wI} são duas representações vetoriais distintas para uma palavra w. A partir deste momento, o vetor v'_{wj} pode ser referido como representação vetorial de saída e v_{wI} como representação vetorial de entrada.

$$\hat{y} = \frac{e^{{v'_{wj}}^T v_{wI}}}{\sum_{m=1}^{V} e^{{v'_{wm}}^T v_{wI}}} \tag{5}$$

O objetivo corresponde a maximizar a probabilidade condicional apresentada na equação 6, na qual w_j representa a palavra que deve ser classificada, w_c representa o

conjunto de palavras do contexto apresentado como entrada. Sabendo que j representa o índice da palavra no vocabulário na camada de saída e no vocabulário, u_j é referente ao escalar resultante da transformação linear na camada de saída para a palavra w_i. A equação 7 pode ser interpretada como uma métrica de entropia cruzada entre duas distribuições de probabilidade.

$$\max \sum_{i=1}^{V} p(w_j|w_c) \tag{6}$$

$$p(w_j|w_c) = u_j - \log \sum_{m=1}^{V} e^{u_m} \tag{7}$$

No entanto, calcular $\sum_{m=1}^{V} e^{u_m}$ é computacionalmente ineficiente, pois é preciso iterar sobre todas as palavras no vocabulário. Por isso, Mikolov et al. [2013b] sugeriram outros métodos para aproximação do *softmax* chamados *Amostragem Negativa* e *Softmax Hierárquico* para tornar o custo computacional menor.

A arquitetura Skip-Gram possui um objetivo de aprendizado inverso ao da arquitetura CBOW, conforme mostrado na figura 3. O objetivo é prever as palavras que contextualizam outra palavra. Assim como na arquitetura CBOW, cada palavra de um vocabulário de tamanho V é representada como um vetor *one-hot encoded* de tamanho V, ou seja, cada palavra é representada por um índice $w \in \{1, ..., V\}$ de um vetor $\vec{v}$ de tamanho V, qual $\vec{v}_w = 1$ e todos os outros índices $\vec{v}_{w'} = 0$.

Nota-se que a camada oculta possui N neurônios e nela é realizada uma transformação linear assim como apresentada na equação 6 para a arquitetura CBOW, na qual x é o vetor *one-hot encoded* de uma palavra. O objetivo corresponde a maximizar a probabilidade condicional apresentada na equação 8, qual w_c se refere às palavras em torno da palavra w_i. No entanto, para calcular $p(w_c|w_i)$ também é preciso calcular $\sum_{m=1}^{V} e^{u_m}$ devido a função *Softmax*, o que é computacionalmente ineficiente, pois é preciso iterar sobre todas as palavras no vocabulário. Por isso, no Skip-Gram é utilizado o *Amostragem Negativa* ou o *Softmax Hierárquico*, como proposto por Mikolov et al. [2013b]. Nas próximas subseções há uma breve explicação sobre os dois métodos, e o motivo da escolha da *Amostragem Negativa* para este trabalho.

$$\max \sum_{j=1}^{V} p(w_c|w_j) \tag{8}$$

Mikolov et al. [2013a] explica que a arquitetura Skip-Gram gera melhores *word embeddings*, embora o custo computacional seja maior. Por isso, nessa dissertação foi escolhida a arquitetura Skip-Gram.

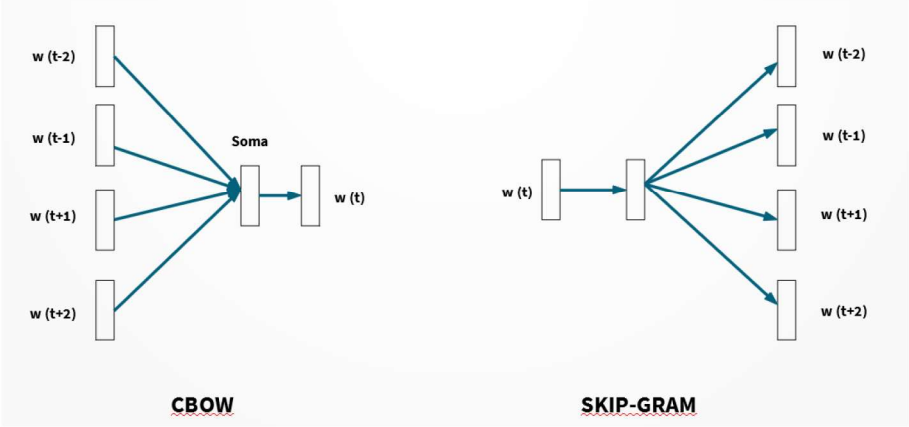

Figura 3 – Arquiteturas do Word2Vec. Retirado de [Mikolov et al., 2013a].

1.4.1 Softmax Hierárquico

O Softmax Hierárquico foi inicialmente proposto por Mnih and Hinton [2009] como uma técnica de aproximação do *Softmax*. Essa aproximação é necessária devido a complexidade computacional quando existem muitas classes. O Softmax Hierárquico implica na construção de uma árvore binária hierárquica indexando todos os termos de um vocabulário em nós para obter a distribuição de probabilidades com menor custo computacional. Havendo um vocabulário de tamanho V, no *Softmax* clássico são avaliadas todas as probabilidades dos V termos dos vocabulários, enquanto no *Softmax Hierárquico*, por se tratar de uma árvore binária, são avaliadas apenas $log_2(V)$ nós [Mikolov et al., 2013b].

Mikolov et al. [2013b] explica que uma palavra w pode ser alcançada de forma eficiente pelo nó da árvore binária. Sendo $n(w, j)$ o j-ésimo nó no caminho da raíz para a palavra w, $L(w)$ o tamanho total do caminho da raíz para w, logo $n(w, 1)$ corresponde a raíz da árvore binária e $n(w, L(w)) = w$. Para qualquer nó n, $ch(n)$ é um nó filho qualquer

de n. Dessa forma, o *Softmax Hierárquico* considera a seguinte probabilidade condicional abaixo.

$$P(w|w_i) = \prod_{j=1}^{L(w)-1} \sigma\left(\phi\Big(n(w,j+1) = ch\big(n(w,j)\big)\Big) \cdot {v'_{n(w,j)}}^T v_{wI}\right) \tag{9}$$

Sendo $\sigma(x) = \frac{1}{1+e^{\ x}}$ a função sigmóide, $v'_{n(w,j)}$ a representação vetorial do nó interno $n(w,j)$ e h é a saída da camada oculta do *Word2Vec*. Dessa forma, a função especial ϕ é definida abaixo:

$$\phi = \begin{cases} 1, \text{ se x é verdadeiro} \\ -1, \text{ se x é falso} \end{cases}$$

A figura 4 mostra um exemplo de árvore binária representando o Softmax Hierárquico. Lopes [2015] explica de forma intuitiva que: a probabilidade de uma palavra w_2 é a probabilidade de um caminho aleatório começando da raíz terminar nó correspondente a palavra w_2. Destarte, conforme exposto pela figura 4, os nós internos em cinza, representam a distribuição de probabilidade, enquanto os nós folha representam as palavras. Para cada nó interno percorrido, é preciso calcular a probabilidade de ir para o nó da direita ou esquerda. A probabilidade de ir para um nó filho a esquerda em um nó interno n é definido pela equação 10, enquanto a equação 11 apresenta a probabilidade de ir para o nó filho a direita. Em ambas equações, v'_n é a representação vetorial do nó n e v_{wI} é a representação vetorial de entrada:

$$p(n, left) = \sigma({v'_n}^T \cdot v_{wI}) \tag{10}$$

$$p(n, right) = 1 - \sigma({v'_n}^T \cdot v_{wI}) = \sigma(-{v'_n}^T \cdot v_{wI}) \tag{11}$$

Como o Softmax Hierárquico possui o formato de uma árvore binária, existem $V-1$ nós intermediários. Desse modo, segundo a figura 11 a probabilidade de uma palavra w_2 ser a saída w_O é calculada conforme a equação 12. Note que $\sum_{i=1}^{V} p(w_i = w_O) = 1$:

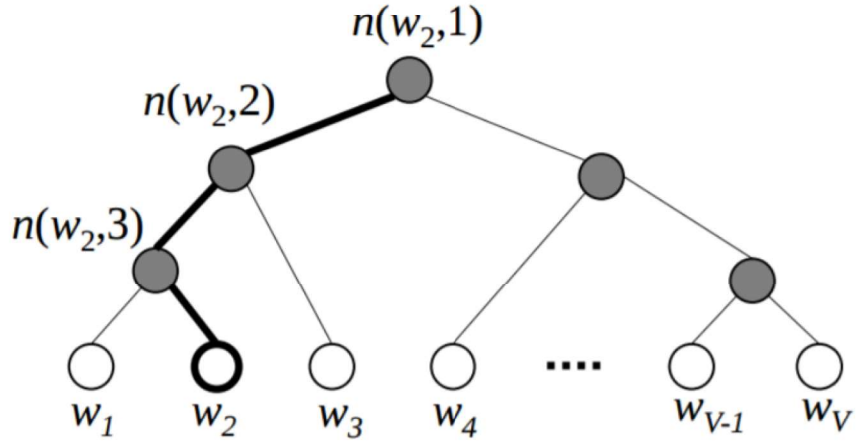

Figura 4 – Exemplo de árvore binária utilizando Softmax Hierárquico. Os nós em cinza representam a distribuição de probabilidade, enquanto os nós brancos são as palavras no vocabulário. Retirado de [Lopes, 2015].

$$\begin{aligned} p(w_i = w_O) &= p\big(n(w_2,1), left\big) \cdot p\big(n(w_2,2), left\big) \cdot p\big(n(w_2,3), right\big) \\ &= \sigma({v'_{w2,1}}^T \cdot v_{wI}) \cdot \sigma({v'_{w2,2}}^T \cdot v_{wI}) \cdot \sigma(-{v'_{w2,3}}^T \cdot v_{wI}) \end{aligned} \tag{12}$$

1.4.2 Amostragem Negativa

A Amostragem Negativa [Mikolov et al., 2013b] é uma forma simplificada da Estimativa Contrastiva de Ruído (NCE), que foi inicialmente proposta por Gutmann and Hyvärinen [2012]. A NCE postula que um bom modelo deve ser capaz de diferenciar dados de ruído por meio de regressão logística [Mikolov et al., 2013b]. A Amostragem Negativa é um método estocástico de aproximação da maximização da *log-verossimilhança* do Softmax [Mikolov et al., 2013b].

Empiricamente, segundo Mikolov et al. [2013b], a Amostragem Negativa supera o Softmax Hierárquico. A subamostragem das palavras frequentes melhora a velocidade de treinamento e gera *word embeddings* de melhor qualidade [Mikolov et al., 2013b]. Destarte, nesse trabalho foi escolhido a Amostragem Negativa ao invés do Softmax Hierárquico.

A Amostragem Negativa é explicada a seguir. Considere D o conjunto de todos os pares de palavras (w, c) que estão no texto e D' o conjunto de todos os pares (w, c) que não estão presentes no texto. Considere também que $D \cup D' = U$ e $D \cap D' = \emptyset$. Com isso, a função z pode ser definida da seguinte maneira:

$$z = \begin{cases} 1, \text{ se o par } (w, c) \in D \\ 0, \text{ se o par} (w, c) \in D' \end{cases}$$

Assim, podemos definir que $p(z = 1|(w, c))$ corresponde a probabilidade do par $(w, c) \in D$ e $p(z = 0|(w, c))$ corresponde a probabilidade do par $(w, c) \in D'$. A probabilidade $p(z = 1|(w, c))$ é definida na equação 13, como um problema de classificação de regressão logística.

$$p(z = 1|(w, c)) = \frac{1}{1 + e^{v_c^T v_w}} \tag{13}$$

O vetor v_c^T se refere a representação vetorial de saída da palavra c e v_w se refere a representação vetorial de entrada da palavra w. O complemento de $p(z = 1|(w, c))$ é $p(z = 0|(w, c))$. Portanto, a $p(z = 0|(w, c))$ é calculada conforme equação:

$$p(z = 0|(w, c)) = \frac{1}{1 + e^{-v_c^T v_w}} \tag{14}$$

Logo, a probabilidade a ser otimizada pelo classificador seria reduzido a equação 15. Observe que dependendo se $z = 0$ ou $z = 1$, apenas uma parcela o cálculo é levada em conta, devido ao expoente. Assim, a função de log-verossimilhança é definida pela equação 15.

$$L(\theta) = \prod_{(w,c) \in D \cup D'} \left(\frac{1}{1 + e^{v_c^T v_w}} \right)^z \left(\frac{1}{1 + e^{-v_c^T v_w}} \right)^{1-z} \tag{15}$$

$$l(\theta) = \sum_{(w,c) \in D} \log \frac{1}{1 + e^{v_c^T v_w}} + \sum_{(w,c) \in D'} \log \frac{1}{1 + e^{-v_c^T v_w}} \tag{16}$$

Observe que, em termos práticos, não é possível obter o conjunto D', uma vez que não é possível estimar todos os pares (w, c) possíveis. A amostragem negativa tratar este problema de forma estocástica, obtendo amostras negativas (também chamadas de

rúido) do conjunto D' de tamanho k. Considere K um conjunto de tamanho k e $K \in D'$. Portanto, sendo $\sigma(x) = \frac{1}{1+e^{-x}}$ a função sigmóide, a equação 17 é a equação 16 adaptada para o cenário estocástico da Amostragem Negativa.

$$l(\theta) = \sum_{(w,c)\in D} \log \sigma(v_c^T v_w) + \sum_{(w,c)\in K} \log \sigma(-v_c^T v_w) \tag{17}$$

1.5- Similaridade por Cosseno

A similaridade entre vetores do mesmo número de dimensões pode ser medida pelo cálculo do cosseno do ângulo formado entre entre eles. Considere os vetores A e B exibidos na figura 6. Esses vetores podem ser referentes, por exemplo, a dois *word embeddings*. Com isso, a similaridade entre esses vetores pode ser medida pelo do cosseno do ângulo θ.

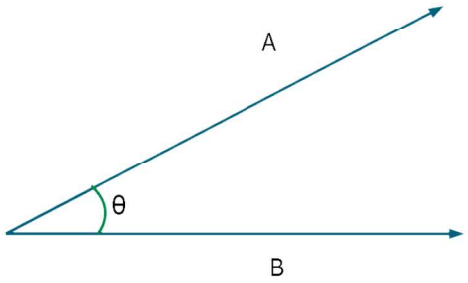

Figura 5 – Exemplo de dois vetores cuja distância ou similaridade pode ser medida pelo cosseno do ângulo θ. Os vetores podem ser referentes a dois *word embeddings*.

Conforme equação 18, 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 (MT) costumam utilizar a medida de similaridade por cosseno para calcular a similaridade entre representações vetoriais. Dentre estes trabalhos, podemos destacar [Maas et al., 2011; Ghosh et al., 2015].

$$cos(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{18}$$

1.6- Classificadores

1.6.1 Support Vector Machine e Linear Support Vector Classifier

A Máquina de Vetores de Suporte, do inglês Support Vector Machine (SVM), 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].

Em dados linearmente separáveis, o hiperplano ideal é definido por uma função de decisão linear com margem ótima (máxima) entre os vetores das duas classes [Cortes and Vapnik, 1995]. Um classificador SVM, cujo objetivo é descobrir um hiperplano linear ótimo, é chamado neste trabalho de Linear Support Vector Classifier (LSVC), assim como outros trabalhos na literatura [Ghosh et al., 2015; Schwartz et al., 2015]. Outros pesquisadores também chamam de Linear Support Vector Machines [Siersdorfer et al., 2010; Gamon, 2004].

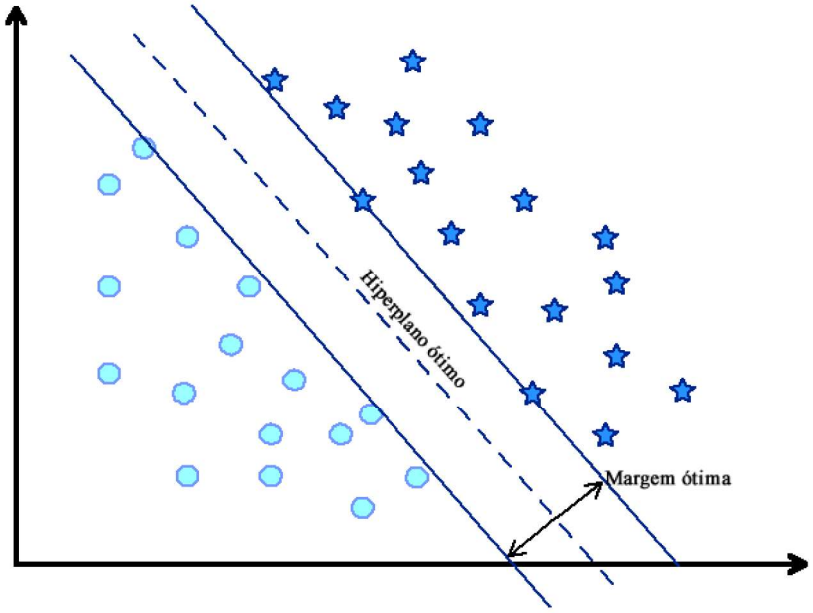

Figura 6 – Exemplo de dados linearmente separáveis com um hiperplano ótimo e uma margem ótima (máxima).

O LSVC visa resolver o problema de otimização exposto na equação 19. Considere

o par (x_i, y_i), sendo x_i um vetor de características de uma amostra e y_i o rótulo de uma amostra de um conjunto de treinamento, dado que $x_i \in \mathbb{R}^n$, $i = 1, ..., l$ e $y_i \in \{-1, 1\}$. O w se refere a um vetor de pesos e $C > 0$ é um parâmetro de penalidade. Nesse problema de otimização, $\varepsilon(w; x_i; y_i)$ se refere a função de custo escolhida, sendo as mais comuns $max(1 - y_i \mathbf{w_T x_i}, 0)$, chamada de L1-SVM e $max(1 - y_i \mathbf{w_T x_i}, 0)^2$, chamada de L2-SVM. Fan et al. [2008] recomendam utilizar a função de custo L2-SVM, a menos que os usuários precisem de um modelo esparso. Na maioria dos casos, a regularização de L1 não dá maior precisão e pode ser um pouco mais lenta no treinamento.

$$wmin = \frac{1}{2} w^T w + C \sum_{i=1}^{l} \varepsilon(w; x_i; y_i) \tag{19}$$

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.

1.6.2 Floresta Aleatória

Floresta Aleatória, em inglês *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 esse conjunto de árvores realizarem um trabalho de classificação, é escolhida a classe que obteve o maior número de votos [Breiman, 2001].

Formalmente, uma Floresta Aleatória é um conjunto de classificadores individuais $\{h(x, \theta_k), k = 1, ..., L\}$, sendo $\{\theta_k\}$ uma família de vetores aleatórios e x os dados de entrada. Breiman [2001] introduz duas propriedades, a força e a correlação, por meio de um limite superior do erro de generalização observado na equação 20. Nesse limite, s representa a força e ρ significa a correlação. A principal conclusão deste resultado é que quanto menor a relação $\frac{\rho}{s^2}$, melhor a floresta, já que oferece maiores chances de obter

uma baixa taxa de erro.

$$PE* \leq \frac{\rho(1-s)}{s^2} \tag{20}$$

Com isso, Breiman [2001] define a função de margem conforme a equação 21, sendo x o dado de entrada, y a classe e P_θ indica a probabilidade da família de vetores aleatórios $\{\theta_k\}$. A força, portanto, é a esperança da margem $mr(x, y)$, conforme equação 22.

$$mr(x, y) = E = P_\theta(h(x,) = y) - \max_{j \neq y} P_\theta(h(x,) = j) \tag{21}$$

$$s = E_{x,y} mr(x, y) \tag{22}$$

Então, Breiman [2001] define a função de margem bruta, conforme a equação 23, sendo $\hat{j}(x, y)$ o índice da classe mais provável das classes erradas, ou seja, da classe que não corresponde a x. O $\hat{j}(x, y)$ é maximizado conforme a equação 24. Logo, a propriedade de correlação corresponde a correlação estatística média entre $rm(\theta, x, y)$ e $rm(\theta', x, y)$ sobre todos os pares de (θ, θ') [Bernard et al., 2010].

$$rm(\theta, x, y) = P_\theta(h(x, \theta) = y) - P_\theta(h(x, \theta) = \hat{j}(x, y) \tag{23}$$

$$\hat{j}(x, y) = arg \max_{j \neq y} P_\theta(h(x, \theta) = j) \tag{24}$$

A generalização da RF não apenas depende da força de cada árvore de decisão na floresta, como também 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.

1.6.3 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 categorias 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, este classificador tende a ter um bom desempenho [McCallum et al., 1998; Lewis, 1998].

Existem dois tipos principais de classificadores Naive Bayes utilizados em classificação de textos, dependendo da forma de distribuição dos atributos ou variáveis aleatórias [McCallum et al., 1998; Lewis, 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 [McCallum et al., 1998; Lewis, 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) [McCallum et al., 1998; Lewis, 1998]. Isto é comum no cenário que 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 categorias 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 25.

$$\hat{c} = \underset{c_j \in C}{\operatorname{argmax}} P(c_k|D) = P(c_k) \prod_{i}^{n} P(w_i|c_k) \tag{25}$$

Expandimos $P(w_i|c_k)$, que representa a probabilidade da palavra w_i pertencer à classe c_k na equação 26. 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 que $\alpha < 1$ a suavização é denominada *Lidstone smoothing* [Tan et al., 2014].

$$P(w_i|c_k) = \frac{x_i \ | \ \alpha}{\sum_{x_i \in T_k} x_i + n\alpha} \tag{26}$$

1.7- F1 score

A área de pesquisa de recuperação de informação (information-retrieval) traz diversas métricas de avaliação de relevância e qualidade de um sistema de busca, sendo o *F1 score* uma delas. O *F1 score* é a média harmônica de outras duas medidas: a precisão e a revocação. Ele também é chamado de *F-measure*, como no trabalho de Forman [2003].

A precisão diz respeito ao percentual de itens classificados como positivos que, na verdade, são positivos [Forman, 2003]. Em outras palavras, a precisão diz respeito ao percentual de acertos de classificação de documentos que foram classificados com uma classe X, realmente são desta classe. A equação 28 ilustra o cálculo da precisão em um contexto de classificação. TP (*true positives*) se referem aos documentos que foram classificados como positivos e de fato são e FP (*false positives*) se referem aos documentos que foram classificados como positivos e que na verdade são negativos.

$$P = \frac{TP}{TP + FP} \tag{27}$$

A outra medida, denominada revocação, em sistemas de busca diz respeito a relevância dos documentos recuperados, enquanto que em aprendizado de máquina diz respeito a porcentagem de documentos positivos que são classificados como positivos. A equação 28 ilustra o cálculo da revocação em um contexto de classificação. TP (*true positives*) se referem aos documentos que foram classificados como positivos e de fato são e FN (*false negatives*) se referem aos documentos que foram classificados como negativos e que na verdade são positivos.

$$R = \frac{TP}{TP + FN} \tag{28}$$

Enfim, o *F1-score*, ou *F1-measure*, é a média harmônica entre as medidas apresentadas anteriormente. Isto é ilustrado na equação 29.

$$F1 = 2 \cdot \frac{R \cdot P}{R + P} \tag{29}$$

1.8- Validação Cruzada *k-fold*

A Validação Cruzada *k-fold*, também chamada de estimativa de rotação, é uma forma de validação de modelos de aprendizado de máquina [Kohavi et al., 1995]. Para estimar capacidade de generalização do modelo, o conjunto de dados é dividido em k subconjuntos aleatórios e mutualmente exclusivos (os *folds*) [Kohavi et al., 1995]. Desses subconjuntos, $k-1$ são utilizados para treinamento e o que restou é utilizado como subconjunto de teste, conforme a figura 7.

Figura 7 – Exemplo de divisão de subconjuntos para validação cruzada 5-fold.

Esse procedimento é repetido k vezes e em cada vez é definido um subconjunto de testes distinto, conforme a figura 8. Em cada iteração é gerado um modelo do qual são capturadas métricas (e.g acurácia e *F1 score*) . Por fim, a capacidade do modelo é obtida pela média das métricas de todas as iterações, como recomendado por Kohavi et al. [1995] em relação a acurácia.

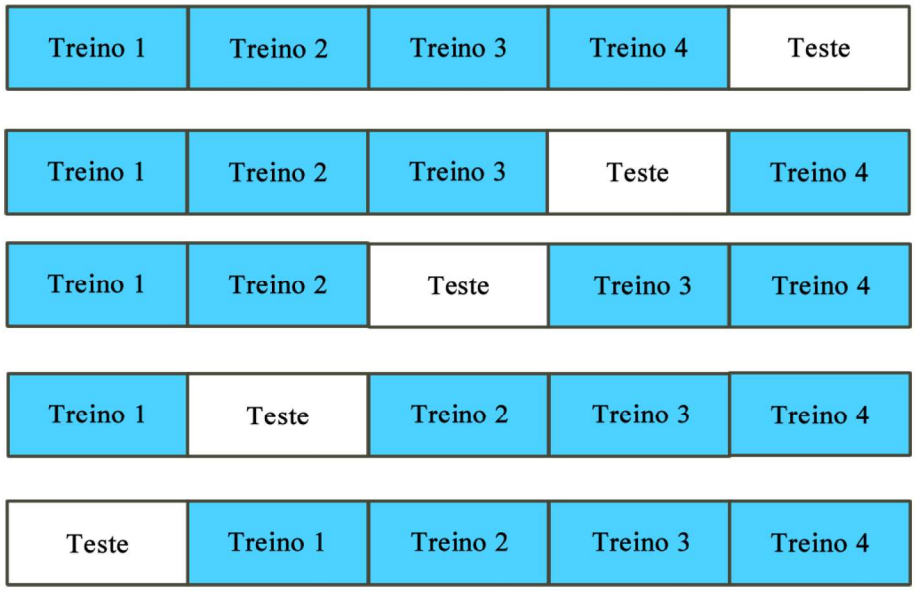

Figura 8 – Exemplo de divisão de subconjuntos para validação cruzada 5-fold em todas as iterações.

2- Trabalhos relacionados

No contexto de trabalhos que lidam com palavras fora do vocabulário se destaca o de Maity et al. [2016], no qual são recolhidos cerca de 1 bilhão de *tweets* para formar um corpus. Então é proposto um modelo de classificação de palavras fora do vocabulário 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 palavras fora do vocabulário. São então definidas 3.500 palavras fora do vocabulário, sendo cada uma delas manualmente anotadas por cinco autores em uma das seis categorias.

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 revocação 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 a revocação 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 categorias: categorias lexicais, categorias de conteúdo e categorias de contexto. As categorias lexicais se relacionam com as propriedades lexicais das palavras em torno das palavras fora do vocabulário, enquanto as categorias de conteúdo se relacionam com o conteúdo dos *tweets* que as palavras fora do vocabulário aparecem. Por conseguinte, as categorias 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 a Regressão Logística, sob o método de validação cruzada de *10-folds*. Também foi adotado o LDA 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 a Regressão Logística obteve melhor área sob a curva ROC em relação ao SVM. Foi observado que as categorias

[1] http://aspell.net/

de conteúdo foram as mais significativas para o alcance do resultado. Logo, há forte diferença semântica entre as seis categorias de palavras fora do vocabulário.

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, foi testado 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 isso, 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 que 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 aos candidatos democratas, no entanto, Bernie Sanders obteve a maior taxa de tweets com sentimento positivo, contrariando os resultados das eleições primárias.

O trabalho de Hartmann et al. [2014] descreve os procedimentos de pré-processamento realizados para tratar palavras fora do vocabulário 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 palavras fora do vocabulário foram detectadas com o uso do vocabulário *Unitex-PB* [Muniz et al., 2005] e parte delas corrigidas com o uso do GNU Aspell. 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

[2]https://www.buscape.com.br

[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 palavras fora do vocabulário presentes no corpus. Com este fim, quatro pares de juízes anotaram manualmente 5.575 *tokens*, correspondentes às palavras fora do vocabulário, indicando quais de 8 categorias cada *token* pertencia. Por fim, foi desenvolvido um *workbench* para normalização textos de avaliações de produtos 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 palavras de opinião. As palavras de opinião são palavras que expressam estados desejáveis (e.g grandes, surpreendentes, etc.) ou indesejáveis (e.g maus, pobres, etc.) [Ding et al., 2008]. O método proposto por Huang et al. [2014] 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 palavras de opinião 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 palavra de opinião 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 Part-Of-Speech (POS) tags, o que pode gerar dificuldades, uma vez que os *POS-Taggers* também perdem 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.

O trabalho de Han et al. [2013] visa tratar palavras fora do vocabulário de textos em inglês do *Twitter* e de Short Message Service (SMS) realizando a normalização de palavras. Este trabalho se concentra na tarefa de normalizar palavras variantes lexicais as levando a sua forma canônica, isso é, contidas em um dicionário. Primeiro são analisados os tipos de palavras fora do vocabulário existentes no domínio de cada conjunto de dados. Então é proposto um método de detectar e normalizar as palavras fora do vocabulário

pelo cálculo de similaridade de *strings*. Os autores também construíram um novo léxico pelo utilizando o método de normalização proposto, além de disponibilizarem o conjunto de dados do *Twitter* que coletaram. Por fim, o impacto da normalização realizada é verificado pelo uso de dois *POS Taggers* em um conjunto de dados do *Twitter* com as *tags* anotadas. Um dos *POS Taggers* era para textos convencionais, enquanto o outro fora construído para o domínio do Twitter. Foi observado que o desempenho de um *POS Tagger* convencional obteve melhora, com o aumento de sua acurácia em 1.6% ($p < 0.01$). No entanto, o uso de um *POS Tagger* construído especificamente para o *Twitter* apresentou desempenho muito superior ao *POS Tagger* convencional. O *POS Tagger* para *Twitter* teve o desempenho piorado ao ser realizada a normalização dos textos do *Twitter*.

Liu et al. [2012] tratam palavras fora do vocabulário com o uso de normalização. O idioma em que esse trabalho se concentra é o inglês. Nele é proposto um método de restaurar uma palavra fora do vocabulário a seu estado canônico, ou seja, contido em um dicionário. Para isso é proposto um método baseado em quatro componentes chave, que recuperam os *top* n candidatos canônicos para uma palavra fora do vocabulário. Assim, três normalizadores sugerem os candidatos mais confiáveis dado uma palavra fora do vocabulário, sendo cada normalizador focado em uma perspectiva diferente. O primeiro normalizador e componente chave é chamado de Enhanced Letter Transformation. Ele aprende um conjunto de transformações de escrita pela geração automática de variações de palavras para normalizar *tokens*. Essas transformações envolvem exclusões, inclusões e substituições de caracteres em palavras. O segundo componente chave e normalizador se chama *Visual Priming*, baseado em um método cognitivo de similaridade visual. O terceiro componente chave é um normalizador denominado *Spell Checker*. Ele combina a similaridade fonética e de *strings*. Por fim, o quarto componente chave consiste em combinar os candidatos encontrados pelos três normalizadores por meio de algumas estratégias. Uma das estratégias proposta consiste em encontrar os *top* n candidatos encontrados em cada normalizador e utilizar estes candidatos como saída do sistema. Isto significa que se foram escolhidos os *top* n candidatos para saída dos normalizadores, a saída será de $3n$ candidatos. Esta estratégia foi denominada *"Oracle"*. Outra estratégia proposta é denominada *Word-level*. Ela foi baseada em prioridades dadas aos normalizadores com maior precisão. Os normalizadores *Enhanced Letter Transformation* e *Spell Checker* possuem maior precisão, logo ganharam maior prioridade nesta estratégia. Baseado na

estratégia *Word-level*, é proposta uma terceira estratégia denominada Message-Level na qual a informação do contexto local é usada para selecionar o melhor candidato. Os experimentos foram conduzidos em quatro conjuntos de dados provenientes do *Twitter* e de SMS. O método proposto atingiu mais de 90% de acurácia na cobertura de palavras em todos os conjuntos de dados.

O trabalho de Wang et al. [2018] se propõe a classificar (em categorias) e predizer substitutos para palavras fora do vocabulário na língua inglesa. Nesse trabalho, cada palavra fora do vocabulário é composta por três partes: a palavra fora do vocabulário, o seu contexto e seus atributos. O contexto de uma palavra, nesse trabalho, se trata de uma breve descrição da palavra fora do vocabulário. Baseado nesses contextos, foram anotados manualmente os atributos correspondentes a cada palavra fora do vocabulário. Para as tarefas de criação dos contextos e de anotação dos atributos foram designados cinco juízes, fluentes em inglês. Assim, cada palavra foi anotada por um subconjunto aleatório de três dos cinco juízes. Os juízes também anotaram categorias para cada palavra fora do vocabulário. Cada uma delas foram classificadas em uma das cinco categorias: mitologia grega, locais, animais, plantas e tecnologia.

Após isso, foram treinados modelos de *word embeddings* utilizando um corpus da Wikipedia, sendo alguns dos modelos treinados com o *Word2Vec* [Mikolov et al., 2013a] e outros modelos com o *Word2GM* [Athiwaratkun and Wilson, 2017]. Os experimentos utilizam similaridade por cosseno para obter os vizinhos mais próximos como possíveis substitutos para as palavras fora do vocabulário. Então, os vizinhos mais próximos de cada palavra são avaliados nos experimentos de acordo com duas pontuações. A primeira pontuação é baseada na posição do primeiro vizinho com a mesma categoria. A segunda pontuação foi baseada na quantidade de atributos em comum. Os resultados indicam que o modelos treinados com o *Word2Vec* obtiveram maior acerto nas duas pontuações definidas.

Em relação ao trabalho de Wang et al. [2018], esta dissertação possui algumas diferenças. Primeiramente, o trabalho de Wang et al. [2018] depende da categorização e anotação manual de cada palavra fora do vocabulário. Além disso, o trabalho não avalia o tratamento de palavras fora do vocabulário em léxicos de AS e o impacto em uma classificação de polaridade. Por fim, o trabalho de Wang et al. [2018] utiliza a língua inglesa.

O trabalho de Zeng et al. [2018] realiza a expansão do LIWC em chinês utilizando

sememes. Sememes são definidos como a menor unidade de significado de linguagem semântica [Bloomfield, 1926]. Para obter os sememes de cada palavra, Zeng et al. [2018] utilizaram o *HowNet* [Dong and Dong, 2003], que possui um vocabulário bem maior que o LIWC em chinês. Então a expansão do LIWC foi considerada um problema de classificação hierárquica com multi-rótulo, qual cada rótulo corresponde a um sememe. A classificação hierárquica é encarada, nesse trabalho, como uma decodificação de seqüência a sequência, em que a entrada é um *word embedding* e a saída são rótulos hierárquicos referentes aos sememes do *HowNet*. Para os *word embeddings* foi utilizado o *Word2Vec* [Mikolov et al., 2013a]. Zeng et al. [2018] então propuseram um modelo denominado Hierarchical Decoder with Sememe Attention (HDSA). A implementação do modelo se baseia em uma rede neural recorrente, mais especificamente a Long Term Short Memory (LSTM) [Hochreiter and Schmidhuber, 1997], que utiliza *embeddings* dos sememes para ganhar informação. Nos experimentos, O HDSA supera modelos do estado-da-arte e os resultados indicam que os sememes foram importantes para obter um léxico de melhor qualidade. A principal fraqueza desse modelo é depender de um dicionário como o HowNet, que possua os sememes para determinado idioma.

Classificamos os trabalhos relacionados de acordo com cada solução apresentada para tratamento de palavras fora do vocabulário na figura 9. O trabalho de Zeng et al. [2018] se enquadra em mais de uma delas. São elas: Normalização, Expansão de Léxico e Classificação.

Este trabalho difere dos demais supracitados. Primeiramente, este trabalho se enquadra na categoria de expansão de léxicos. Mesmo que os trabalhos de Huang et al. [2014], Rezapour et al. [2017] e Zeng et al. [2018] se encaixem na mesma categoria que este trabalho, há diversas diferenças nas abordagens. Diferente do trabalho de Huang et al. [2014], este trabalho não está limitado ao uso de POS-Taggers, ou de uma classe gramatical específica. Este trabalho também difere do de Rezapour et al. [2017], pois não está limitado a expandir o léxico existente com *hashtags* de forma manual. O método de expansão do léxico neste trabalho é realizado por um algoritmo e não envolve apenas *hashtags*, mas qualquer palavra. O trabalho de Zeng et al. [2018] propõe uma rede neural para expandir o LIWC com informação semântica. No entanto, como destacado anteriormente, este trabalho depende de um dicionário de sememes, como o *HowNet*. Ao contrário do trabalho de Zeng et al. [2018], este trabalho utiliza apenas o corpus para treinar os *word embeddings* e as características do próprio léxico para expandí-lo.

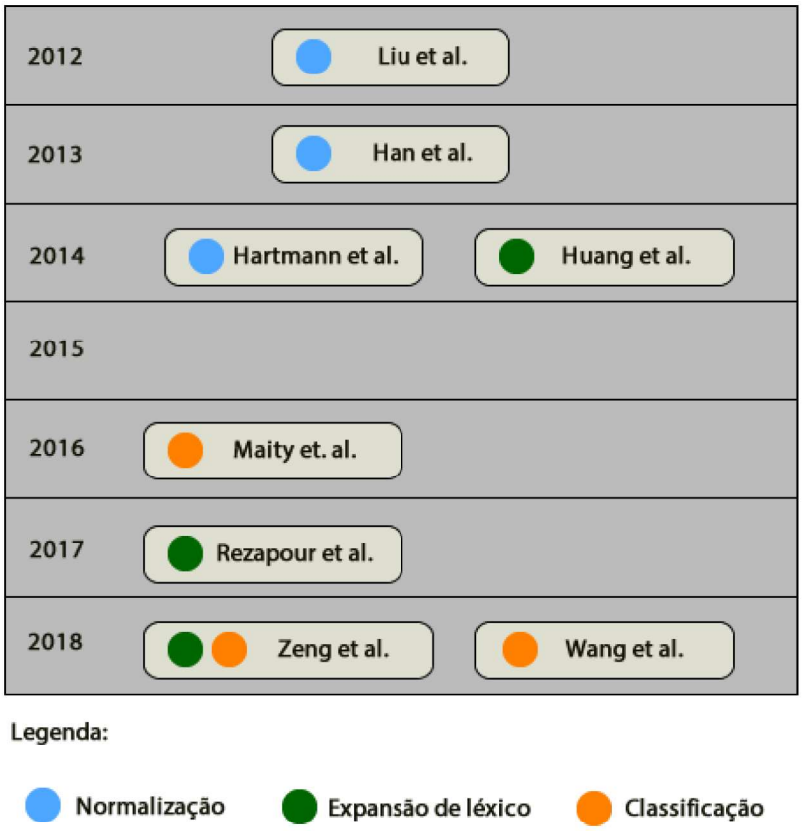

Figura 9 – Síntese dos trabalhos relacionados e suas categorizações.

3- Proposta

Este capítulo possui o objetivo de apresentar a proposta dessa dissertação para resolver o problema de palavras fora do vocabulário em léxicos de Análise de Sentimentos. Este capítulo está organizado conforme a seguir. A seção 3.1 apresenta o algoritmo *IKLex 2*, fruto da pesquisa realizada para esta dissertação. Esta seção descreve o passo a passo realizado por esse algoritmo e como ele gera um novo léxico baseando-se em um modelo de *word embeddings* já treinado utilizando os vizinhos mais próximos. A seção 3.2 apresenta um exemplo de execução do *IKLex 2*.

3.1- Input KNN Lexicon 2

De acordo com a hipótese distribucional, as palavras que ocorrem nos mesmos contextos, tendem a ter o mesmo significado [Harris, 1954]. Baseado nisso, redes neurais que possuem o objetivo de aprender *word embeddings* como o *Word2Vec* [Mikolov et al., 2013a] e o *FastText* [Bojanowski et al., 2016] fazem uso de um contexto de tamanho fixo em sentenças. Intuitivamente, isso significa que palavras que compartilham contextos parecidos sejam semelhantes entre si, e que contextos que compartilham muitas palavras também sejam semelhantes [Goldberg and Levy, 2014]. Dessa maneira, uma palavra ausente em um léxico poderia ser substituída por outra palavra similar, utilizando um espaço vetorial de *word embeddings*, dado que palavras semelhantes compartilham o mesmo contexto.

Por isso, esta dissertação propõe o algoritmo 1, denominado *IKLex 2* (*Input KNN Lexicon 2*). Ele trata as ausências de palavras em um léxico considerando os vetores em um modelo de *word embeddings*. A primeira versão desse algoritmo foi publicada por Nascimento et al. [2018b] como fruto da pesquisa dessa dissertação. No entanto, essa dissertação apresenta uma nova versão do algoritmo, adicionando um novo parâmetro de similaridade mínima que será explicado a seguir.

O *IKLex 2* recebe como entrada um conjunto de documentos D, um conjunto Lex

de palavras contidas no léxico, um modelo de *word embeddings* $WEmb$ já treinado, um número k de vizinhos mais próximos a recuperar e o parâmetro $simMin$ correspondente a similaridade por cosseno mínima. O algoritmo retorna um novo léxico no fim da execução. Dessa maneira, dada uma palavra desconhecida pelo léxico, são obtidas as k palavras com maior similaridade por cosseno no modelo de word embeddings recebido como parâmetro, de maneira ordenada. A diferença entre o *IKLex 2* e o *IKLex* é que no *IKLex 2* as palavras com uma similaridade por cosseno menor que uma similaridade mínima definida como parâmetro são desconsideradas. Isso possibilita calibrar o algoritmo para que o léxico não substitua uma palavra por outra dissimilar.

Portanto, as categorias da palavra ausente no léxico serão as mesmas categorias da palavra mais próxima que faça parte do léxico. Dessa forma, uma palavra ausente é considerada como se fosse a mais próxima semanticamente. No final do algoritmo é retornada uma versão nova do léxico fornecido como entrada, contendo um novo conjunto de palavras.

Algoritmo 1 – IKLex2(D, Lex, $WEmb$, k, $simMin$)

Input:
- D = Conjunto de documentos
- Lex = Conjunto de palavras contidas no léxico
- $WEmb$ = Modelo de Word Embeddings treinado
- k = Número de vizinhos mais próximos
- $simMin$ = Similaridade por cosseno mínima

Output: $nLex$, léxico modificado

```
nLex ← obtemCopiaLexico(Lex)
voc ← obterVocabulario(D)
for all w ∈ voc do
    if w ∉ Lex then
        kVizinhos ← obterKNNOrd(w, k, WEmb)
        kVizinhos ← removerPalavrasSimMin(kVizinhos, simMin)
        nLex[w] ← 0⃗
        for all v ∈ kVizinhos do
            if v ∈ Lex then
                nLex[w] ← obterCaracteristicasPalavra(v, Lex)
                break
            end if
        end for
    end if
end for
return nLex
```

O *IKLex 2* funciona da seguinte forma. O passo inicia o novo léxico obtendo uma

cópia do léxico original. O passo 2 obtém o vocabulário de todos os documentos do conjunto de dados. Para cada palavra w do vocabulário, o passo 4 verifica se o léxico não a contém. Caso o léxico não contenha a palavra, o passo 5 obtém as k palavras mais próximas (ordenado da mais próxima para a mais distante por meio da similaridade por cosseno) com base no modelo de *word embeddings* $WEmb$. Posteriormente são removidas as palavras com similaridade por cosseno menor que a definida no parâmetro $simMin$, no passo 6.

Em seguida, para cada palavra v contida nos k vizinhos mais próximos, o passo 9 verifica se a palavra v está contida no léxico. Caso esteja, o passo 10 atribui as categorias da palavra v no léxico para a palavra w no novo léxico e descarta os demais vizinhos mais próximos. Caso nenhum dos vizinhos mais próximos esteja contido no léxico, é atribuído um vetor nulo $\vec{0}$ às categorias, conforme inicialização da variável efetuada no passo 7. Por fim, o algoritmo retorna um novo léxico que será utilizado para a classificação de polaridade.

3.2- Exemplo de Execução do IKLex 2

A figura 10 ilustra um exemplo da execução do *IKLex 2*. Considere os parâmetros $k = 3$ e $simMin = 0,5$. O algoritmo recebe como entrada o léxico de AS, um conjunto de documentos e um modelo de *word embeddings* já treinado. Então, algoritmo produz o vocabulário do conjunto de documento. Para cada palavra do vocabulário que não esteja contida no léxico de AS, ele busca os 3 vizinhos mais próximos e suas respectivas similaridades por cosseno.

Na figura 10, as palavras que não estão no léxico de AS e estão contidas no vocabulário do conjunto de documentos são: Amo, Laranja, Fazer e Gata. Para destacar os vizinhos mais próximos para cada palavra fora do vocabulário é apresentada uma matriz, na qual os vizinhos mais próximos que estão contidos no léxico de AS e que atenderam a condição de similaridade mínima por cosseno se encontram destacados em verde. Os vizinhos mais próximos que estão contidos no léxico de AS mas que não atenderam a condição de similaridade mínima por cosseno se encontram destacados em azul. Por exemplo, a palavra “Amo”, o vizinho mais próximo contido no léxico de

AS é a palavra “Amor”, que obteve a similaridade por cosseno de $0,75$. Para a palavra “Laranja”, o vizinho mais próximo contido no léxico de AS é a palavra “Maçã”, no entanto a similaridade por cosseno é de $0,35$, não obedecendo a similaridade mínima por cosseno.

Ao final da execução do algoritmo é produzido um novo léxico contendo as palavras do léxico recebido como entrada e mais as palavras “Amo”, “Fazer” e “Gata”. A palavra “Amo” recebe as características da palavra “Amor”, a palavra “Fazer” recebe as características da palavra “Executar” e a palavra “Gata” recebe as características da palavra “Gato”. Portanto, a palavra “Amo” é tratada como a palavra “Amor”, a palavra “Fazer” como a palavra “Executar” e a palavra “Gata” como a palavra “Gato”.

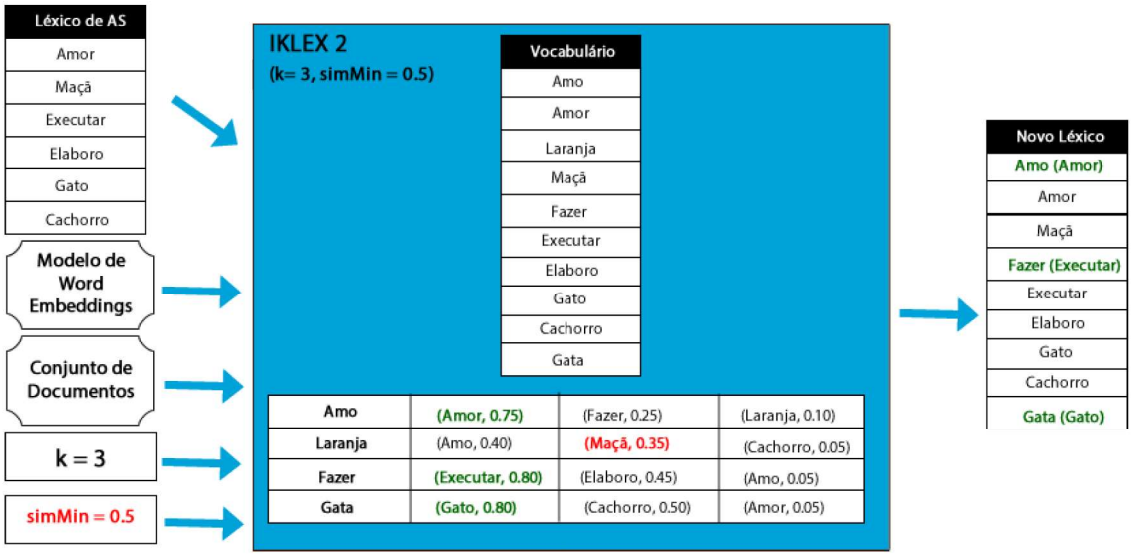

Figura 10 – Exemplo de execução do IKLex 2 ilustrando o léxico produzido.

4- Avaliação Experimental

Este capítulo discorre sobre os experimentos realizados neste trabalho. Tais experimentos são divididos em dois tipos: os experimentos que utilizam o *IKLex 2* e os experimentos que utilizam apenas o LIWC. Os experimentos aconteceram em três conjuntos de dados em Português do Brasil. Os três conjuntos de dadossão derivados de redes sociais. Foram realizados diversos experimentos com o algoritmo *IKLex 2*, variando seus dois parâmetros.

Este capítulo está organizado como segue: a seção 4.1 explica a metodologia utilizada para conduzir os experimentos. A seção 4.2 descreve os conjuntos de dados utilizados nos experimentos e apresenta uma pequena análise exploratória. A seção 4.3 apresenta as propriedades gerais dos experimentos realizados. A seção 4.4 discorre sobre a execução do algoritmo *IKLex 2* e apresenta gráficos de quantas novas palavras e termos foram tratados. A seção 4.5 apresenta os classificadores utilizados e os parâmetros definidos. A seção 4.6 apresenta os resultados obtidos para cada conjunto de dados. A seção 4.7 apresenta um teste de hipótese para descobrir se as diferenças das médias obtidas nos experimentos são estatisticamente significantes. Por fim, a seção 4.8 discute sobre os resultados obtidos.

4.1- Metodologia

Foi definida uma metodologia a ser seguida para conduzir os experimentos. Os experimentos devem seguir dois fluxos de trabalho diferentes. O primeiro fluxo de trabalho diz respeito aos passos definidos para preparar os experimentos conduzidos apenas com o léxico, sem alterações. Esse fluxo pode ser visualizado na figura 11.

A explicação dos passos desse primeiro fluxo é dada como segue. Dado um conjunto de dados original, representado por um conjunto de documentos, é realizada uma fase de pré-processamento que produzirá um conjunto de dados pré-processado. Após essa etapa, é realizada a preparação das categorias, que varia de acordo com o

léxico utilizado (neste trabalho é utilizado o LIWC).

A fase de preparação das categorias prepara o conjunto de treinamento com base nas características do léxico recebido como entrada. Por sua vez, o conjunto de treinamento é utilizado para a fase de treinamento de classificadores. O treinamento dos classificadores tem como saída as métricas para a fase de avaliação dos classificadores. As métricas são utilizadas para definir a qualidade dos classificadores treinados.

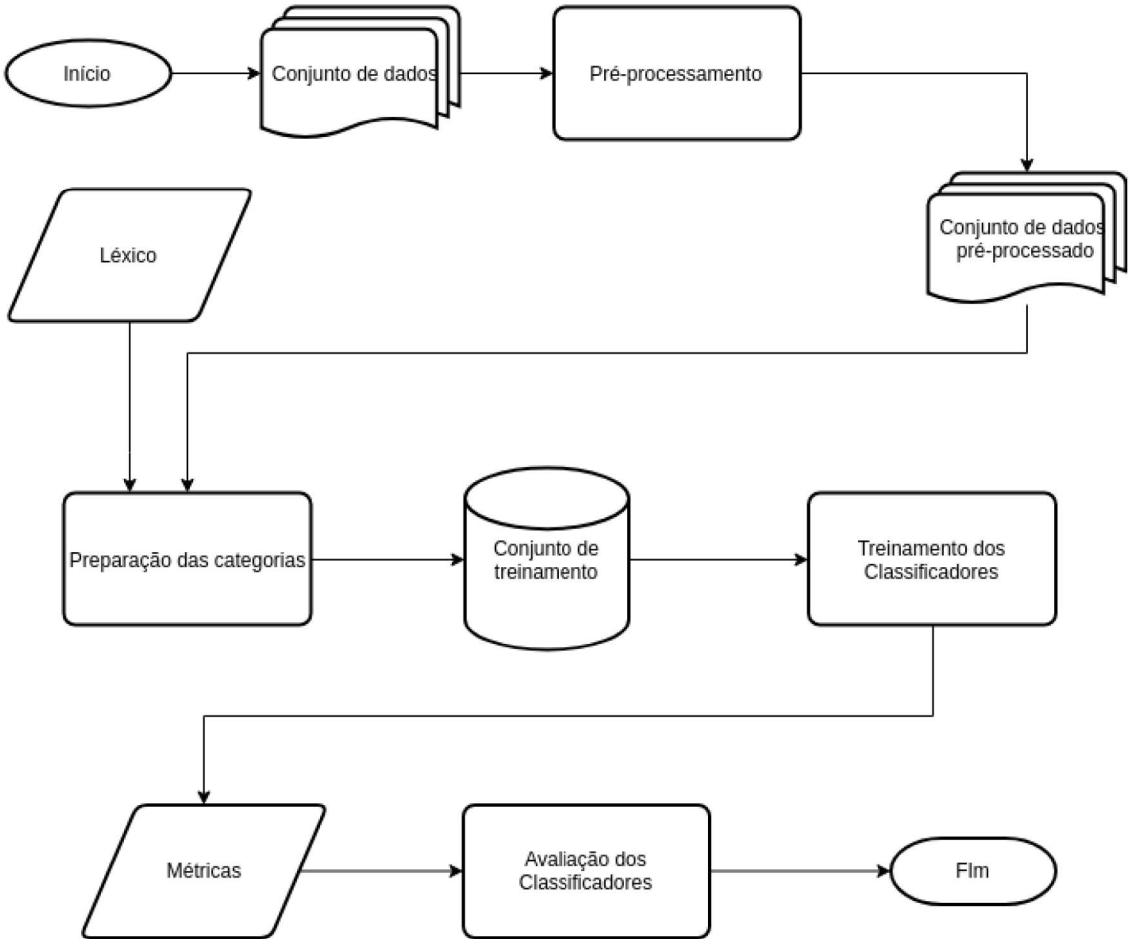

Figura 11 – Fluxo de trabalho para os experimentos realizados sem o IKLex

O fluxo de trabalho mostrado na figura 12 descreve os passos realizados para os experimentos conduzidos com o uso do algoritmo *IKLex 2*. Assim como no fluxo de trabalho anterior, dado um conjunto de dados original, representado por um conjunto de documentos, é realizada uma fase de pré-processamento que produzirá um conjunto de dados pré-processado. Após essa etapa, o conjunto de dados pré-processado é utilizado para realizar o treinamento dos *word embeddings*.

Por conseguinte, ao obter o modelo de *word embeddings* completamente treinado, o *IKLex 2* é executado, recebendo o léxico original como entrada, o modelo de *word*

embeddings e o conjunto de dados pré-processado. O *IKLex 2* gera um novo léxico, que é utilizado como entrada junto com o conjunto de dados pré-processado para o próximo passo de preparação das categorias.

A fase de preparação das categorias prepara um conjunto de treinamento que é utilizado para a fase de treinamento de classificadores. O treinamento dos classificadores tem como saída as métricas para a fase de avaliação dos classificadores. As métricas são utilizadas para definir a qualidade dos classificadores treinados e serão comparadas com as métricas dos classificadores do fluxo da figura 11.

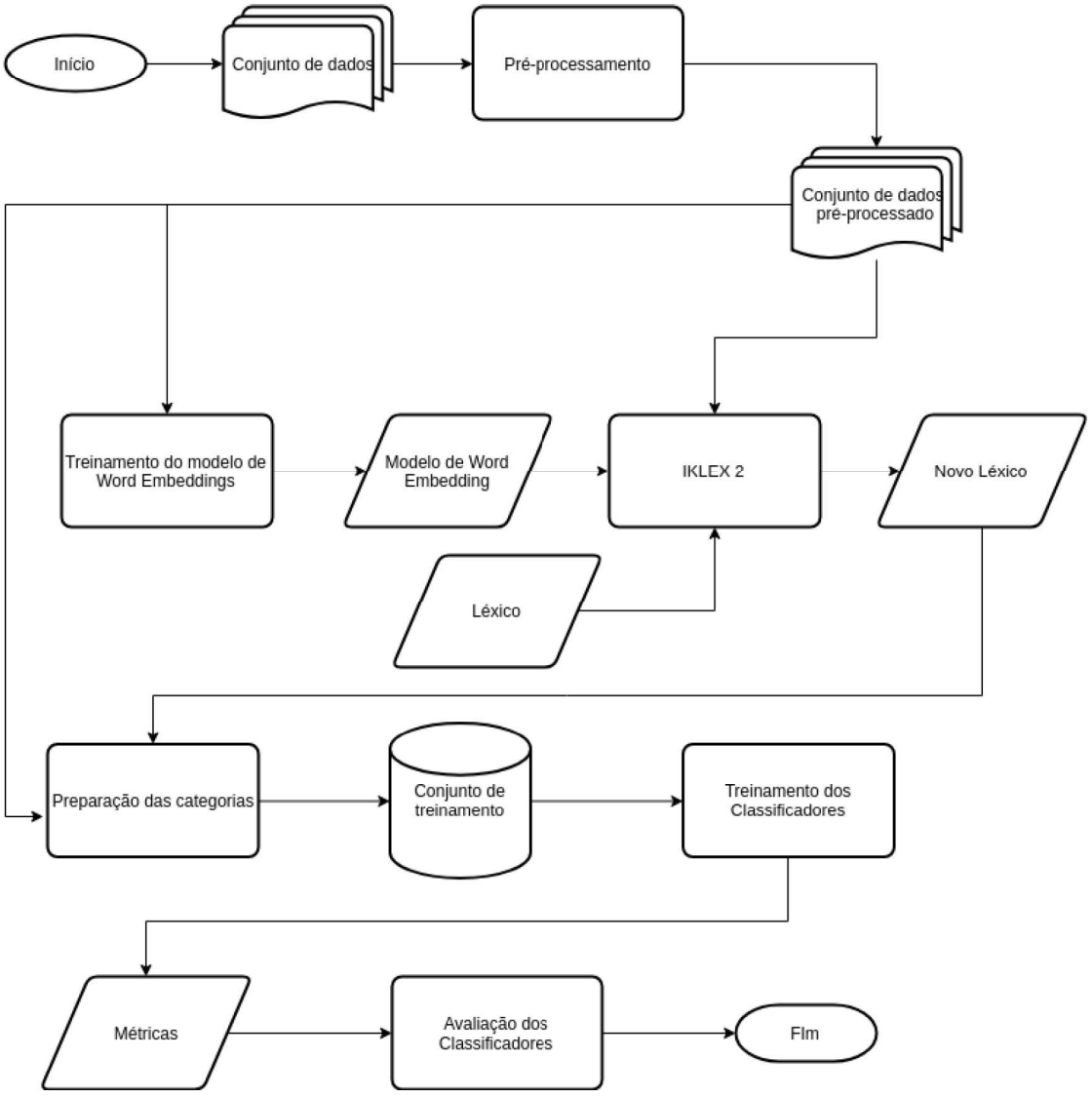

Figura 12 – Fluxo de trabalho para os experimentos realizados com o IKLex 2

4.2- Conjuntos de Dados

Foram selecionados três conjuntos de dados para os experimentos realizados nesta dissertação. Um dos conjuntos de dados é denominado $MQDEmotion2018$[1]. Esse conjunto de dados foi extraído de uma rede social brasileira denominada Meu Querido Diário[2] e foi publicado por Nascimento et al. [2018a]. Nesta rede social usuários possuem um diário virtual, no qual descrevem seus dias, podendo interagir com outros usuários ao redigirem 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 foi extraída uma amostra contendo as emoções de felicidade e tristeza. Essa amostra possui 5.000 documentos, sendo 2.500 documentos correspondendo a emoção *felicidade* e 2.500 correspondendo à emoção *tristeza*.

Outro conjunto de dados utilizado para os experimentos é denominado *TAS-PT* [3], publicado por Cavalcante [2017]. Trata-se de um conjunto de dados retirado do twitter para análise de polaridade. Para realizar a coleta, *tweets* com os *emoticons* “:)”ou “:-)”são rotuladas como positivas e mensagens com os *emoticons* “:(”ou “:-(”são rotuladas como negativas [Cavalcante, 2017].

Infelizmente devido a regras de privacidade do *Twitter*, o conjunto de dados não foi publicado com todo o texto, sendo necessário utilizar a Interface de Programação de Aplicações (do inglês, *Application Programming Interface - API*) do *Twitter* para obtê-los. Devido a isso, não foi possível obter o número original dos *tweets*, mas foi possível recuperar um bom número. Foram recuperados com a API 30407 *tweets* positivos e 28853 *tweets* negativos. Após recuperar os *tweets* do conjunto de dados *TAS-PT*, foi removido de cada *tweet* os *emoticons* “:)”, “:-)”, “:(”e “:-(”para que não afetem os experimentos.

O terceiro conjunto de dados foi publicado por Kansaon et al. [2018]. Apesar das políticas de privacidade, os textos dos *tweets* foram disponibilizados[4]. Foram coletados pelos autores *tweets* de diversos sentimentos associados. Foi definido por Kansaon et al. [2018] que o *tweet* precisa conter a hashtag com o nome do sentimento selecionado

[1] https://github.com/LaCAfe/MQDEmotion2018.git
[2] http://www.mqd.com.br
[3] https://github.com/pauloemmilio/dataset
[4] https://github.com/danielkansaon/BraSNAM2018-Dataset-Analise-de-sentimentos-em-tweets-em-portugues-brasileiro

[Kansaon et al., 2018]. Assim, foram coletados *tweets* que possuem as *hashtags*: #Triste, #Chateado, #Feliz, #Amor, #Raiva, #Inveja, #Ironia [Kansaon et al., 2018]. Essas *hashtags* foram removidas do conjunto de dados para que não afetem os experimentos.

Neste trabalho, consideramos apenas os *tweets* com as *hashtags* #Feliz e #Triste para a classificação de polaridade. Os *tweets* que correspondem aos sentimentos de felicidade totalizam 1961 e os que correspondem ao sentimento de tristeza totalizam 2787.

4.2.1 Análise exploratória

Uma vez definidos os conjuntos de dados utilizados nos experimentos, foi preciso realizar uma breve análise exploratória dos dados. Foram definidos alguns parâmetros para análise a seguir: o tamanho do vocabulário, os termos do vocabulário fora do léxico LIWC, o número de palavras, o número de palavras fora do léxico do LIWC, média de palavras por documento, média de palavras fora do léxico do LIWC por documento.

Em relação a terminologia utilizada nesta subseção, é necessário uma breve explicação. Os termos do vocabulário são palavras contidas no vocabulário, sem levar em conta sua cardinalidade no conjunto de dados. Por sua vez, as palavras levam em conta a cardinalidade de seu termo correspondente contido no conjunto de dados.

A tabela 1 apresenta esses parâmetros para o conjunto de dados MQD. É possível observar, nesse conjunto de dados, que os termos que não estão presentes no léxico do LIWC correspondem a $68,25\%$ de todo o vocabulário presente. Esses termos fora do vocabulário (fora do léxico do LIWC) correspondem a 319.115 palavras de um total de $1.165.311$.

Tabela 1 – Análise exploratória do conjunto de dados MQD

Parâmetro	Valor
Tamanho do vocabulário	48.285 termos
Termos do vocabulário fora do léxico do LIWC	32.955 termos
Número de palavras	$1.165.311$ palavras
Número de palavras fora do léxico do LIWC	319.115 palavras
Média de palavras por documento	$233,06$ palavras
Média de palavras fora do léxico do LIWC por documento	$63,82$ palavras

A tabela 2 apresenta os parâmetros do conjuntos de dados *TAS-PT*. Por se tratar de um conjunto de dados proveniente do *Twitter* o número de palavras por documento é

bem inferior que o do conjunto de dados MQD. O número de palavras fora do vocabulário é de grande proporção, chegando em média a 34% do conteúdo de cada documento.

Tabela 2 – Análise exploratória do conjunto de dados TAS-PT

Parâmetro	Valor
Tamanho do vocabulário	67.782 termos
Termos do vocabulário fora do léxico do LIWC	56.240 termos
Número de palavras	635.012 palavras
Número de palavras fora do léxico do LIWC	215.601 palavras
Média de palavras por documento	$10,72$ palavras
Média de palavras fora do léxico do LIWC por documento	$3,64$ palavras

A tabela 3 apresenta os parâmetros para o conjunto de dados *KANSAON*. Conforme exposto, existem muitas palavras fora do léxico do LIWC, chegando a cerca de $40,6\%$ de todas as palavras contidas no conjunto de dados. A quantidade de termos no vocabulário que estão ausentes no léxico do LIWC também é expressiva.

Tabela 3 – Análise exploratória do conjunto de dados KANSAON

Parâmetro	Valor
Tamanho do vocabulário	11.859 termos
Termos do vocabulário fora do léxico do LIWC	8.029 termos
Número de palavras	62.635 palavras
Número de palavras fora do léxico do LIWC	25.435 palavras
Média de palavras por documento	$13,19$ palavras
Média de palavras fora do léxico do LIWC por documento	$5,36$ palavras

4.3- Propriedades gerais dos experimentos

Para realizar os experimentos, foi necessário realizar o pré-processamento dos conjuntos de dados. Primeiramente, foi realizada uma conversão de todas as letras maiúsculas em letras minúsculas. Depois, foram removidos todos os *links* do texto.

Posteriormente, ainda na fase de pré-processamento, foi realizado um processo chamado de *tokenização*. O processo de *tokenização* consiste em dividir o texto em unidades menores. Essas unidades, denominadas *tokens*, podem ser palavras, sentenças, frases e parágrafos Stavrianou et al. [2007]. Esta dissertação utiliza *tokens* de palavras. O processo de tokenização permitiu realizar a análise exploratória descrita na seção 4.2.1.

Em seguida, foram removidas todas as palavras que ocorrem menos que cinco

vezes em cada conjunto de dados, conforme adotado por Kusner et al. [2015]. Por fim, é realizado um outro processo de *tokenização* por sentenças, para que sejam treinados os *word embeddings*. Com todo o processo de *tokenização* concluído, é possível iniciar a fase de preparação das categorias para realizar o treinamento dos classificadores que fazem uso de características do LIWC.

Uma vez realizado o pré-processamento, para os experimentos realizados com o *IKLex 2* é necessário realizar o treinamento dos modelos de *word embeddings* utilizando os *tokens*. Os modelos de *word embeddings* foram treinados utilizando o *Word2Vec* com a arquitetura *Skip-Gram*. Para realizar o treinamento, alguns parâmetros foram definidos. Os vetores gerados possuem 300 dimensões, conforme utilizado em diversos trabalhos na literatura [Mikolov et al., 2013a; Ouyang et al., 2015; Ombabi et al., 2017]. O tamanho do contexto (*window size*) utilizado corresponde a 5, conforme adotado por Mihaylov and Nakov [2016] e palavras com menos de cinco ocorrências foram desconsideradas, conforme adotado por Kusner et al. [2015].

Outra propriedade definida para os experimentos é o léxico utilizado. O léxico escolhido nesta dissertação é o LIWC, mais especificamente a versão 2007 em Português do Brasil. Conforme descrito na seção 1.3, o LIWC possui o objetivo de extrair e analisar componentes emocionais, cognitivos e estruturais presentes na fala e em textos [Pennebaker et al., 2003]. O LIWC é adotado em diversos trabalhos de Análise de Sentimentos, como em [Reis et al., 2015] e [Araújo et al., 2013].

A escolha do léxico impacta diretamente na fase de preparação das categorias. No caso do LIWC 2007 para Português do Brasil, cada palavra possui 64 categorias. Assim, na fase de preparação de categorias é possível representar cada palavra como um vetor de 64 dimensões. Os vetores que representam cada palavra contida em um documento são somados para representar cada documento no conjunto de dados. Essa abordagem foi adotada por trabalhos como [Tavares and Guedes, 2017] e [Rodrigues et al., 2017]. A figura ilustra um vetor de n dimensões representando uma sentença. Cada dimensão em representa uma categoria do LIWC. Na sentença apresentada, existem duas palavras que possuem a categoria correspondente à dimensão x_2. Para o LIWC 2007 em Português do Brasil, $n = 64$.

Os léxicos gerados pelo *IKLex 2*, neste trabalho, possuem as mesmas propriedades do léxico do LIWC, por se tratarem de uma expansão do último. Cada palavra é representada por um vetor de 64 categorias. Dessa forma, todos os documentos são

Figura 13 – Vetor representando uma sentença. Retirado de [Rodrigues et al., 2017]

representados por um vetor de 64 dimensões.

4.4- Execução do IKLex 2

Após o treinamento dos *word embeddings* para cada conjunto de dados, é possível executar o *IKLex 2* para gerar o novo léxico expandido a partir do LIWC para realizar os experimentos. Com isso, esta seção ilustra o número termos e de palavras tratadas com o *IKLex 2* para cada conjunto de dados. Assim, é possível analisar o comportamento de como funciona a execução do algoritmo *IKLex 2* e ter ideia de como o novo léxico gerado difere do LIWC e como ele pode impactar nos experimentos.

Esta seção apresenta duas figuras para cada conjuntos de dados. Cada figura contém dez gráficos. Uma figura se refere ao número de termos tratados pelo *IKLex 2* e a outra figura se refere ao número de palavras tratadas pelo *IKLex 2*. A diferença entre os termos e as palavras é que os termos são palavras contidas em um vocabulário, sem levar em conta o número de vezes que esse termo ocorre no conjunto de dados. As palavras possuem um termo único correspondente no vocabulário ou léxico e podem ocorrer uma ou mais vezes no conjunto de dados. Portanto os gráficos que se referem às palavras levam em conta essas ocorrências.

Em cada conjunto de dados o *IKLex 2* foi executado diversas vezes, variando os parâmetros. Cada um dos dez gráficos ilustrados em cada figura diz respeito a execução do *IKLex 2* para um parâmetro k, variando de 1 a 10. Cada gráfico possui um eixo x que se refere às diferentes execuções do *IKLex 2*, variando o parâmetro de similaridade por cosseno mínima $simMin$ de acordo com um progressão aritmética finita, no intervalo inclusivo de $0,05$ a $0,90$, no qual $simMin_n = simMin_1 + (n-1)r$, $n \geq 1$ e $n \leq 18$ e $r = 0,05$.

A figura 14 ilustra o número de termos ausentes no léxico do LIWC tratados pelo

IKLex 2 no conjunto de dados MQD. O número máximo de termos tratados pelo *IKLex 2* e adicionados ao novo léxico gerado é de 2.921. Os gráficos indicam que o número dos termos tratados pelo *IKLex 2* aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,5$ e $0,25$, havendo um platô quando $simMin < 0,25$. Também há um aumento do número de termos tratados pelo *IKLex 2* em relação ao parâmetro k.

A figura 15 ilustra o número de palavras tratadas pelo novo léxico gerado pela execução do algoritmo *IKLex 2*. Ao todo, o número máximo de palavras tratadas foi 226.940. O número de palavras aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,60$ e $0,25$, havendo um platô quando $simMin < 0,25$. Também há um aumento do número de palavras tratadas pelo *IKLex 2* em relação ao parâmetro k.

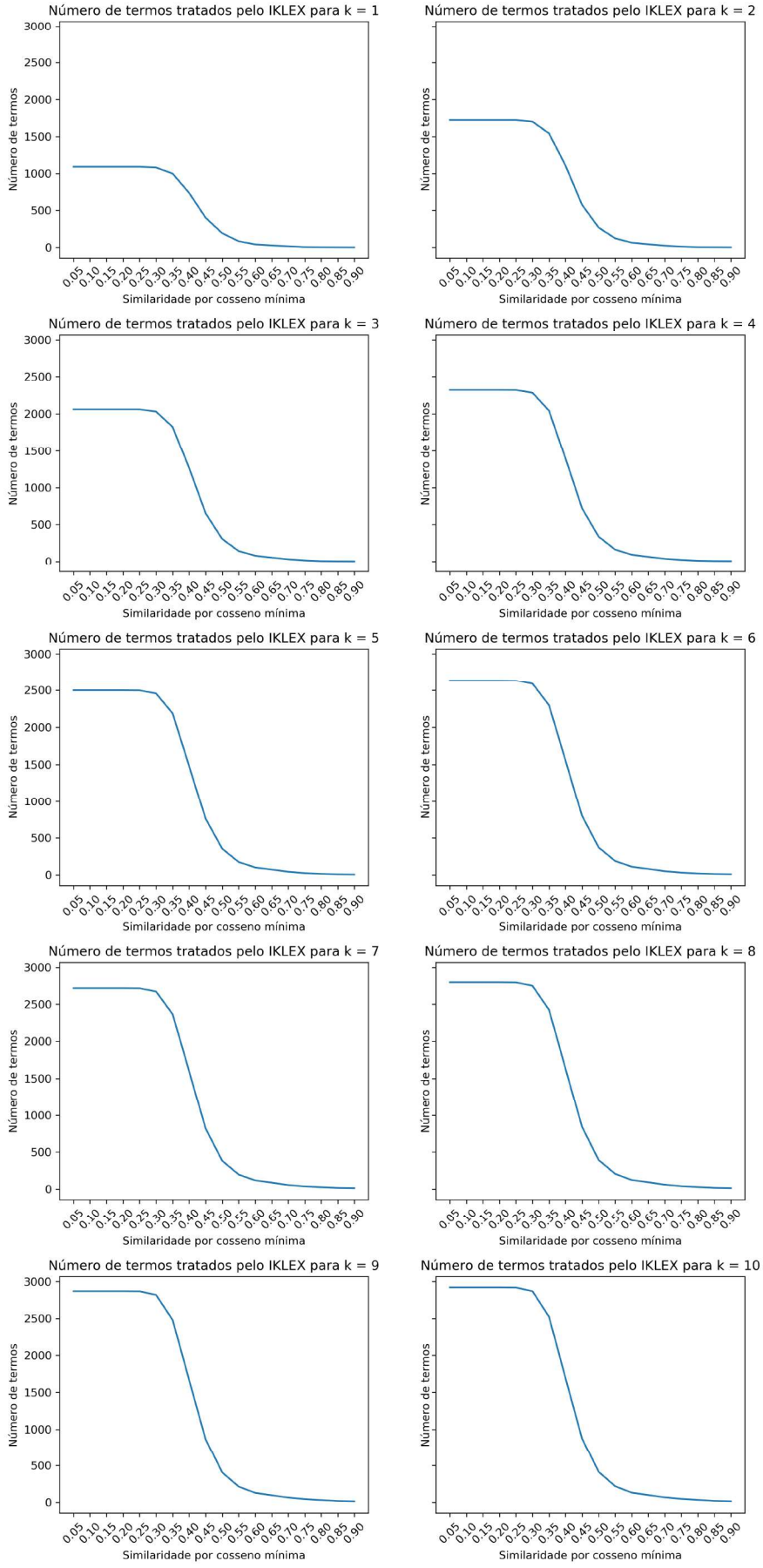

Figura 14 – Número de termos tratados pelo IKLex 2 no conjunto de dados MQD

A figura 16 ilustra o número de termos ausentes no léxico do LIWC tratados pelo *IKLex 2* no conjunto de dados TAS-PT. O comportamento é semelhante ao do conjunto de dados MQD, apresentado na figura 14. O número máximo de termos tratados pelo *IKLex 2* e adicionados ao novo léxico gerado é de 2.872. Os gráficos indicam que o número dos termos tratados pelo *IKLex 2* aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,5$ e $0,25$, havendo um platô quando $simMin < 0,25$. Também há um aumento do número de termos tratados pelo *IKLex 2* em relação ao parâmetro k.

A figura 17 ilustra o número de palavras tratadas pelo novo léxico gerado pela execução do algoritmo *IKLex 2*. Ao todo, o número máximo de palavras tratadas foi 135.090. O número de palavras aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,50$ e $0,25$, havendo um platô quando $simMin < 0,25$. Também há um aumento do número de palavras tratadas pelo *IKLex 2* em relação ao parâmetro k.

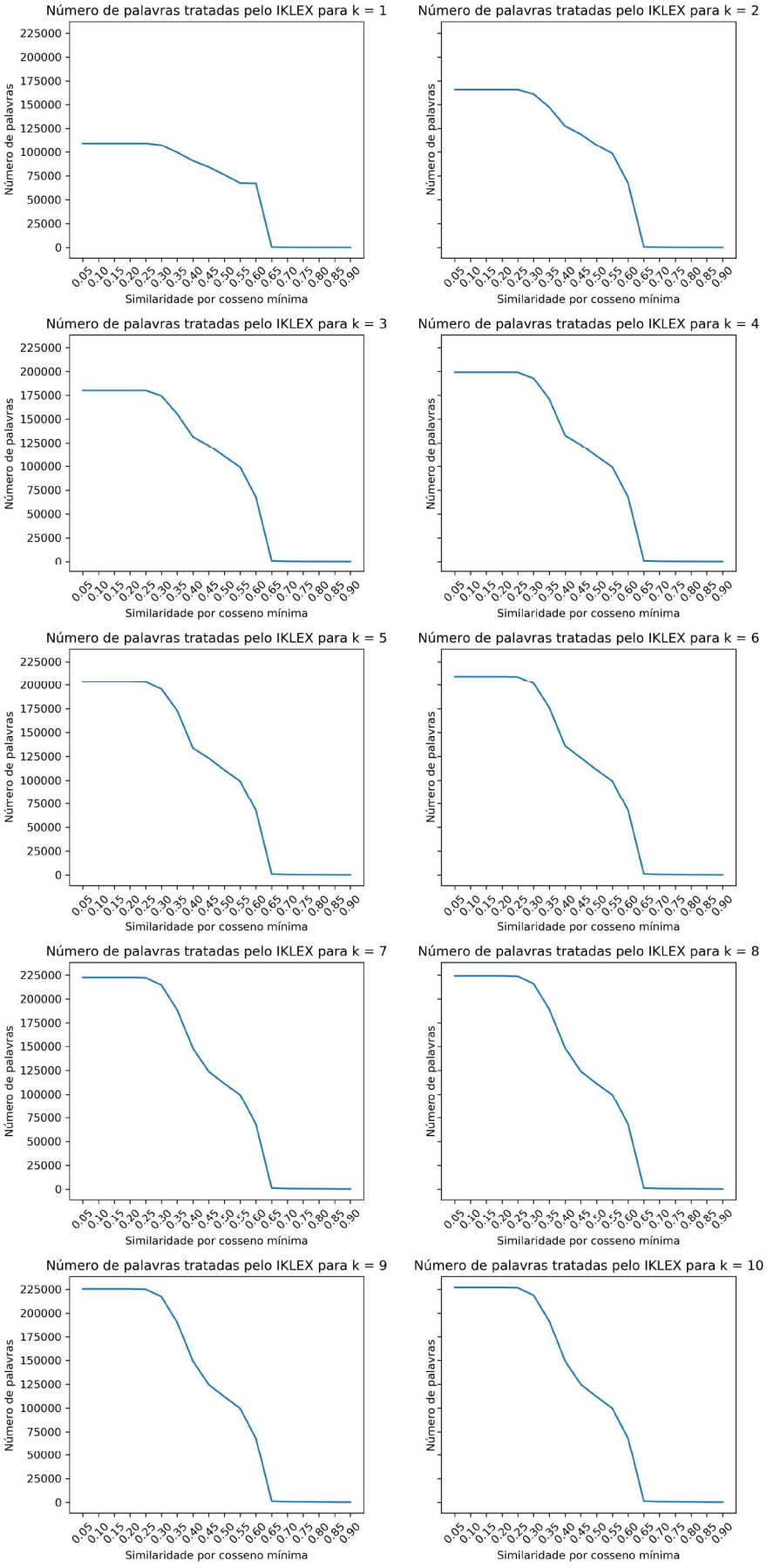

Figura 15 – Número de palavras tratadas pelo IKLex 2 no conjunto de dados MQD

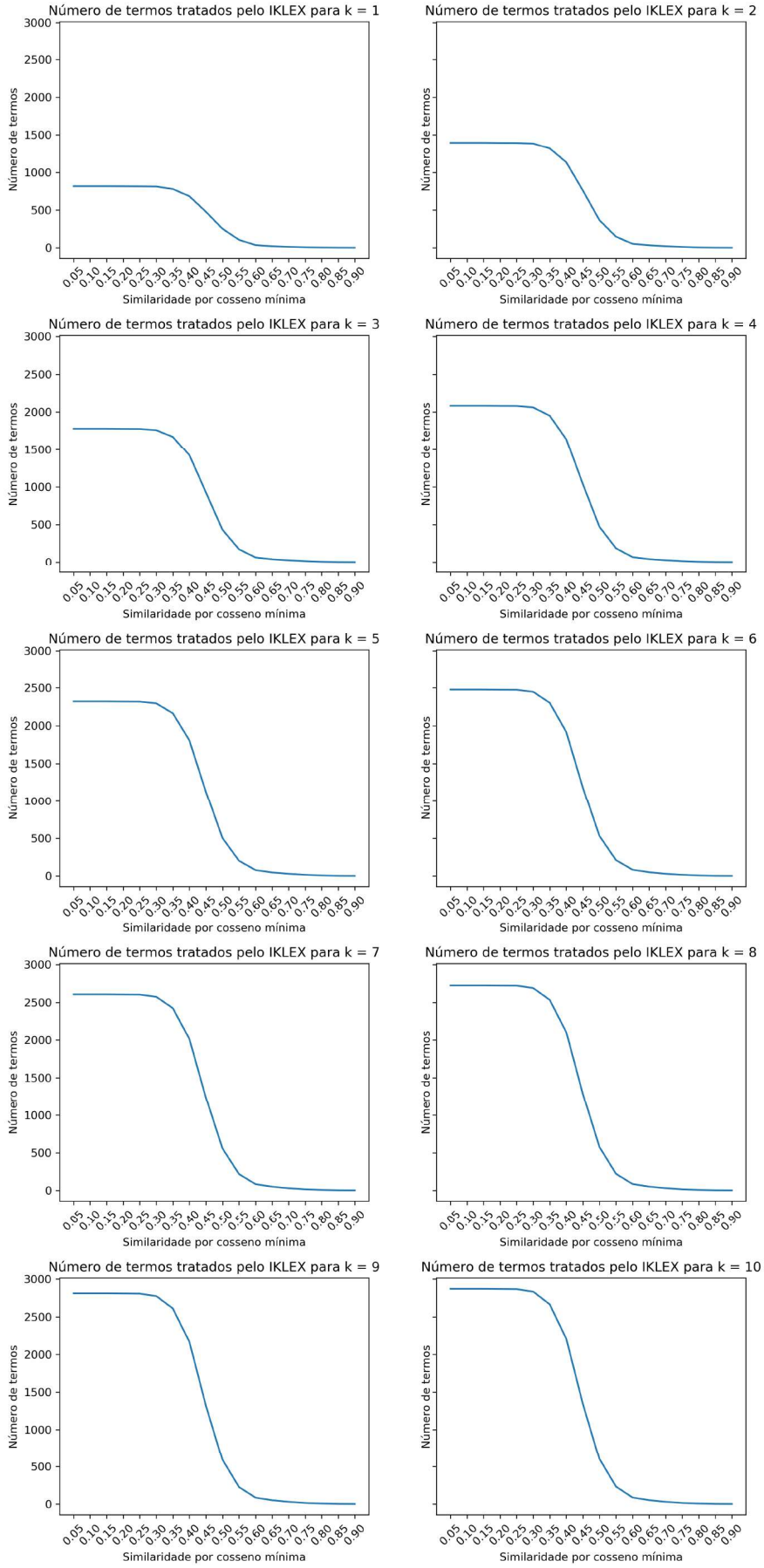

Figura 16 – Número de termos tratados pelo IKLex 2 no conjunto de dados TAS-PT

A figura 18 ilustra o número de termos ausentes no léxico do LIWC tratados pelo *IKLex 2* no conjunto de dados KANSAON. O comportamento é semelhante aos dos conjuntos de dados MQD e TAS-PT, apresentados nas figuras 14 e 16. O número máximo de termos tratados pelo *IKLex 2* e adicionados ao novo léxico gerado é de apenas 310. Nenhum termo único foi tratado quando o parâmetro $simMin$ é $0,90$. Os gráficos indicam que o número dos termos tratados pelo *IKLex 2* aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,6$ e $0,3$, havendo um platô quando $simMin < 0,45$. Assim como nos outros conjuntos de dados, há um aumento do número de termos tratados pelo *IKLex 2* em relação ao parâmetro k.

A figura 19 ilustra o número de palavras tratadas pelo novo léxico gerado pela execução do algoritmo *IKLex 2*. Ao todo, o número máximo de palavras tratadas foi 13.189. O número de palavras aumenta de forma mais acentuada quando $simMin$ se encontra no intervalo entre $0,50$ e $0,25$, havendo um platô quando $simMin < 0,3$. Como nenhum termo único é tratado quando $simMin$ é $0,90$, o mesmo ocorre com as palavras. Também há um aumento do número de palavras tratadas pelo *IKLex 2* em relação ao parâmetro k.

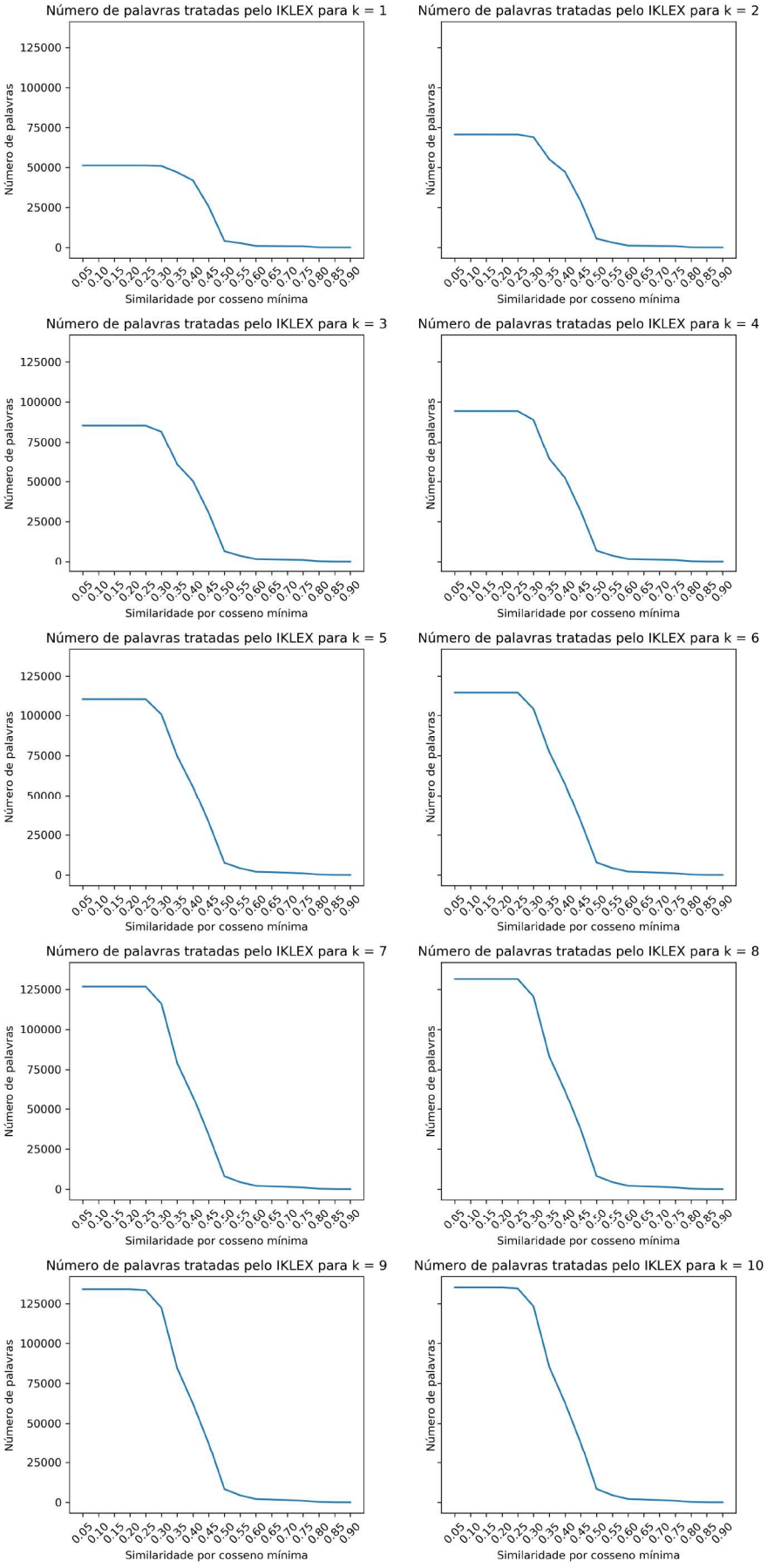

Figura 17 – Número de palavras tratadas pelo IKLex 2 no conjunto de dados TAS-PT

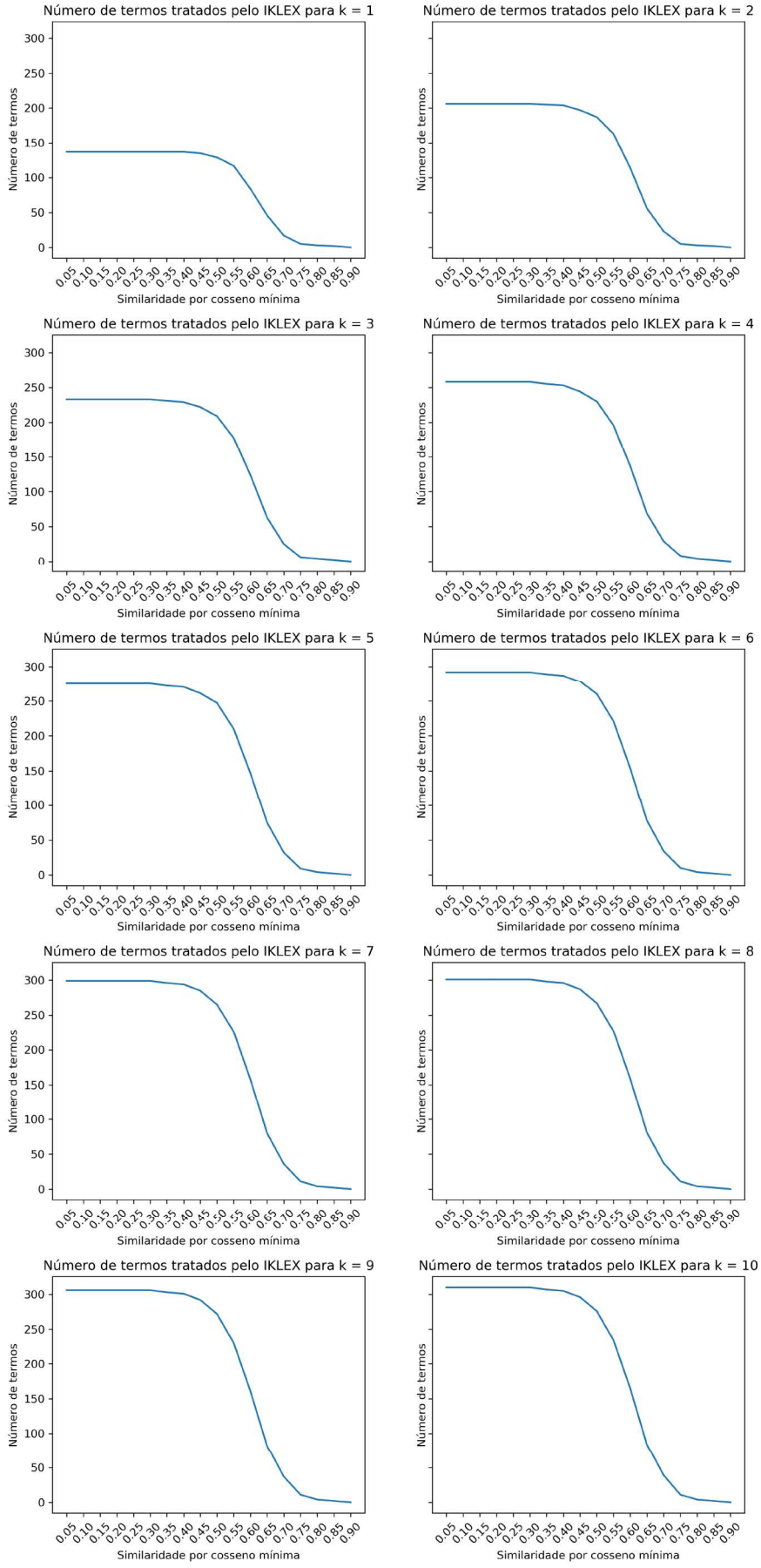

Figura 18 – Número de termos tratados pelo IKLex 2 no conjunto de dados KANSAON

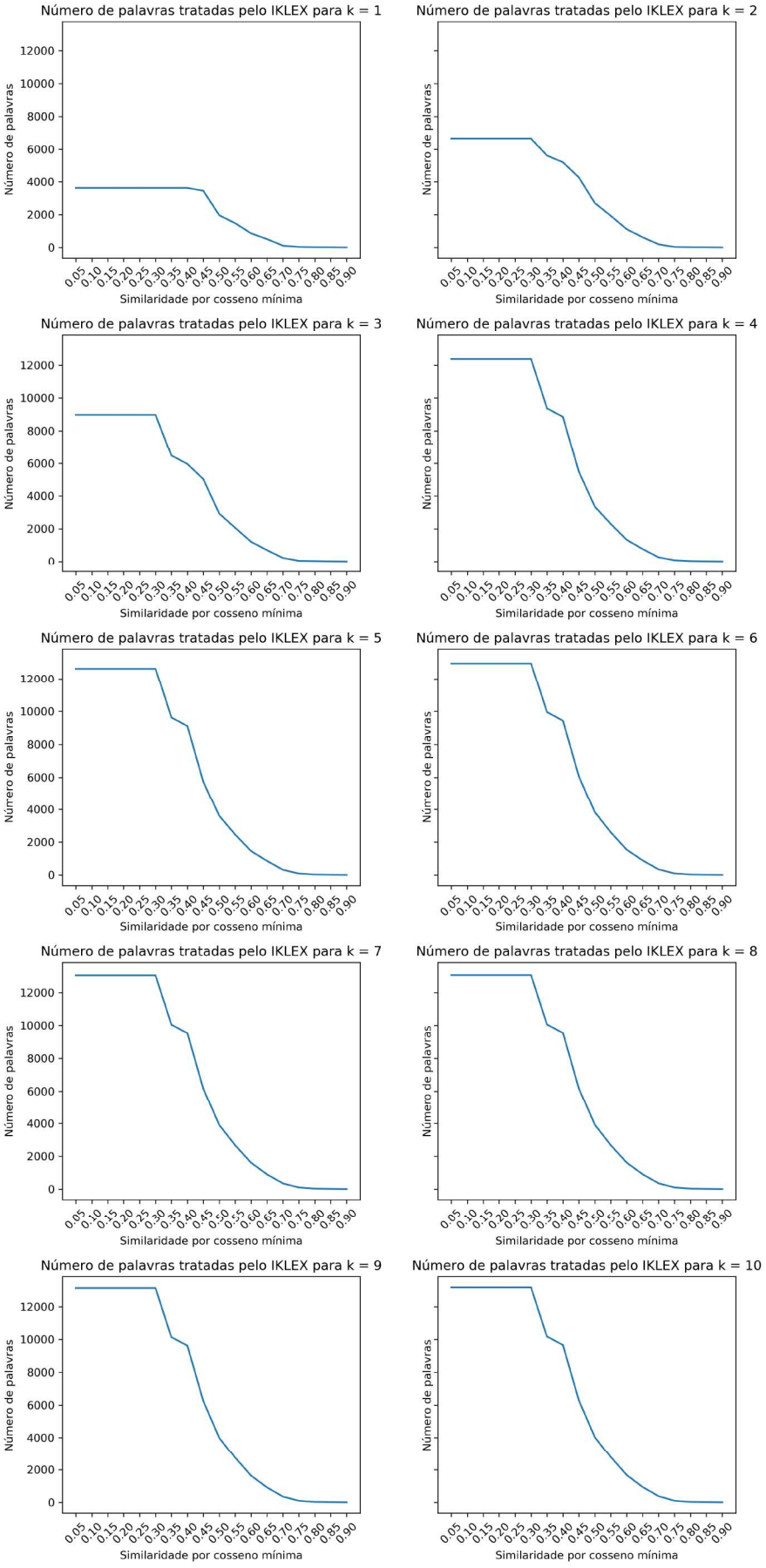

Figura 19 – Número de palavras tratadas pelo IKLex 2 no conjunto de dados KANSAON

4.5- Classificadores

Foram escolhidos três algoritmos de classificação amplamente utilizados [Feldman, 2013; Gupte et al., 2014; Ali et al., 2012] em tarefas de AS, sendo eles o *Multinomial Naive Bayes* (MNB), *Linear Support Vector Classifier* (LSVC) e *Random Forests* (RF). Após preparar os conjuntos de dados para a classificação é necessário definir alguns aspectos relacionados a forma de treinamento e parâmetros dos classificadores MNB, RF e LSVC. Para treinar os classificadores foi escolhida a biblioteca scikit-learn[5] da linguagem Python. Todos os classificadores foram treinados utilizando o método de validação cruzada de *10-folds*, conforme feito por Mullen and Collier [2004] e Wilson et al. [2005] e outros trabalhos na literatura.

O classificador MNB foi inicializado com suavização de Lidstone, com o valor de 1×10^{-6}, conforme adotado por Melville et al. [2009]. A suavização de Lidstone foi adotada pois tende a ter um desempenho superior em tarefas de classificação de textos em relação a suavização de Laplace [Agrawal et al., 2000]. O RF foi treinado com 64 árvores, levando em consideração o trabalho de Oshiro et al. [2012], no qual 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 1.6.1, 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 categorias é menor que o conjunto de treinamento. Demais hiperparâmetros do LSVC foram definidos como os padrões para o algoritmo nesta biblioteca.

Após o treinamento dos classificadores, é preciso definir uma métrica para avaliação da qualidade dos classificadores em relação ao teste. A métrica escolhida para a avaliação dos classificadores é o *F1 score* apresentada na seção 1.7. Essa métrica é habitualmente utilizada em aprendizado de máquina [Forman, 2003] e é um aspecto importante quando se avalia o desempenho de um sistema de Processamento de Linguagem Natural (NLP) ou um sistema de Recuperação de Informação (IR) [Huang et al., 2015]. Por se tratar da utilização do método *10-fold cross validation* para treinamento dos classificadores, é realizada a média dos *F1 scores* de cada teste realizado na validação cruzada.

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

A técnica de validação cruzada é utilizada para treinar diversos modelos com o intuito de avaliar os hiperparâmetros k e de similaridade mínima por cosseno ($simMin$) do algoritmo *IKLex 2*. Para isso, os classificadores MNB, LSVC e RF são treinados para diferentes combinações desses parâmetros. O parâmetro k foi variado no intervalo inclusivo de 1 a 10, enquanto o parâmetro $simMin$ foi variado no intervalo inclusivo de $0,05$ a $0,90$, com saltos de $0,05$. Além disso, também são gerados os modelos que utilizam apenas o LIWC, sem modificações, ou seja, sem o uso do *IKLex 2*.

4.6- Resultados

Esta seção apresenta os resultados obtidos em cada conjunto de dados. Para cada conjunto de dados, são apresentadas as métricas de cada classificador descrito na seção 4.5. As métricas são as médias dos *F1 scores* de cada teste realizado na validação cruzada para cada classificador. Conforme explicado na seção 4.5, foi realizado o treinamento de diversos modelos variando os hiperparâmetros k e $simMin$ do algoritmo *IKLex 2*.

São apresentados alguns gráficos de métricas obtidas com os classificadores treinados. Cada figura nesta subseção possui 10 gráficos. Cada gráfico apresenta os valores referentes aos classificadores treinados com um determinado parâmetro k com o *IKLex 2*, variando de 1 a 10. Para cada gráfico as métricas são dispostas em duas dimensões, variando de acordo com o parâmetro de similaridade por cosseno mínima ($simMin$). Cada gráfico também apresenta o *F1 score* obtido para o classificador treinado com o LIWC.

4.6.1 Conjunto de dados MQD

A figura 20 apresenta os resultados para o classificador MNB. A média de *F1 scores* do classificador MNB treinado com o *IKLex 2* foi até $1,5\%$ superior em relação ao classificador MNB treinado com o LIWC. A melhora dos classificadores treinados com

o *IKLex* é mais destacada quando o parâmetro $simMin$ está entre $0,45$ e $0,30$ e $k > 6$. O melhor resultado é obtido com $k = 9$, $simMin$ entre $0,05$ e $0,20$, e a média de *F1 scores* igual a $71,3\%$. O classificador MNB treinado com o LIWC obteve a média de *F1 scores* igual a $69,8\%$. Existe uma tendência de que quanto maior o k e o quanto menor o parâmetro $simMin$, melhores os resultados.

A figura 21 apresenta os resultados para o classificador LSVC. A média de *F1 scores* do classificador LSVC treinado com o *IKLex 2* foi até $2,7\%$ superior em relação ao classificador LSVC treinado com o LIWC. A melhora dos classificadores treinados com o *IKLex* é mais destacada quando o parâmetro $simMin$ está entre $0,45$ e $0,20$. O melhor resultado é obtido com $k = 9$, $simMin$ entre $0,05$ e $0,20$, e a média de *F1 scores* igual a $70,8\%$. O classificador LSVC treinado com o LIWC obteve a média de *F1 scores* igual a $68,1\%$.

A figura 22 mostra as médias dos *F1 scores* obtidos com o classificador RF. O classificador RF treinado com o léxico do LIWC obteve a média de *F1 scores* igual a $69,7\%$. Os gráficos indicam que os melhores resultados com o classificador treinado com o *IKLex 2* foram obtidos quando o parâmetro $simMin <= 0,40$. Os melhores resultados foram obtidos com $k = 3$ e $simMin$ no intervalo de $0,05$ até $0,2$, havendo uma melhora de até $2,1$ pontos percentuais em relação ao classificador treinado com o léxico do LIWC.

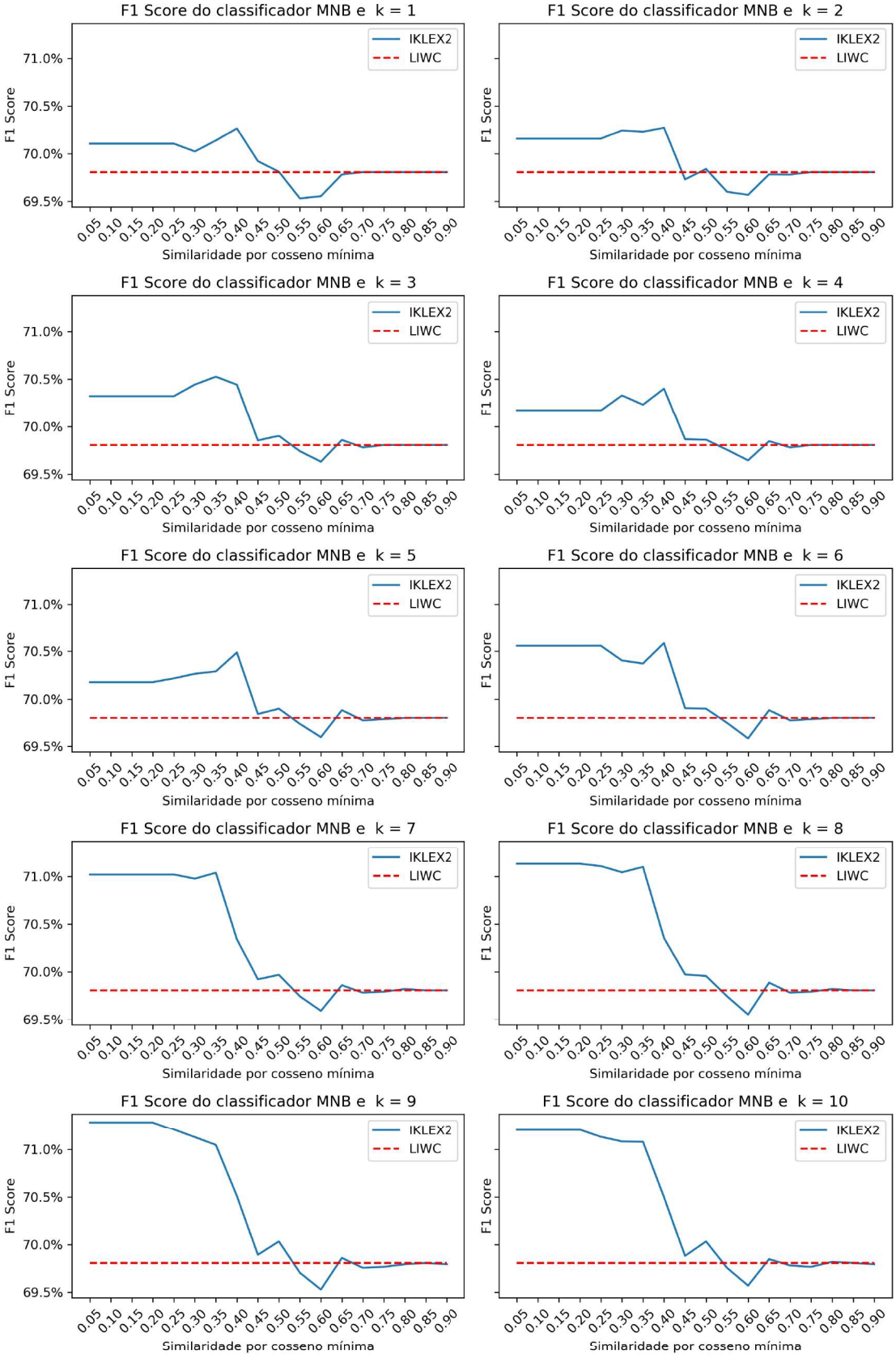

Figura 20 – Gráficos que mostram os F1 scores obtidos para o classificador MNB no conjunto de dados MQD.

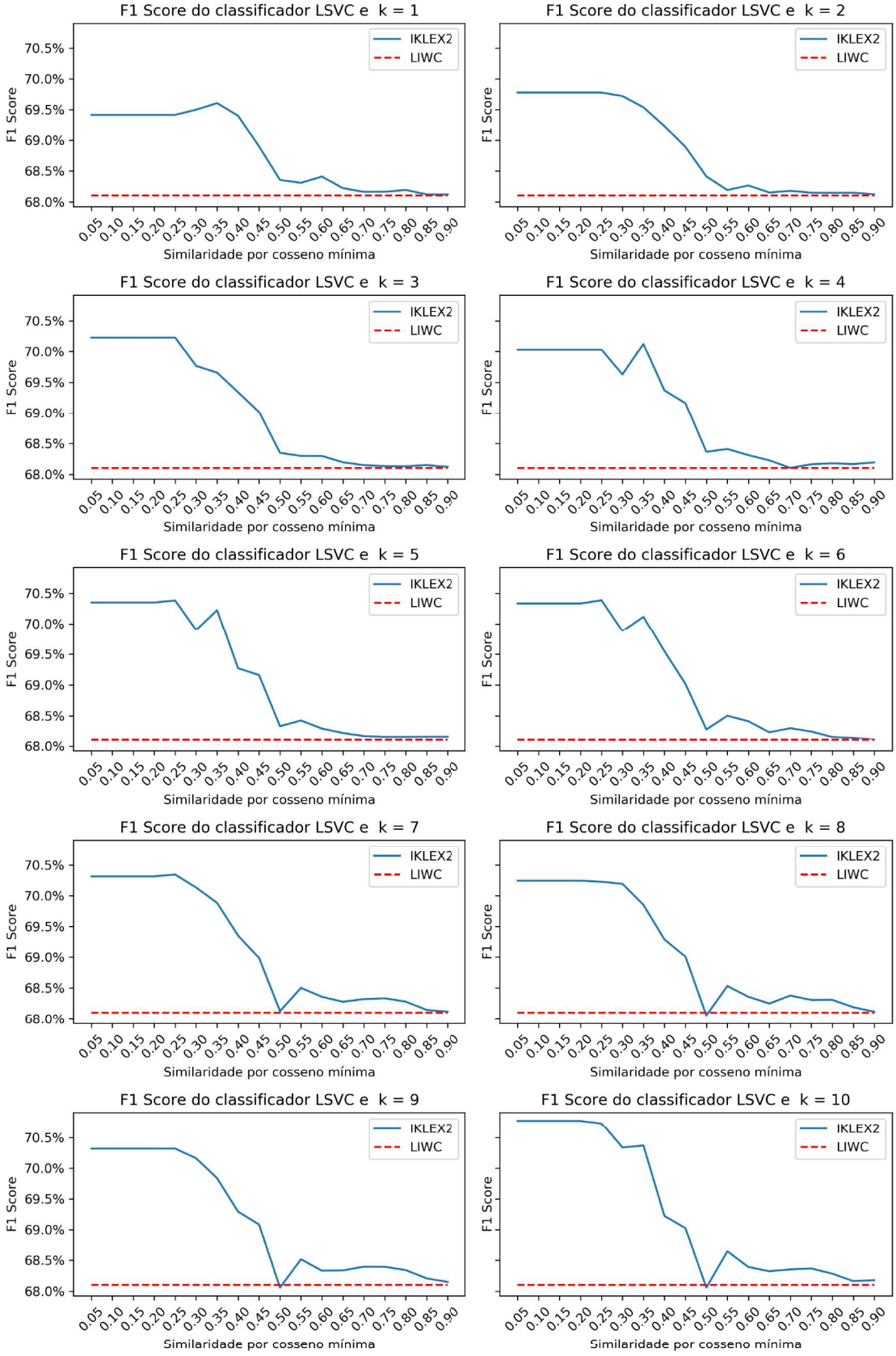

Figura 21 – Gráficos que mostram os F1 scores obtidos para o classificador LSVC no conjunto de dados MQD.

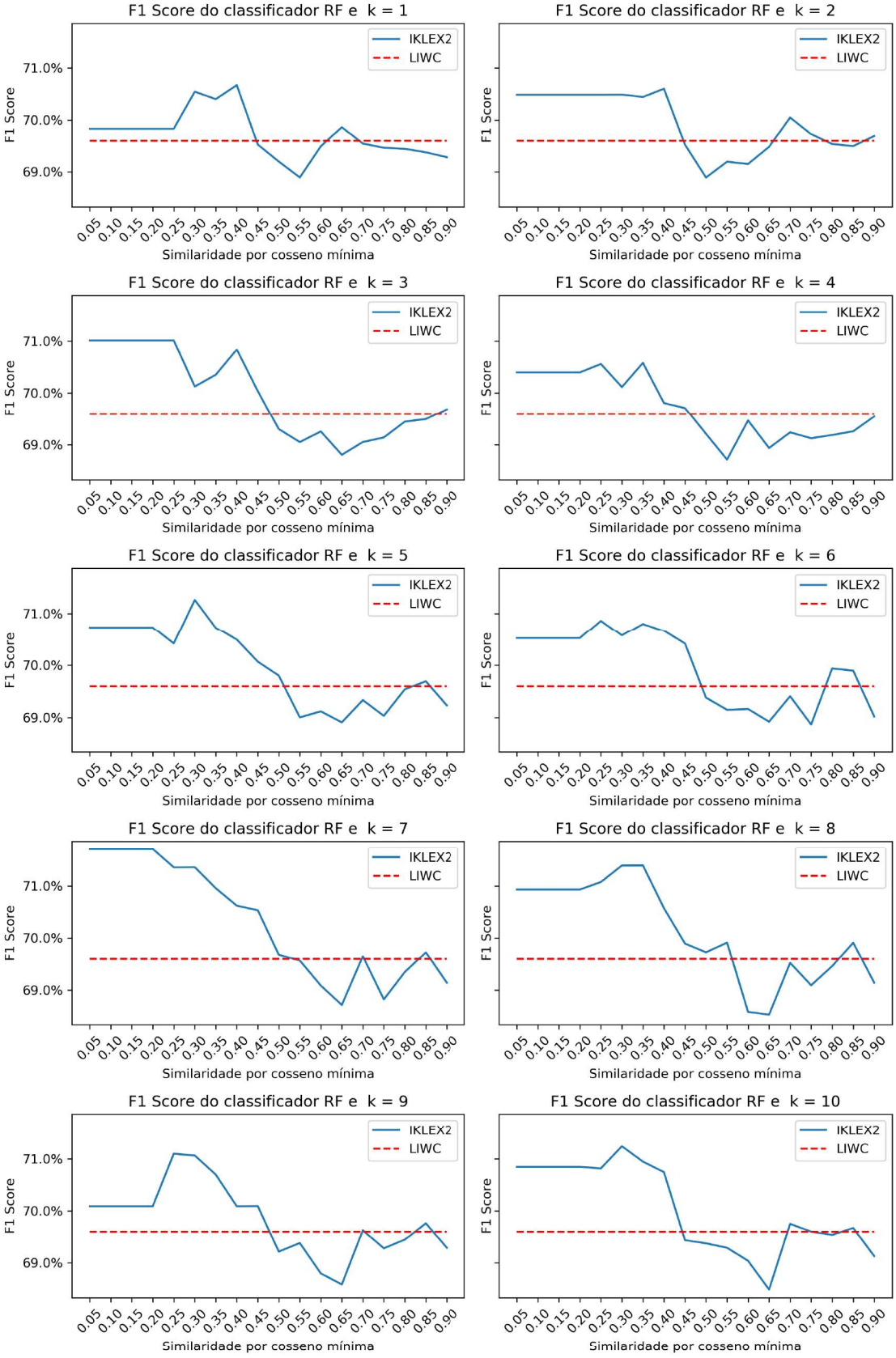

Figura 22 – Gráficos que mostram os F1 scores obtidos para o classificador RF no conjunto de dados MQD.

4.6.2 Conjuntos de dados TAS-PT

A figura 23 apresenta os resultados para o classificador MNB no conjunto de dados TAS-PT. Os gráficos indicam que classificador treinado com o LIWC obteve a média de *F1 scores* igual a $61,3\%$. Os melhores resultados do classificador MNB treinado com o *IKLex 2* são alcançados quando o parâmetro $k > 2$ e $simMin$ se encontra entre $0,05$ e $0,3$. O melhor resultado foi obtido quando $k = 7$ e $simMin$ entre $0,05$ e $0,3$, atingindo a média de $63,6\%$.

A figura 24 apresenta os resultados obtidos com o classificador LSVC no conjunto de dados TAS-PT. O classificador treinado com o LIWC obteve $63,1\%$ de *F1 score*. Os resultados indicam que o classificador LSVC treinado com o *IKLex 2* obteve melhor desempenho quando o parâmetro $simMin$ se encontra no intervalo entre $0,05$ e $0,3$. O classificador LSVC treinado com o *IKLex 2* obteve a média de *F1 scores* igual a $65,4\%$, cerca de $2,3$ pontos percentuais a mais em relação ao classificador LSVC treinado com o LIWC.

A figura 25 apresenta os *F1 scores* obtidos com o classificador RF no conjunto de dados TAS-PT. O classificador RF treinado com o LIWC obteve a média de *F1 scores* igual a $67,5\%$. O classificador RF treinado com o *IKLex 2* obteve melhores resultados quando o parâmetro $simMin$ esteve entre $0,05$ e $0,3$. O melhor resultado foi alcançado quando $k = 4$ e $simMin = 0,2$, obtendo a média de *F1 scores* igual a $69,2\%$, cerca de $1,7$ pontos percentuais a mais em relação ao mesmo classificador treinado com o LIWC.

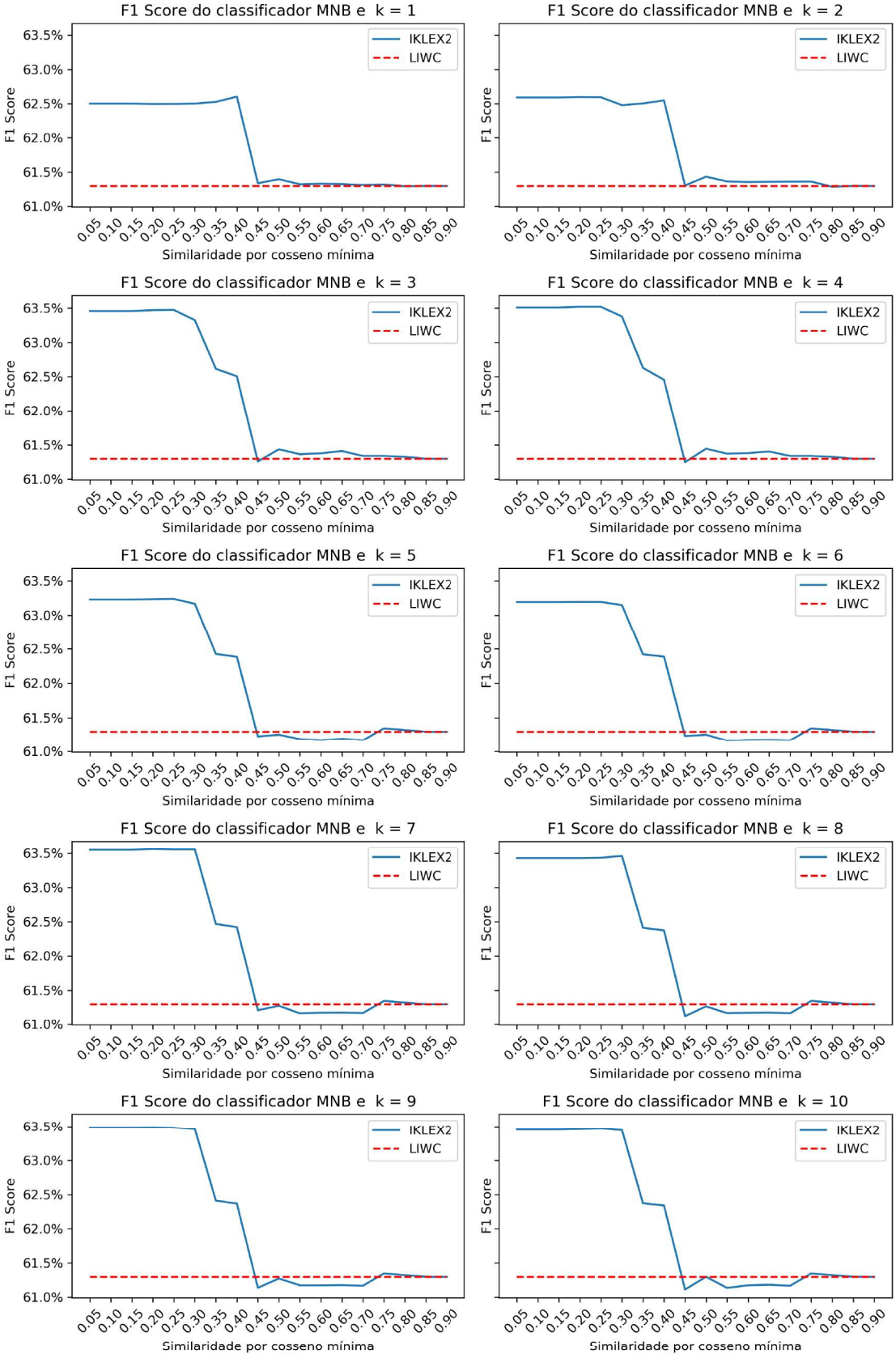

Figura 23 – Gráficos que mostram os F1 scores obtidos com o classificador MNB no conjunto de dados TAS-PT.

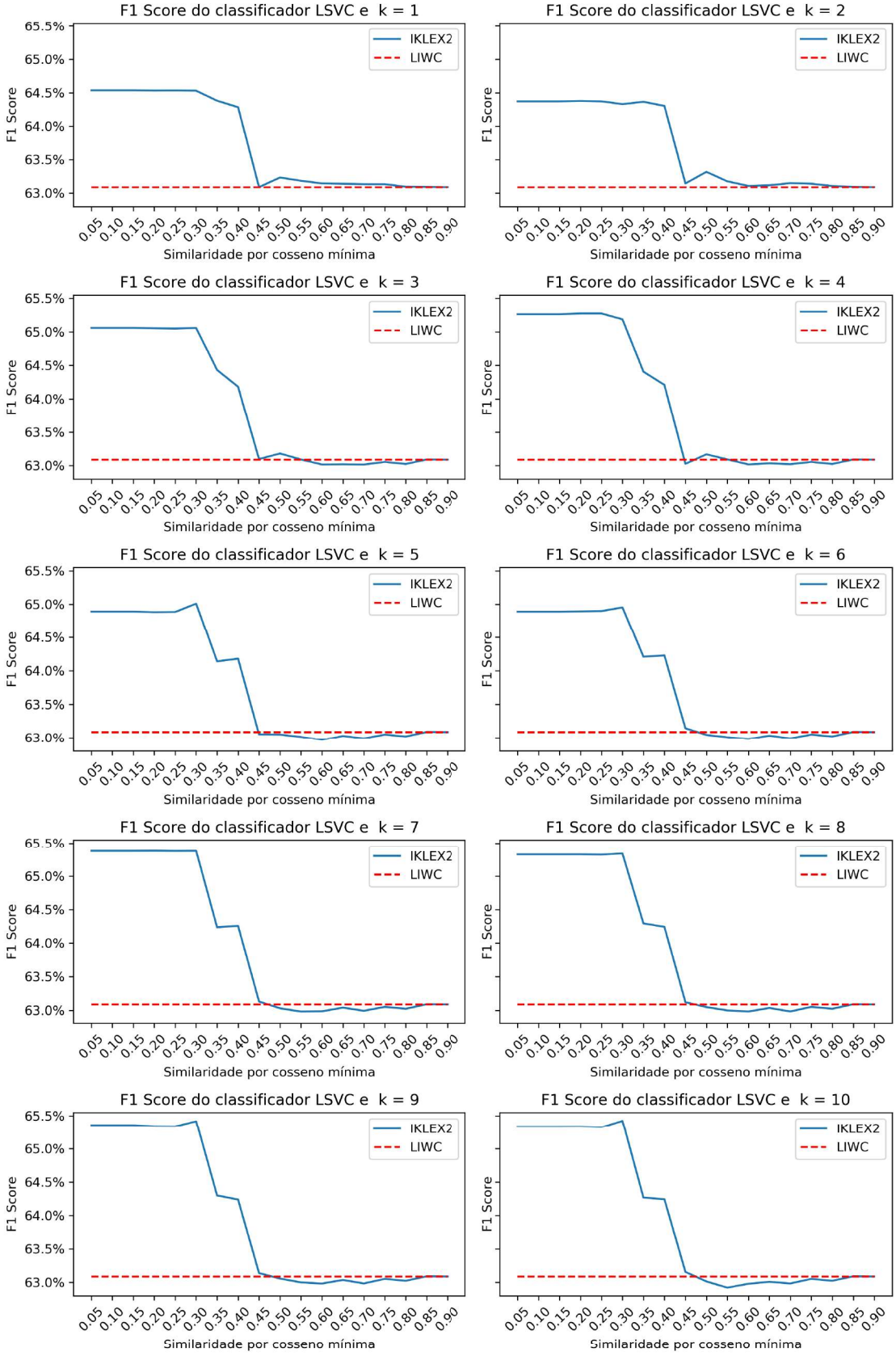

Figura 24 – Gráficos que mostram os F1 scores obtidos com o classificador LSVC no conjunto de dados TAS-PT.

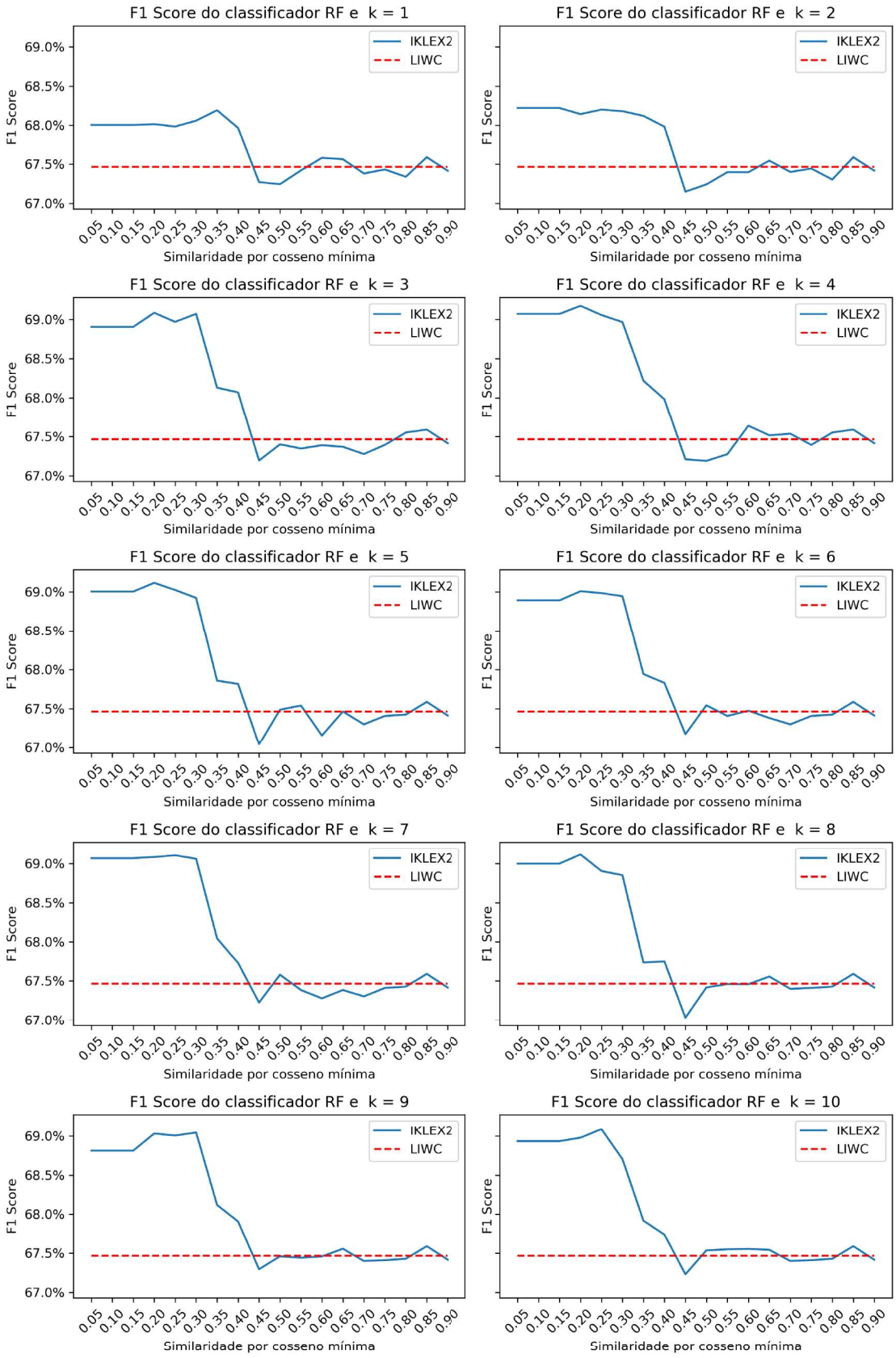

Figura 25 – Gráficos que mostram os F1 scores obtidos com o classificador RF no conjunto de dados TAS-PT.

4.6.3 Conjunto de dados KANSAON

A figura 26 apresenta os *F1 scores* obtidos com o classificador MNB. O classificador MNB treinado com o LIWC obteve a média de *F1 scores* igual a $73,4\%$. O classificador MNB treinado com o *IKLex 2* tende a obter melhores resultados quando o parâmetro $simMin$ se encontra entre $0,05$ e $0,45$. O classificador MNB obteve o melhor resultado quando $k = 5$ e $simMin = 0,35$, obtendo a média de *F1 scores* máxima de $76,7\%$, cerca de $3,3$ pontos percentuais a mais em relação ao classificador MNB treinado com o LIWC.

A figura 27 apresenta os *F1 scores* obtidos com classificador LSVC. O classificador LSVC treinado com o LIWC obteve a média de *F1 scores* igual a $76,6\%$. O classificador LSVC treinado com o *IKLex 2* tende a obter melhores resultados quando o parâmetro $simMin$ se encontra entre $0,05$ e $0,3$, alcançando a média máxima de $78,7\%$, quando $k = 10$, cerca de $2,1$ pontos percentuais a mais em relação ao classificador LSVC treinado com o LIWC.

A figura 28 apresenta os *F1 scores* obtidos com o classificador RF. O classificador RF treinado com o LIWC obteve a média de *F1 scores* igual a $78,1\%$. O classificador RF treinado com o *IKLex 2* tende a obter melhores resultados quando o parâmetro $simMin$ se encontra entre $0,05$ e $0,3$, alcançando a média máxima de 80% quando $k = 5$, cerca de $1,9$ pontos percentuais a mais em relação ao classificador RF treinado com o LIWC.

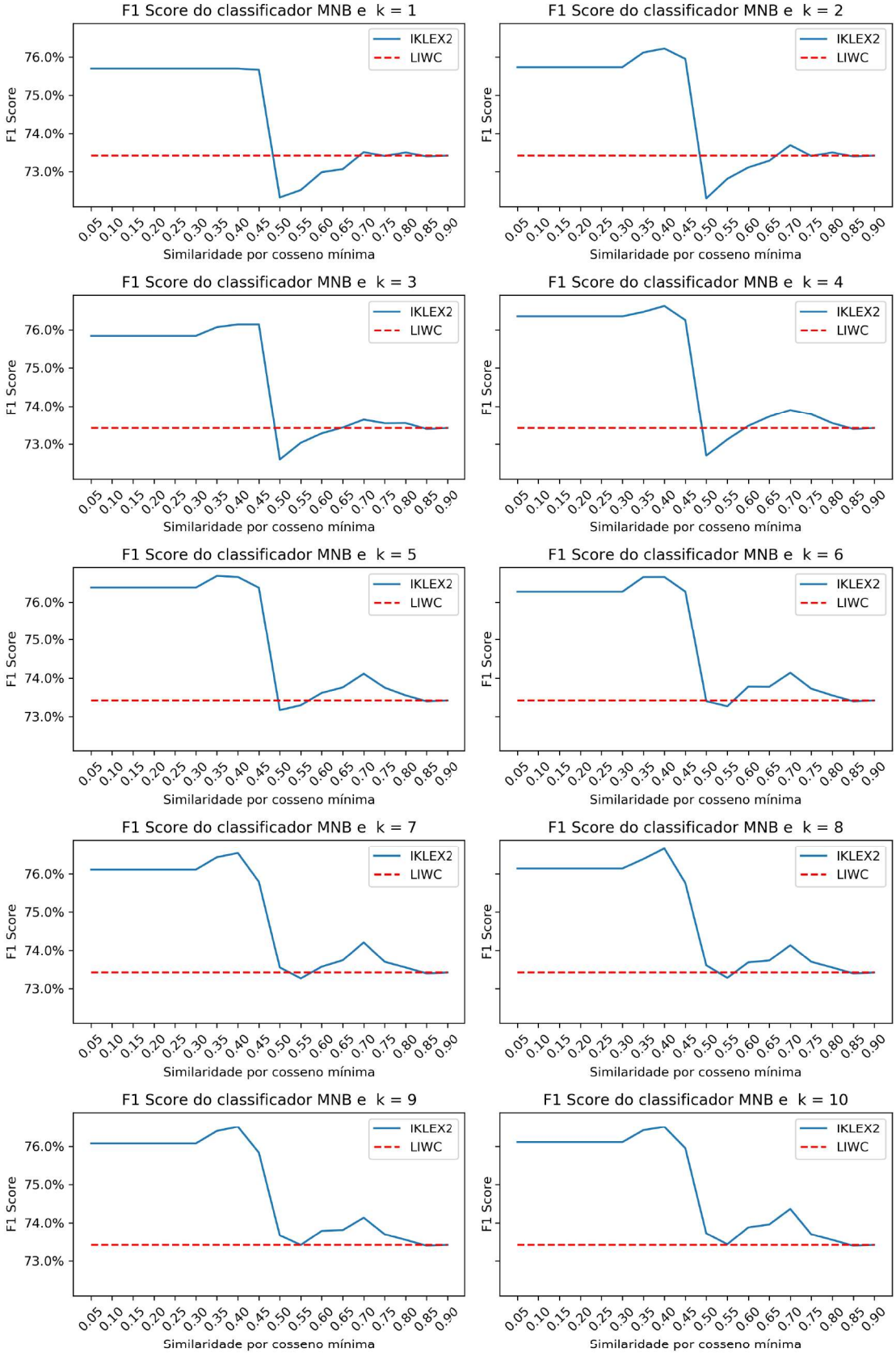

Figura 26 – Gráficos que mostram os F1 scores obtidos com o classificador MNB no conjunto de dados KANSAON.

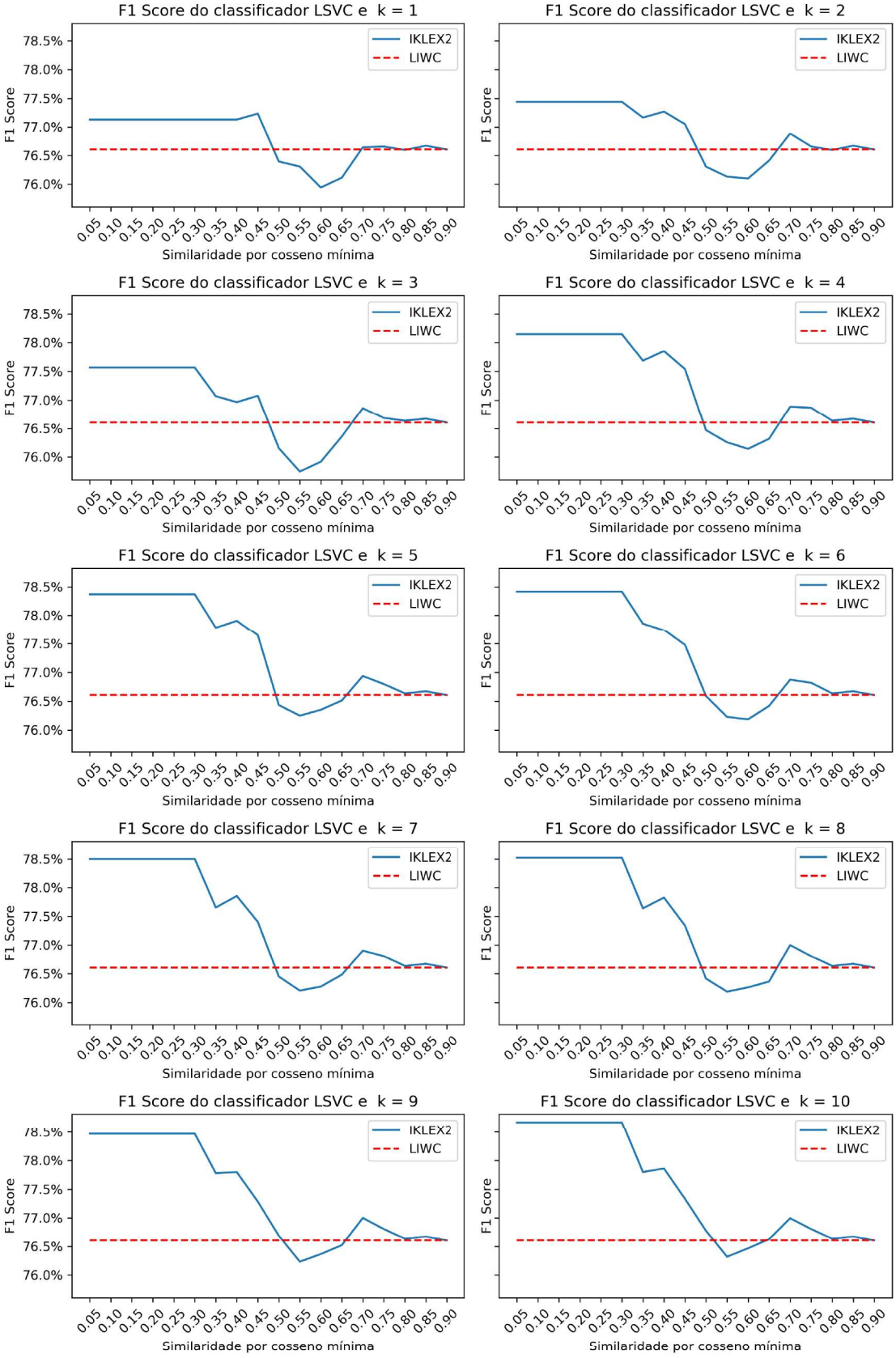

Figura 27 – Gráficos que mostram os F1 scores obtidos com o classificador LSVC no conjunto de dados KANSAON.

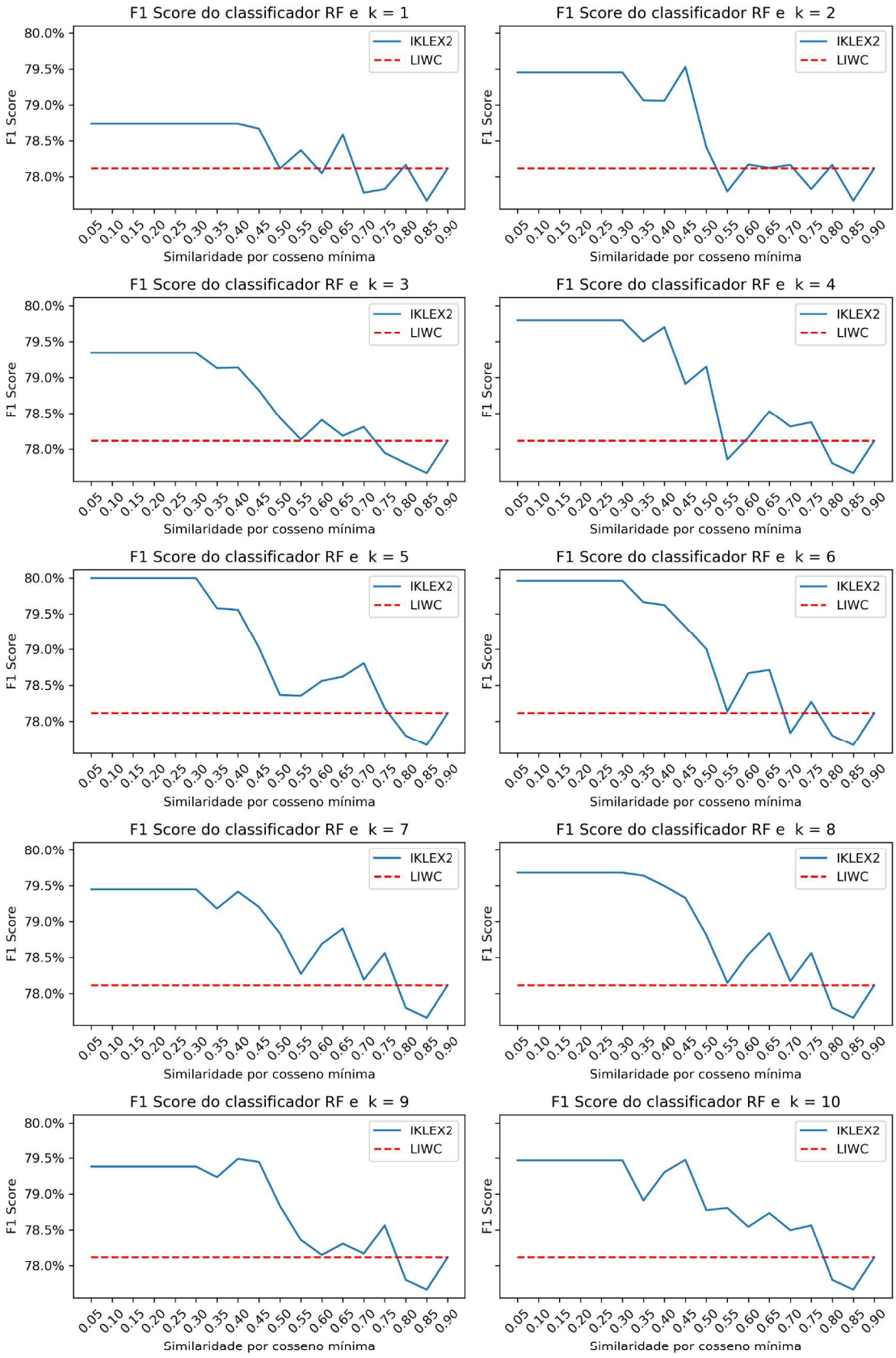

Figura 28 – Gráficos que mostram os F1 scores obtidos com o classificador RF no conjunto de dados KANSAON.

4.7- Teste de Hipótese

A seção anterior apresentou os resultados das médias de *F1 scores* obtidas para classificador treinado utilizando o método de validação cruzada de *10-folds*. Para cada um dos *10-folds* o conjunto de dados é dividido aleatoriamente em 10 subconjuntos independentes. Desses subconjuntos, 9 desses subconjuntos são utilizados para treinamento e apenas 1 é utilizado para teste ou validação. Esse processo é descrito com maiores detalhes na seção 1.8.

Como cada *fold* treina um modelo independente, é possível extrair métricas, como o *F1 score*. Assim, a estimativa de generalização do modelo é feita obtendo a média de cada métrica para todos os *folds*. No entanto, obter as médias sem considerar a variância não é suficiente para afirmar que determinada média resultante de uma validação cruzada é diferente da média obtida por outros modelos de validação cruzada.

Desse modo, é necessário descobrir se a diferença das médias de *F1 scores* obtidas pelos classificadores treinados com o LIWC e com o *IKLex 2* são estatisticamente significantes. Logo, ao realizar um teste de hipótese é possível responder se o uso do *IKLex 2* melhora a classificação de polaridade. Para realizar esse teste de hipótese foi selecionado o conjunto de dados, o parâmetro $simMin$ e k e o classificador que o *IKLex 2* obteve melhor desempenho em relação ao classificador treinado com o léxico do LIWC. Nesse cenário, para comprovar a hipótese geral do trabalho, estabelecem-se as hipóteses nula (H_θ) e alternativa (H_1) abaixo:

- H_θ, não há diferença entre os léxicos produzidos pelo algoritmo *IKLex 2* e o LIWC para a classificação de polaridade.
- H_1, os léxicos produzidos pelo *IKLex 2* melhoram a tarefa de classificação de polaridade.

Em síntese aos diversos experimentos realizados, a tabela 4 mostra os melhores *F1 scores* alcançados com os classificadores treinados utilizando o léxico expandido pelo *IKLex 2*. A tabela também mostra os *F1 scores* alcançados com os classificadores treinados com o LIWC. Os resultados são apresentados para cada conjunto de dados. Em todos os classificadores e conjuntos de dados o léxico do *IKLex 2* desempenhou melhor que o LIWC.

Tabela 4 – Tabela contendo os F1 scores dos melhores resultados do IKLex e dos resultados alcançados com o LIWC para cada classificador. A tabela também contém a melhora em pontos percentuais (p.p) alcançada pelo IKLex 2 em relação ao LIWC.

Conjunto de dados	MNB			LSVC			RF		
	IKLEX	LIWC	Melhora	IKLEX	LIWC	Melhora	IKLEX	LIWC	Melhora
MQD	71,3%	69,8%	1,5 p.p	70,8%	68,1%	2,7 p.p	71,8%	69,7%	2,1 p.p
TAS-PT	63,6%	61,3%	2,3 p.p	65,4%	63,1%	2,3 p.p	69,2%	67,5%	1,7 p.p
KANSAON	76,7%	73,4%	**3,3 p.p**	78,7%	76,6%	2,1 p.p	80%	78,1%	1,9 p.p

Conforme destaque na tabela 4, a maior diferença se encontra no classificador MNB, no conjunto de dados KANSAON, treinado com o *IKLex 2*. Esse classificador obteve a média de *F1 scores* com $3,3$ pontos percentuais a mais em relação ao mesmo classificador treinado com o LIWC. Os melhores parâmetros para o *IKLex 2* nesse classificador foram $k = 5$ e $simMin = 0,35$. Logo, o teste de hipótese a ser realizado levará em conta esses parâmetros k e $simMin$. Essa comparação deve ser com o mesmo conjunto de dados. O teste de hipótese deve comparar o classificador MNB treinado com o léxico gerado pelo *IKLex 2* e o classificador MNB treinado com o léxico do *LIWC*.

O teste de hipótese se deu da forma a seguir: os classificadores definidos para comparação foram treinados novamente, utilizando o método de validação cruzada de *10-folds*. Esse método de validação cruzada é repetido 30 vezes, obtendo ao todo 300 *F1 scores* para cada modelo. Por conseguinte, é realizado o teste paramétrico de *Wilcoxon signed rank test*, conforme recomendado por Demšar [2006]. O intervalo de confiança para este experimento é de 95%.

A média obtida para os modelos treinados com o léxico do *IKLex 2* foi de $76,69\%$ e a média obtida para os modelos treinados com o LIWC foi $73,28\%$. O *p-valor* obtido foi de aproximadamente 1.7344×10^{-6}, descartando a hipótese nula. Portanto, é possível afirmar que o uso do *IKLex 2* impacta diretamente na qualidade da classificação de polaridade.

4.8- Discussão

Foram realizados experimentos em três conjuntos de dados em Português do Brasil. Para cada conjunto de dados foram utilizados três algoritmos de classificação distintos. O objetivo é observar se ao utilizar o *IKLex 2* para tratar palavras fora do

vocabulário e expandir um léxico existente há uma melhora na classificação de polaridade. Os resultados indicam que o *IKLex 2* melhora a classificação de polaridade.

Para descobrir se há essa melhora em relação ao léxico original, foram treinados diversos classificadores. Alguns classificadores foram treinados com o léxico do LIWC e foram comparados com os classificadores treinados com o léxico expandido pelo *IKLex 2*, também derivado do LIWC. Em relação aos classificadores treinados com os léxicos gerados pelo algoritmo *IKLex 2* foi preciso treinar um para cada variação de parâmetro.

O *IKLex 2* possui dois parâmetros: k e $simMin$. O parâmetro k diz respeito ao número de vizinhos próximos obtidos e o parâmetro $simMin$ diz respeito a similaridade por cosseno mínima que um vizinho precisa ter em relação a palavra fora do vocabulário. Essa combinação de parâmetros resultou em diversos experimentos.

Em relação ao parâmetro k, mesmo quando de $k = 1$ ou $k = 2$ é possível observar melhora nos resultados. No entanto, é recomendável que k não seja tão pequeno, pois o número de palavras tratadas pelo *IKLex 2* pode ser ínfimo. Todavia, quando $k >= 5$, observa-se que quanto maior o parâmetro k, menor o impacto deste nos resultados. Isso também tem relação com o número de palavras tratadas pelo algoritmo (quanto maior o número de vizinhos próximos, menos palavras são introduzidas ao novo léxico, quando $k >= 5$). Como raramente há o impacto negativo de k para valores altos, este trabalho recomenda o uso de $k >= 5$.

Em relação a similaridade por cosseno mínima ($simMin$), os melhores resultados são observados quando o parâmetro se encontra em valores pequenos, geralmente entre $0,05$ e $0,30$. Isso indica que o parâmetro $simMin$ em valores altos pode limitar a melhora nos classificadores, ao mesmo tempo que impede que palavras dissimilares sejam adicionadas ao léxico.

Por fim, o *IKLex 2* melhora a qualidade do léxico original ao introduzir novas palavras que não existiam. Em todos os três algoritmos de classificação elegidos foram observadas melhoras. Essa melhora foi observada em três conjuntos de dados distintos.

Considerações finais

A expansão do uso da internet levou ao surgimento de serviços orientados a opinião, como as redes sociais e sites de avaliações. Nesses serviços, usuários expressam seus sentimentos e expõem suas opiniões sobre os mais diversos assuntos. Assim, surgiu uma grande quantidade de dados para análise.

Esses dados despertaram o interesse de pesquisadores da área de AS. Esses pesquisadores desenvolveram abordagens diferentes para realizar tarefas de AS: a abordagem de aprendizado de máquina, a abordagem baseada em léxico e a abordagem híbrida. A abordagem baseada em léxicos e a abordagem híbrida enfrentam um problema de linguagem natural em comum, que são as palavras fora do vocabulário, especialmente quando estão envolvidos textos provenientes de redes sociais.

Essa dissertação discorreu sobre o problema de palavras fora do vocabulário em léxicos utilizados em tarefas de AS. Ainda que métodos baseados em léxicos geralmente tenham desempenho inferior em relação aos métodos baseados em aprendizado de máquina, os métodos baseados em léxicos continuam sendo competitivos, pois os métodos baseados em aprendizado de máquina supervisionados necessitam de amostras manualmente anotadas que nem sempre estão disponíveis [Hailong et al., 2014].

Na literatura estudada, autores tratam o problema de palavras fora do vocabulário utilizando as seguintes soluções: Normalização textual; expansão de um léxico já existente; e por meio da criação de um classificador. Este trabalho se encontra na categoria que expande um léxico já existente. No entanto, essa expansão não é manual.

A proposta deste trabalho baseou-se na hipótese distribucional de Harris [1954]. Essa hipótese se baseia na ideia de que palavras que ocorrem no mesmo contexto tendem a ter o mesmo significado [Harris, 1954]. Portanto, uma palavra fora do vocabulário pode ser substituída por outra palavra que esteja contida no léxico, caso essa palavra seja mais similar ou tenha contextos parecidos.

Os *word embeddings* também se baseiam na hipótese distribucional, pois utilizam o contexto das palavras para construir o espaço vetorial. Por isso, nessa dissertação foi utilizado o *Word2Vec* [Mikolov et al., 2013a] para gerar *word embeddings* com o fim de utilizar esses vetores. O espaço vetorial gerado pelo *Word2Vec* possui sentido semântico

[Mikolov et al., 2013a], sendo possível realizar operações algébricas entre os vetores e medir a similaridade entre eles.

O objetivo desta dissertação é fornecer uma forma de tratar palavras fora do vocabulário ao utilizar métodos de AS baseados em léxicos e métodos híbridos, no nível de documentos e sentenças. Dessa forma, foi elaborado um algoritmo denominado *IKLex 2*, que utiliza *word embeddings* para substituir uma palavra fora do vocabulário por outra palavra semanticamente mais próxima. O algoritmo recebe como entrada um léxico já existente e expande este léxico no fim do processo.

Experimentos e Resultados

Os experimentos neste trabalho consistem em realizar comparações de léxicos gerados pelo algoritmo *IKLex 2* e o léxico original utilizado como entrada para o algoritmo. Para avaliar os novos léxicos gerados pelo *IKLex 2* foi escolhida uma abordagem de AS híbrida. Por se tratar de uma abordagem híbrida, a qualidade dos resultados foi avaliada de acordo com o desempenho de cada classificador treinado utilizando um léxico. O desempenho de cada classificador é medido por meio da métrica *F1 score*, apresentada na seção 1.7.

Foram realizados diversos experimentos em três conjuntos de dados em Português do Brasil. Para os experimentos foram utilizados três classificadores, o MNB, LSVC e o RF. Os classificadores foram treinados utilizando o método de validação cruzada *k-fold*. Os experimentos foram divididos em dois fluxos de trabalho distintos, conforme explicado na seção 4.1. O fluxo a ser executado depende se o experimento em questão utiliza o léxico original ou se há a execução do *IKLex 2* para expandir o léxico. O léxico escolhido para os experimentos é o LIWC na versão 2007 para Português do Brasil.

Os experimentos foram agrupados de acordo com o conjunto de dados e classificador utilizado. Para cada conjunto de dados, foram executados três experimentos que utilizaram o LIWC, sendo um para cada classificador. O *IKLex 2* foi executado diversas vezes, variando os parâmetros k e $simMin$, gerando um léxico diferente em cada execução. Cada léxico gerado pelo *IKLex 2* foi utilizado para o treinamento em cada um dos três classificadores. Assim, todos os classificadores treinados com o LIWC, em um

determinado conjunto de dados, foram comparados com os classificadores treinados com os léxicos gerados pelo *IKLex 2*.

Os resultados indicam que o algoritmo proposto melhora em até $3,3$ pontos percentuais as médias de *F1 score* obtidas em uma classificação de polaridade. Todos os experimentos realizados em todos os conjuntos de dados apresentaram resultados positivos. Isso indica que o objetivo proposto por essa dissertação foi alcançado: tratar palavras fora do vocabulário com o algoritmo *IKLex 2* realmente melhora a classificação de polaridade. Essa hipótese foi comprovada por um teste de hipótese não-paramétrico sobre o melhor resultado.

Limitações do estudo e trabalhos futuros

Ao longo do desenvolvimento da pesquisa foi notada grande dificuldade em se obter conjuntos de dados em Português do Brasil. Isso se deve não apenas ao limitado estudo de AS restrito a esse idioma, como também a questões de privacidade. No caso do *Twitter*, a atual política de desenvolvedores[6] só permite que sejam compartilhados os *Tweets IDs*. Isso significa que muitos *Tweets* podem não existir mais quando forem coletados pela API fornecida.

Outra limitação desse estudo foi que, no início da pesquisa existia apenas a versão 2007 do LIWC para Português do Brasil como a mais atualizada. Como a língua evoluiu nesses anos, inclusive com um novo acordo ortográfico[7], o vocabulário desse léxico já está defasado. Durante o desenvolvimento desta pesquisa, foi desenvolvida uma nova versão do LIWC para Português do Brasil por Carvalho et al. [2019]. Em trabalhos futuros devem ser realizados experimentos com essa nova versão do léxico.

Nos experimentos realizados, o *IKLex 2* utiliza *word embeddings* baseados no mesmo conjunto de dados em que é realizada a classificação de polaridade. Provavelmente as relações semânticas entre as palavras diferem entre conjuntos de dados distintos. Devem ser realizados experimentos com *word embeddings* treinados com outros conjuntos de dados para verificar se os léxicos gerados pelo *IKLex 2* desempenham da mesma forma que os léxicos gerados pelos experimentos desta dissertação.

[6] https://developer.twitter.com/en/developer-terms/policy

[7] http://www.planalto.gov.br/ccivil$_03/_{Ato}2007-2010/2008/Decreto/D6583.htm$

Esta dissertação também não realizou experimentos com outros idiomas além do Português Brasileiro. Além disso, este estudo se limitou ao uso de apenas um léxico, o LIWC 2007 em Português do Brasil. É fundamental que em trabalhos futuros se façam experimentos com outros léxicos e outros idiomas.

Existem também outras redes neurais para gerar *word embeddings*, como o *GloVe* [Pennington et al., 2014] e o *FastText* [Bojanowski et al., 2016]. É preciso realizar um estudo mais detalhado para descobrir com quais dessas redes neurais o *IKLex 2* obtém melhor desempenho. Dessa forma, é necessário realizar diversas comparações entre elas, com diferentes hiperparâmetros.

Em trabalhos futuros, também devem ser explorados os casos em que as palavras estão ausentes nos dicionários dos modelos de *word embeddings*. Esse problema se torna uma realidade quando é utilizado um modelo de *word embeddings* treinado em outro conjunto de dados. Nesses casos, os *n-gramas* presentes no *FastText* [Bojanowski et al., 2016] podem fazer a diferença para melhores resultados.

Referências Bibliográficas

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (2008). Finding high-quality content in social media. In *Proceedings of the 2008 international conference on web search and data mining*, pages 183–194, New York, NY, USA. ACM.

Agrawal, R., Bayardo, R., and Srikant, R. (2000). Athena: Mining-based interactive management of text databases. In *International Conference on Extending Database Technology*, pages 365–379. Springer.

Ali, J., Khan, R., Ahmad, N., and Maqsood, I. (2012). Random forests and decision trees. *IJCSI International Journal of Computer Science Issues*, 9(5):272–278.

Araújo, M., Gonçalves, P., Benevenuto, F., and Cha, M. (2013). Métodos para análise de sentimentos no twitter. In *Proceedings of the 19th Brazilian symposium on Multimedia and the Web (WebMedia'13)*, Salvador, Brazil. ACM.

Athiwaratkun, B. and Wilson, A. G. (2017). Multimodal word distributions. *arXiv preprint arXiv:1704.08424*.

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*, Fortaleza, Ceará. Brazilian Computer Society.

Bekkerman, R. and Allan, J. (2004). Using bigrams in text categorization. Technical report, Technical Report IR-408, Center of Intelligent Information Retrieval, UMass ....

Bernard, S., Heutte, L., and Adam, S. (2010). A study of strength and correlation in random forests. In *International Conference on Intelligent Computing*, pages 186–191, Changsha, China. Springer.

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

Bloomfield, L. (1926). A set of postulates for the science of language. *Language*, 2(3):153–164.

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

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

Camacho-Collados, J. and Pilehvar, T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *arXiv preprint arXiv:1805.04032*.

Carvalho, F., Rodrigues, R. G., dos Santos, G., Cruz, P., Ferrari, L., and Guedes, G. P. (2019). Evaluating the 2015 brazilian portuguese liwc lexicon with sentiment analysis in social networks. In *CSBC 2019 - 8º BraSNAM ()*, Belém, Brazil.

Cavalcante, P. E. C. (2017). Um dataset para análise de sentimentos na língua portuguesa.

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

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 231–240, New York, NY, USA. ACM.

Dong, Z. and Dong, Q. (2003). Hownet-a hybrid language and knowledge resource. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 820–824, Beijing, China. IEEE.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Farra, N., Challita, E., Assi, R. A., and Hajj, H. (2010). Sentence-level and document-level sentiment mining for arabic texts. In *2010 IEEE international conference on data mining workshops*, pages 1114–1119, Sydney, Australia. IEEE.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

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

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

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, Denver, Colorado. Association for Computational Linguistics, Association for Computational Linguistics.

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, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gupte, A., Joshi, S., Gadgul, P., Kadam, A., and Gupte, A. (2014). Comparative study of classification algorithms used in sentiment analysis. *International Journal of Computer Science and Information Technologies*, 5(5):6261–6264.

Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.

Hailong, Z., Wenyan, G., and Bo, J. (2014). Machine learning and lexicon based methods for sentiment classification: A survey. In *Web Information System and Application Conference (WISA), 2014 11th*, pages 262–265, St. Petersburg, Russia. IEEE.

Han, B., Cook, P., and Baldwin, T. (2013). Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

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.*, Reykjavik, Iceland. European Language Resources Association-ELRA.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, A. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand.

Huang, H., Xu, H., Wang, X., and Silamu, W. (2015). Maximum f1-score discriminative training criterion for automatic mispronunciation detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):787–797.

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, Baltimore, Maryland. Association for Computational Linguistics.

Kansaon, D. P., Brandão, M. A., and de Paula Pinto, S. A. (2018). Análise de sentimentos em tweets em português brasileiro. In *7º Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2018)*, volume 7, Belém, Pará. SBC.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, Montreal, Quebec. American Association for Artificial Intelligence.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, Lille, France.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308, Baltimore, Maryland. Association for Computational Linguistics.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–50, New York, NY, USA. ACM, ACM. ACM Order No.: 606920.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, Berlin, Heidelberg. Springer, Springer Berlin Heidelberg.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

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, Jeju Island, Korea. Association for Computational Linguistics.

Lopes, E. D. (2015). *Utilização do modelo skip-gram para representação distribuída de palavras no projeto Media Cloud Brasil*. PhD thesis.

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, Stroudsburg, PA, USA. Association for Computational Linguistics, 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, New York, NY. ACM.

Malcolm, N. (1954). Wittgenstein's philosophical investigations. *The Philosophical Review*, 63(4):530–559.

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, Madison, Wisconsin. Citeseer.

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

Melville, P., Gryc, W., and Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284, New York, NY, USA. ACM, ACM. 618092.

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, San Diego, California. Association for Computational Linguistics.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, Nevada.

Mnih, A. and Hinton, G. E. (2009). A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, Vancouver, B.C., Canada.

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*, Barcelona, Spain. Association for Computational Linguistics.

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, São Leopoldo, Rio Grande do Sul.

Nascimento, G., Duarte, F., and Guedes, G. P. (2018a). Emoções em português do brasil: um conjunto de dados e resultados de base. In *7º Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2018)*, volume 7, Belém, Pará. SBC.

Nascimento, G., Duarte, F., and Guedes, G. P. (2018b). Handling out-of-vocabulary words in lexicons to polarity classification. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, page 47, Belém, Pará. ACM.

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.

Noferesti, S. and Shamsfard, M. (2015). Using linked data for polarity classification of patients' experiences. *Journal of Biomedical Informatics*, 57:6 – 19.

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, West Kurdofan, Sudan. 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 Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 2359–2364, Liverpool, UK. IEEE.

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*, Portorož, Slovenia. European Language Resources Association (ELRA).

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, Doha, Qatar. Association for Computational Linguistics.

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

Reis, J. C., Gonçalves, P., Araújo, M., Pereira, A. C., and Benevenuto, F. (2015). Uma abordagem multilıngue para análise de sentimentos. In *IV Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2015)*, Recife, Pernambuco.

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, San Diego, California. IEEE.

Rodrigues, R. G. a., Gomes, R. R., Rodrigues, K. T., and Guedes, G. P. (2017). Tatmaster: Psycholinguistic divergences in automatically translated texts. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, WebMedia '17, pages 205–208, New York, NY, USA. ACM.

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, Denver, Colorado. Association for Computational Linguistics, Association for Computational Linguistics.

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

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

Saif, H., He, Y., Fernandez, M., and Alani, H. (2016). Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52(1):5–19.

Schwartz, H. A., Park, G., Sap, M., Weingarten, E., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., Berger, J., Seligman, M., et al. (2015). Extracting human temporal

orientation from facebook language. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 409–419, Denver, Colorado.

Siersdorfer, S., Minack, E., Deng, F., and Hare, J. (2010). Analyzing and predicting sentiment of images on the social web. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 715–718, New York, NY, USA. ACM, ACM. 433107.

Stavrianou, A., Andritsos, P., and Nicoloyannis, N. (2007). Overview and semantic issues of text mining. *ACM Sigmod Record*, 36(3):23–34.

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, Reykjavik, Iceland.

Tan, L. K.-W., Na, J.-C., Theng, Y.-L., and Chang, K. (2011). Sentence-level sentiment polarity classification using a linguistic approach. In *International Conference on Asian Digital Libraries*, pages 77–87, Beijing, China. Springer.

Tavares, R. and Guedes, G. P. (2017). Classificação de filmes: uma abordagem utilizando o liwc. In *6º Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2017)*, volume 6, São Paulo, Brazil. Brazilian Computer Society.

Tsytsarau, M. and Palpanas, T. (2012). Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514.

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, Uppsala, Sweden. Association for Computational Linguistics, Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

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

Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., and Tounsi, L. (2014). Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 223–229, Dublin, Ireland. Association for Computational Linguistics.

Wang, H., Wang, Y., Lu, M., and Choe, Y. (2018). English out-of-vocabulary lexical evaluation task. *arXiv preprint arXiv:1804.04242*.

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, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

Wittgenstein, L. (2009). *Philosophical investigations*. John Wiley & Sons.

Zeng, X., Yang, C., Tu, C., Liu, Z., and Sun, M. (2018). Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana.