UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO MARCOS CARVALHO LIMA

UMA AVALIAÇÃO DO DESEMPENHO DE COMBINAÇÕES CLASSIFICADOR-REPRESENTAÇÃO DE TEXTO EM CATEGORIZAÇÃO DE TEXTOS CURTOS

FORTALEZA – CEARÁ

2019

JOÃO MARCOS CARVALHO LIMA

UMA AVALIAÇÃO DO DESEMPENHO DE COMBINAÇÕES CLASSIFICADOR-REPRESENTAÇÃO DE TEXTO EM CATEGORIZAÇÃO DE TEXTOS CURTOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. José Everardo Bessa Maia

FORTALEZA – CEARÁ

2019

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Lima, João Marcos Carvalho.
Uma avaliação do desempenho de combinações classificador-representação de texto em categorização de textos curtos [recurso eletrônico] / João Marcos Carvalho Lima. - 2019.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 85 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2019.
Área de concentração: Ciência da Computação.
Orientação: Prof. Dr. José Everardo Bessa Maia.

1. Representação de texto. 2. Categorização de texto. 3. Modelos Tópicos. 4. Word Embedding. 5. Textos curtos. I. Título.

JOÃO MARCOS CARVALHO LIMA

UMA AVALIAÇÃO DO DESEMPENHO DE COMBINAÇÕES CLASSIFICADOR-REPRESENTAÇÃO DE TEXTO EM CATEGORIZAÇÃO DE TEXTOS CURTOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 28/10/2019.

BANCA EXAMINADORA

Prof. Dr. José Everardo Bessa Maia (Orientador)
Universidade Estadual do Ceará – UECE

Prof. Dr. Gustavo Augusto Lima de Campos
Universidade Estadual do Ceará – UECE

Profa. Dra. Francisca Raquel de Vasconcelos Silveira
Instituto Federal do Ceará – IFCE

À minha mãe Raimunda e ao meu pai Marcos Jones, por serem meu maior alicerce durante essa jornada e por todos os ensinamentos sobre a vida.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. José Everardo Bessa Maia, pela orientação deste trabalho, por cada aula, ensinamento, incentivo, paciência e atenção durante essa jornada. Muito obrigado pela orientação e formação.

Agradeço a minha mãe Raimunda e ao meu pai Marcos Jones, por todo amor, dedicação, apoio incondicional. Agradeço a minha tia/madrinha Maria Aparecida, por está presente em cada momento especial de minha vida e por cada contribuição em minha formação.

Agradeço a minha namorada Ana Clara, não apenas por compreender cada momento longe, mas pelo apoio e por todos esses anos ao seu lado. Obrigado pelo constante incentivo, paciência e amor.

Agradeço a cada Professor do MACC, pelo conhecimento transmitido.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, pela concessão da bolsa de estudo.

A todos obrigado por permitirem que esta dissertação seja uma realidade.

"É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitoria nem derrota."

(Franklin Roosevelt)

RESUMO

Categorização de textos é uma tarefa em Processamento de Linguagem Natural (PLN) cujo desempenho depende de duas classes de algoritmos: algoritmos de representação textual e algoritmos de classificação. As primeiras soluções para representação textual foram baseadas apenas na contagem das palavras de um determinado corpus existentes no documento, como é o caso do modelo *Bag-of-Words*. Esta é uma abordagem eficiente em alguns casos, apesar de desconsiderar as relações semânticas entre as palavras e apresentar alta dimensionalidade na representação. Para tentar solucionar esse problema surgiram técnicas que representam textos em um nível semântico e com baixa dimensionalidade. Modelos Tópicos e *Word Embedding* são abordagens de representação de texto encontradas na literatura que exploram um nível semântico através da exploração de contexto para a formação de conceitos. Entretanto, em se tratando de textos curtos ou muito curtos, o contexto torna-se muito limitado. Esta dissertação apresenta uma avaliação comparativa de desempenho de diversas combinações de classificador com representações de texto na tarefa de categorização de textos curtos e muito curtos. Adicionalmente, o trabalho propõe e avalia uma representação denominada Combinação Tópicos *Embeddings* (CTE). Os experimentos são realizados sobre coleções de documentos publicamente disponíveis e fornecem subsídios para escolha de combinações classificador-representação entre aqueles avaliados. Textos de duas aplicações de interesse comercial são utilizados: análise de sentimentos e classificação de notícias. Os classificadores considerados são o SVM e o *Naive Bayes* e as representações são TF-IDF, LDA (Modelos Tópicos), *Word Embedding* e a representação proposta CTE.

Palavras-chave: Representação de texto. Categorização de texto. Modelos Tópicos. *Word Embedding*. Textos curtos.

ABSTRACT

Text categorization is a task in Natural Language Processing (NLP) whose performance depends on two classes of algorithms: textual representation algorithms and classification algorithms. The first solutions for textual representation were based only on the word count of a specific corpus in the document, as is the case with the Bag-of-Words model. This is an efficient approach in some cases, despite disregarding the semantic relations between words and presenting high dimensionality in the representation. To try to solve this problem, techniques have emerged that represent texts at a semantic level and with low dimensionality. Topic Models and Word Embedding are text representation approaches found in the literature that explore a semantic level through the exploration of context for the formation of concepts. However, in the case of short or very short texts, the context becomes very limited. This dissertation presents a comparative evaluation of the performance of several combinations of classifier with text representations in the task of categorizing short and very short texts. Additionally, the work proposes and evaluates a representation called Combination Topics Embeddings (CTE). The experiments are performed on public document collections and provide subsidies for choosing classifier-representation combinations among those evaluated. Texts from two applications of commercial interest are used: sentiment analysis and news classification. The classifiers considered are SVM and Naive Bayes and the representations are TF-IDF, LDA (Topic Models), Word Embedding and the proposed representation CTE.

Keywords: Text representation. Text categorization. Topic Model. Word Embedding. Short texts.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

BoC	*Bag-of-Concepts*
BoW	*Bag-of-words*
CBOW	*Continuos Bag-of-words*
CTE	Combinação Tópicos *Embeddings*
GLO	*Glove*
IMDB	*Large Movie View Dataset - Internet Movie Database*
LDA	*Latent Dirichlet Allocation*
LSA	*Latent Semantic Analysis*
LSI	*Latent Semantic Indexing*
NB	*Naive Bayes*
PLN	Processamento de Linguagem Natural
RCV-1	*Reuters Corpus Volume 1*
SRN	Sistema Recomendador de Notícias
SSTb	*Stanford Sentiment Treebank*
STS	*Stanford Twitter Sentiment*
SVD	*Singular Value Decomposition*
SVM	*Support Machine Vector*
TF	*Term Frequency*
TF-IDF	*Term Frequency–Inverse Document Frequency*
WV	*Word2vec*
Z5News	Coleção própria EN - 5 Categorias
Z5NewsBrasil	Coleção própria PT-BR - 5 Categorias

SUMÁRIO

1 INTRODUÇÃO

É conhecimento comum que a internet tornou-se o meio de comunicação convergente da era atual. Ela permite que cada usuário seja ao mesmo tempo um produtor e um consumidor de informação multimídia (texto, imagem, vídeo e som). Essa produção de informação resulta na geração de uma avalanche de dados que são impossíveis de serem analisados sem o apoio de ferramentas igualmente velozes.

Mais especificamente, o desafio é analisar textos que são dados não estruturados. Um texto é uma informação produzida em linguagem natural que segue um conjunto de regras para cada tipo de linguagem. Cada texto pode ser uma opinião apresentada em uma rede social (Twitter, Facebook, YouTube), uma resenha de um filme, uma notícia sobre determinado assunto (política, economia, entretenimento) ou documentos internos de uma organização (relatórios médicos).

Uma vez que um texto apresenta informações disponíveis de maneira não estruturada, o principal desafio em analisar textos é desenvolver técnicas que mapeiem esses dados não estruturados para um espaço dimensional que atenda às tarefas de Processamento de Linguagem Natural (PLN), entre elas, classificação de documentos[1], recuperação da informação, extração de informação e sumarização de texto.

Essa dissertação lida com a tarefa de classificação de documentos texto, mais especificamente com textos curtos, por exemplo, uma opinião da internet ou um trecho de uma notícia. A tarefa de classificação consiste em categorizar automaticamente um texto em uma ou mais classes de um conjunto de classes predefinidas. Um exemplo é a classificação de opiniões sobre determinada temática nas classes positiva (a favor), neutra ou negativa (contra), tarefa essa que é de grande interesse para que empresas possam conhecer a polaridade das opiniões sobre suas marcas, produtos e serviços, e consequentemente melhorar suas políticas de vendas ou suporte. Essa categoria de problemas é denominada análise de sentimentos (PANG; LEE *et al.*, 2008; LIU, 2012).

Outro exemplo é um Sistema Recomendador de Notícias (SRN) (ADENIYI; WEI; YONGQUAN, 2016). Nesse caso, um fluxo de notícias de diversas fontes deve ser classificado em assuntos para ser distribuído aos assinantes de acordo com os seus perfis de interesse. Também nesse caso, o perfil de um assinante nunca está restrito a apenas um assunto. Nota-se que este problema é mais complexo que o anterior pois pode possuir um número grande de classes

[1] Este trabalho utiliza indistintamente os termos categorização e classificação de textos.

que podem representar, por exemplo, conceitos semânticos (esportes, crimes, ...) ou regiões geográficas (um brasileiro morando na Ásia pode está interessado em "notícias do Brasil") e uma mesma notícia pode ser rotulada como pertencendo a várias classes.

A dificuldade de categorização de textos curtos fica clara quando examina-se operações inversamente relacionadas a esta. Textos curtos podem ainda ser oriundos das tarefas de sumarização de textos (LIN; NG, 2019; NASAR; JAFFRY; MALIK, 2018) ou extração de frases ou palavras chave (HASAN; NG, 2014; MERROUNI; FRIKH; OUHBI, 2019). O primeiro caso geralmente gera textos curtos e o segundo geralmente gera textos muito curtos. Essas tarefas podem ser vistas, aproximadamente, como tarefas em sentido inverso daquela de categorização. Em categorização deseja-se extrair o sentido de um texto curto que deve está representando uma descrição mais ampla enquanto que em sumarização deseja-se comprimir o sentido de um texto amplo em um texto curto ou muito curto.

Ao contrário de documentos normais, textos curtos ou muito curtos não apresentam a co-ocorrência de palavras ou informações de contexto suficientes para que as medidas de similaridade funcionem bem. Um caminho para enfrentar essa quantidade limitada de informação presente em textos curtos tem sido o uso de ontologias e tesauros externos para apoiar a tarefa de classificação (WANG *et al.*, 2014; SANTOS; GATTI, 2014; SONG *et al.*, 2014). Uma método para isso é o denominado *Word Embedding*, que representa uma palavra em um modelo de espaço vetorial baseado em semântica distribucional. Entretanto, esse caminho não é sem problemas.

Por outro lado *Word Embedding* pode ser gerado a partir de grandes corpora gerais de toda a língua, tais como a *Wikipedia*, *WordNet* ou corpora da web, ou a partir de corpus de textos do domínio de interesse. Há vantagens e desvantagens em cada um. Neste trabalho, além de investigar o desempenho de classificação com as principais representações tradicionais, investiga-se uma nova representação na qual combina-se *Word Embedding* (*Word2Vec*, *Glove*) (pré treinado em grandes corpora abertos ou treinado no corpus de domínio) com tópicos extraídos de um corpus do domínio.

1.1 MOTIVAÇÃO

A problemática do mapeamento de textos, dados não estruturados, para um espaço dimensional é denominada de Representação de Textos. Em classificação de documentos, assim como a escolha do classificador impacta significativamente na performance do modelo, o método de representação de texto utilizado afeta diretamente na performance do classificador.

Uma representação de texto baseada em frequência de termos apresenta bons resultados para uma abordagem bastante simples que lida apenas com o nível sintático dos termos, contudo, em contrapartida representar um documento dessa maneira implica em alta dimensionalidade para o espaço de *features*. Abordagens que investigam um nível semântico como contexto conseguem reduzir o espaço dimensional, porém, as vezes esses métodos não superam modelos de frequências, podendo apresentar bons resultados dependendo do domínio textual e dependendo de onde essa informação semântica é extraída (seja em um nível fechado ao domínio que o termo se encontra, ou seja explorando domínio externos para o termo).

Em se tratando de textos curtos, uma atenção é necessária. Uma vez que a informação presente em um texto curto pode ser apresentada de maneira objetiva ou que dependa de um contexto de outros textos, a eficiência de representações de textos de nível semântico deve ser investigada. Diante isso, essa dissertação investiga a eficiência de algumas das consolidadas representações textuais que exploram contexto, Modelos Tópicos e *Word Embedding*, na tarefa de classificação de textos curtos. Adicionalmente, o trabalho investiga um novo método de representação que combina Modelos Tópicos e *Word Embedding* através de medidas de similaridade.

O trabalho desenvolvido nesta dissertação é parte integrante do projeto Luppar (SILVA, 2018; SOUZA, 2019), o qual é um projeto de pesquisa em Recuperação de Informação (RI) na Universidade Estadual do Ceará. No restante desta introdução o leitor encontrará os objetivos, a metodologia e a estrutura desta dissertação.

1.2 OBJETIVOS

A questão de pesquisa posta neste trabalho pode ser assim resumida. Conceitos são construções fundamentadas em contexto. Alguns conceitos podem ser detectados a partir de contextos curtos mas outros precisam de um contexto mais amplo para ficarem bem caracterizados. Isso torna legítimo perguntar se conceitos são úteis na categorização de textos curtos, uma vez que o contexto nesses textos é, muitas vezes, limitado. As duas principais representações existentes que capturam a noção de conceito são Modelos Tópicos e *Word Embedding*. Partindo dessa questão, e considerando estas representações padrões, os objetivos da dissertação foram:

1.2.1 Objetivo Geral

Avaliar o desempenho de combinações de representação por conceito versus classificador em categorização de textos curtos.

1.2.2 Objetivos Específicos

O plano de trabalho foi elaborado com os seguintes objetivos específicos:

a) Obter e processar as representações por conceitos das coleções de documentos necessárias para realizar o plano experimental de comparar o desempenho com o estado da arte. Considerar as aplicações de análise de sentimento e classificação de notícias.
b) Avaliar o desempenho em categorização de textos curtos das representações padrões TF-IDF, Modelos Tópicos, e *Word Embedding*. A representação TF-IDF será tomada como linha de base (*baseline*).
c) Analisar medidas de similaridade entre tópicos e representação por *Word Embedding*.
d) Avaliar o desempenho de uma nova representação obtida combinando propriedades da representação padrão por Modelos Tópicos e *Word Embedding* e comparar com as representações padrões.

1.3 METODOLOGIA

O desenvolvimento deste trabalho seguiu aproximadamente os seguintes passos:

- Definição do problema, do escopo da pesquisa e realização da revisão bibliográfica pertinente. Foi definido trabalhar com classificação de texto aplicado em análise de sentimento e categorização de notícias.
- Aquisição de bases de dados e planejamento dos experimentos. Foi decidido utilizar bases publicamente disponíveis em língua inglesa tanto para os experimento em análise de sentimentos quanto para aqueles em categorização de notícias, assim como bases próprias em português coletada de sites de notícias.
- Desenvolvimento de uma representação de documentos texto baseada em tópicos e em contexto e avaliação de desempenho de combinações da representação proposta e das clássicas com diversos classificadores. As representações clássicas utilizadas foram TF-

IDF, Modelos Tópicos e *Word Embedding*. Os classificadores foram SVM e Naive Bayes. A representação proposta foi denominada CTE (Combinação Tópicos-*Embedding*).

- Programação, testes e avaliação de um módulo classificador de documentos texto utilizando Python 3.6 e as bibliotecas NLTK e gensim.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está assim organizado. O Capítulo 2 apresenta uma revisão teórica de representação de documentos texto e das técnicas de classificação utilizadas, e descreve uma nova representação a ser investigada. No Capítulo 3 é feita uma breve revisão de alguns trabalhos relacionados relevantes para esta pesquisa. No Capítulo 4 a metodologia experimental do trabalho é explicitada. O Capítulo 5 apresenta os resultados dos experimentos e uma discussão pertinente. Finalmente, no Capítulo 6 uma conclusão juntamente com as limitações do trabalho e alguns caminhos de investigação são apresentados.

2 FUNDAMENTAÇÃO TEÓRICA

Processamento de Linguagem Natural é uma área que tem como objetivo desenvolver modelos computacionais que dependam de informações expressas em língua natural para compreensão e geração de linguagem (RUSSELL *et al.*, 2010). Processar uma língua natural significa realizar análise em níveis léxico, morfológico, sintático, semântico e pragmático.

A língua natural é um objeto de comunicação entre seres humanos e o processamento dela por sistemas inteligentes é motivado pela comunicação desses sistemas com seres humanos e a extração de informação de uma língua natural (BATES, 1995). Contudo, a extração de informação de uma língua é um ponto crucial nesse desenvolvimento por depender de fatores complexos até mesmo para humanos. Como exemplo dessa dificuldade pode-se citar: a ambiguidade do sentido das palavras de uma certa frase, uma ironia dita em um discurso e o uso de diferentes linguagens.

Neste capítulo apresenta-se resumidamente uma discussão conceitual sobre abordagens de representação textual de classificação de documentos texto. As representações utilizadas no trabalho são: TF-IDF, Modelos Tópicos, *Word Embedding* CTE (abordagem proposta). E os classificadores são SVM e *Naive Bayes*. Descrições detalhadas destas técnicas podem ser encontradas no trabalho de Manning, Raghavan e Schütze (MANNING; RAGHAVAN; SCHÜTZE, 2010) e Webb e Copsey (WEBB; COPSEY, 2011).

2.1 REPRESENTAÇÃO DE DOCUMENTOS TEXTO

Representar um texto, nesse contexto, significa mapear uma informação textual em um espaço vetorial de números reais. Para uma melhor compreensão de representação de texto alguns conceitos são essenciais. Neste trabalho refere-se a um documento como um texto que possa representar uma notícia, uma opinião ou um artigo (escritos em uma determinada língua). Um corpus como sendo uma coleção de documentos que represente uma língua em um determinado cenário ou domínio, por exemplo uma coleção de artigos científicos, uma coleção de notícias de um jornal, uma coleção de comentários de uma rede social. Um termo com sendo uma palavra presente um documento (*1-gram*) ou uma combinação de palavras (*n-gram*) e um dicionário como sendo o conjunto de palavras únicas presentes em uma coleção de documentos.

Cada tarefa de PLN, tais como classificação de documentos e recuperação da informação, necessitam de um determinado corpus devidamente rotulado para cada problemática. Classificação de documentos, descrito mais detalhadamente adiante, é a tarefa de associar au-

tomaticamente um rótulo (classe) para um determinado documento, enquanto recuperação da informação é a tarefa de recuperar um conjunto de documentos relevante para uma determinada consulta. A terceira tarefa relevante em PLN é a extração de informação (RUSSELL *et al.*, 2010).

Extrair e selecionar *features* são etapas que contribuem na performance de um algoritmo de classificação (ALLAHYARI *et al.*, 2017). Uma vez que o objetivo de um classificador é encontrar uma função que consiga separar linearmente ou não linearmente as fronteiras entre classes, *features* bem definidas devem capturar informações relevantes sobre o conteúdo de cada classe.

Em PLN, as *features* para um classificador são o conteúdo do próprio texto. Contudo, compreender o conteúdo e extrair informações que identifique o tipo de informação presente no texto é um grande desafio. Por exemplo, para um pessoa entender que uma determinada notícia, sem identificação do tipo, está relacionada a esportes, é necessário a observação da existência de alguns termos como: futebol, artilheiro, campeonato, liga, técnico, entre outras.

Entretanto essa mesma pessoa pode encontrar uma outra notícia com os mesmos termos citados e em adicional: sonegação, prisão, polícia. Correspondendo agora a uma notícia sobre esportes e polícia (crimes fiscais cometidos por integrantes de um time esportivo). Lidando agora com dois conceitos dentro de uma mesma notícia.

Entender o contexto e conceitos de um texto são pontos chaves na compreensão textual por humanos. Quando fala-se em analisar uma opinião buscamos pontos específicos que caracterize o texto como uma opinião negativa ou positiva através de termos, expressões ou conjunto de termos. Contudo, ao analisar de forma individual cada termo ou expressão pode ocorrer problemas de ambiguidade, sendo necessário observar o contexto de cada termo presente no texto. Por exemplo, ao analisar a sentença de uma opinião sobre um restaurante temos:

*"Esse restaurante fornece um café bem **quente**."*

Sabe-se que café é uma bebida que deve ser consumida quente, concluindo que a opinião dada é positiva. Em uma outra opinião sobre um determinado bar tem-se:

*"A cerveja servida nesse bar é muito **quente**."*

Trazendo para o cenário nacional, onde é entendido que cerveja é consumida gelada, conclui-se que a opinião dada é negativa. Existem dois significados para o termo quente, onde só é possível analisar o significado observando o contexto dos termos que aparecem ao seu redor, no caso cerveja e café.

Assim, percebe-se que informações sobre contexto e conceitos podem ser usadas como *features* no processo de representar cada texto. De forma que a sentença:

*"A cerveja servida nessa bar é muito **quente**"*

pode ser mapeada em um vetor:

$$D = \begin{bmatrix} C_1 & C_2 & C_3 & \dots & C_n \end{bmatrix}, \tag{2.1}$$

de tamanho n, em que n corresponde a quantidade de conceitos e que cada C_i corresponde a conceitos atribuídos arbitrariamente ou descobertos de maneira automática.

Entretanto as mais tradicionais técnicas de representação de texto lidam com o nível sintático da linguagem. Abordagens baseadas na frequência de termos, do inglês *Term Frequency* (TF), em um determinado texto tem sido a mais tradicional desde o início das pesquisas em PLN. As duas principais são *Bag-of-Words* (BoW) e Frequência dos Termos - Inverso da Frequência nos Documentos, do inglês *Term Frequency–Inverse Document Frequency* TF-IDF, descritos a seguir. Contudo essas técnicas assumem que cada termo é independente e descarta completamente as relações de similaridade entre termos e o contexto. Além disso, apresentam como desvantagem representar os textos em espaço de muito alta dimensão (ZHANG; YOSHIDA; TANG, 2011; ASSENT, 2012). Apesar dessa alta dimensão, em tarefas de classificação de documentos essas abordagens apresentam resultados satisfatórios.

2.1.1 BoW

BoW é um modelo baseado em frequência de termos que representa um documento com base na ocorrência dos termos que o compõe (BAEZA-YATES; RIBEIRO-NETO, 2013). Por exemplo, dados os seguintes documentos, em seguida descreve-se o modelo BoW com base em *1-gram*:

1. Eu vivi a vida arduamente.
2. Eu trabalhei arduamente todo dia.
3. Eu nunca trabalhei na vida e nunca vivi arduamente.

Considerando os documentos apresentados, o modelo BoW extrai o dicionário de termos da coleção. Seja $D = \{d_i\}$ o conjunto de documentos da coleção, i o índice de documentos

da coleção, $T = \{t_i\}$ o conjunto de termos do dicionário e j o índice de termos, e seja cada termo recebendo uma ponderação w_{ji} em cada documento da seguinte maneira:

$$M = \begin{array}{c} \\ d_1 \\ d_2 \\ \dots \\ d_{ii} \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \dots & t_j \end{array} \\ \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,j} \\ w_{2,1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ w_{i,1} & \dots & \dots & w_{i,j} \end{bmatrix} \end{array}. \tag{2.2}$$

A matriz M, também chamada de matriz termo-documento ou documento-termo, apresenta a representação BoW para cada documento de D. Cada documento sendo representado como $d_i = \{t_1 : w_{1,i}, t_2 : w_{2,i} ... t_j : w_{j,i}\}$ de modo que w_{ji} corresponda a quantas vezes o termo t_j ocorre no documento d_i.

Mais especificamente, os documentos apresentados são representados conforme mostra o Quadro 1. Observa-se que muitos termos possuem peso 0 por não haver ocorrência no documento, tornando-se M uma matriz esparsa.

Quadro 1 – Representação de documentos com BoW

	A	ARDUAMENTE	DIA	E	EU	NA	NUNCA	TODO	TRABALHEI	VIDA	VIVI
D1	1	1	0	0	1	0	0	0	0	1	1
D2	0	1	1	0	1	0	0	1	1	0	0
D3	0	1	0	1	1	1	2	0	1	1	1

Fonte – Elaborado pelo autor

2.1.2 TF-IDF

O TF-IDF é uma variante de frequência de termos, que ao invés de somente atribuir a ocorrência de um termo como peso W_{ij}, leva em consideração o inverso da frequência do termo em documentos(BAEZA-YATES; RIBEIRO-NETO, 2013). Essa frequência inversa significa o quanto um termo presente em um documento d_i ocorre em outros documentos da coleção. Se esse termo tem alta ocorrência na coleção de documentos ele recebe um peso menor, caso contrário receberá um peso alto, pois será mais discriminativo. Matematicamente, o TF-IDF é

modelado da seguinte maneira:

$$w_{ji} = \begin{cases} (1 + \log f_{ij}) \times \log \frac{N}{n_i}, & \text{se } f_{ij} \geq 0 \\ 0, & \text{senão,} \end{cases} \tag{2.3}$$

onde f_{ij} é a frequência do termo i no documento j, N é a quantidade de documentos da coleção e n_i quantidade de documentos em que o termo i ocorre. Utilizando o exemplo dos documentos apresentados, a representação do TF-IDF é dada conforme mostra o Quadro 2.

Quadro 2 – Representação de documentos com TF-IDF

	A	ARDUAMENTE	DIA	E	EU	NA	NUNCA	TODO	TRABALHEI	VIDA	VIVI
D1	0.59	0.34	0	0	0.34	0	0	0	0	0.45	0.45
D2	0	0.32	0.5	0	0.32	0	0	0.55	0.42	0	0
D3	0	0.2	0	0.34	0.2	0.34	0.68	0	0.26	0.26	0.26

Fonte – Elaborado pelo autor

2.1.3 Modelos Tópicos

Considerando o problema de alta dimensionalidade apresentado nos modelos baseados em frequência de termos, surgem técnicas que representam documentos com informações semânticas e com menor nível de dimensão.

No caso de Modelos Tópicos, que são abordagens não-supervisionadas baseadas em modelos estatísticos, eles são usados para encontrar tópicos que ocorrem em uma coleção de documentos (PAPADIMITRIOU *et al.*, 2000; BLEI, 2012). Cada tópico pode ser interpretado como um grupo formado por termos que compartilham alguma relação semântica. A ocorrência de um termo em um tópico é dada pela probabilidade desse termo pertencer a esse tópico.

Por exemplo, a Tabela 1 mostra o resultado da extração de tópicos sobre uma coleção de documentos apresentado por Liu *et al.*, onde são apresentados três tópicos (Proteína, Câncer, Computação) com os cinco termos mais frequentes em cada tópico. Na Tabela 1, os termos em cada tópico estão ordenadas na ordem decrescente de probabilidade e refletem os conceitos de cada tópico. O tópico 1 é sobre Proteína, por apresentar termos, como: Proteína, Célula, Gene. O tópico 2, sobre Câncer, por conter termos como Tumor, Câncer, Morte. Já o tópico 3, é sobre Computação por conter termos como Computador, Algoritmo, Matemática.

Matematicamente, um tópico é uma distribuição de probabilidade sobre os termos de um vocabulário, onde cada tópico é representado por uma mistura de termos. Da mesma

Tabela 1 – Termos frequentes de três tópicos

Tópicos	Proteína	Câncer	Computação
Termos	Proteína	Tumor	Computador
	Célula	Câncer	Modelo
	Gene	Doenças	Algoritmo
	DNA	Morte	Dado
	Polipeptídeo	Médico	Matemática

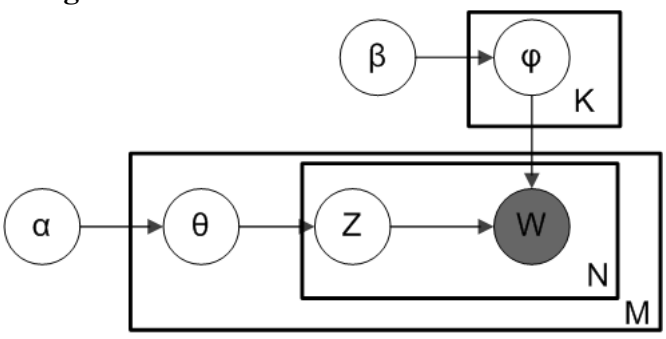
Fonte – (LIU *et al.*, 2016) (Adaptado)

maneira, cada documento apresenta uma mistura tópicos. Formalmente, sendo D uma coleção de documentos, W um dicionário da coleção de documentos D, n um número de tópicos a ser extraído, o objetivo de Modelos Tópicos é que a cada documento da coleção D e que para cada termo de W, sejam atribuídos múltiplos tópicos. Contudo, o ponto chave é como encontrar essa distribuição de probabilidade de termos para cada tópico e a distribuição de tópicos para cada documento.

A saída de um modelo em tópico é uma matriz de distribuição de documentos de tamanho d x n, onde d corresponde a quantidade de documentos e n a quantidade de tópicos extraídos, e uma matriz de distribuição de tópicos de tamanho t x w, onde t corresponde a quantidade de tópicos extraídos e w a quantidade de termos do dicionário. Uma vez que termos e documentos possuem tópicos, pode-se dizer que todo termo e todo documento possui uma probabilidade de aparecer em cada tópico.

Existem diversas abordagens na literatura para extração de tópicos, a mais popular e usada nesta dissertação é o *Latent Dirichlet Allocation* (LDA) (BLEI *et al.*, 2003). LDA é baseado na distribuição de *Dirichlet*, que é usada para modelar termos sobre um tópico e tópicos sobre os documentos. A Figura 1 apresenta o processo de extração de tópicos via LDA, explicado a seguir.

Figura 1 – Latent Dirichlet allocation - LDA



Fonte – (BLEI *et al.*, 2003)

Formalmente, dado o conjunto de documentos D e sendo W o conjunto de termos (vocabulário) e T o conjunto de tópicos, onde T é o resultado da inferência estatística no conjunto de termos W , o LDA modela a geração de documentos dentro de um corpus como o seguinte processo: 1) Uma mistura de k tópicos, θ, é amostrada a partir de uma Distribuição de *Dirichlet* a priori, que é parametrizada por α; 2) Um tópico z_n é amostrado a partir da distribuição multinomial, $p(\theta, \alpha)$, que modela $p(z_n = i|\theta)$; 3) Um termo, w_n, é então amostrado (dado o tópico z_n) através da distribuição multinomial $p(w|z_n)$. LDA é um modelo gerativo e a interação entre essas etapas está representada na Figura 1.

2.1.4 Word Embedding

Word Embedding é um modelo que lida com o nível semântico e que pode ser definido como um conjunto de técnicas de modelagem de linguagem e aprendizagem de *features*, onde o objetivo consiste em mapear termos em um vetor de números reais (JAYAWARDANA *et al.*, 2017). Esse mapeamento também consiste em transformar um espaço de alta dimensão (TF-IDF) em um espaço de menor dimensão. Fundamenta-se no modelo de linguagem baseado na hipótese distribucional, que presume que termos que ocorrem em um contexto similar têm um significado similar(HARRIS, 1954).

Essa busca por representar termos como vetores iniciou com o desenvolvimento de modelos de espaço vetorial para problemas de recuperação da informação. Depois surgiram trabalhos para redução de dimensionalidade na matriz de co-ocorrência (COLLOBERT, 2014; KIM; CHOI, 2000; BULLINARIA; LEVY, 2007). Mais recentemente surgiram trabalhos que usam redes neurais para reduzir a alta dimensão de vetores de termos e representar cada termo em um vetor de contexto(MIKOLOV *et al.*, 2013). *Word Embedding* tem apresentado um bom desempenho em diversas tarefas de PLN.

Nos modelo BoW e TF-IDF eram os documentos que eram representados por vetores. Em *Word Embedding* são os termos que recebem uma representação vetorial. Segue-se descrevendo algumas abordagens de *Word Embedding* (baseado em matriz de co-ocorrência e em redes neurais).

Métodos baseados em Matriz de Co-ocorrência

A matriz de co-ocorrência, ou também chamada de matriz termo por termo, é uma matriz simétrica C (BAEZA-YATES; RIBEIRO-NETO, 2013). Considerando M^T a transposta

de M que é a matriz de frequência de termos de tamanho dxt sendo d os documentos e t os termos, C é definido por

$$C = M \cdot M^T, \tag{2.4}$$

e é uma matriz de correlação de termos em cada elemento c_{uv} é:

$$c_{uv} = \sum_{d_j} w_{uj} \times w_{vj}. \tag{2.5}$$

A medida que t_u e t_v co-ocorram, maior será a correlação c_{uv}, e isso denota que termos que ocorrem próximos tem uma forte correlação e podem apresentar significados ou contextos parecidos.

Aqui cada linha de C, pode ser interpretada como o vetor de representação de cada termo. Contudo aqui ocorre o mesmo problema de representação de documento pela matriz termo-documento: a medida que o vocabulário aumenta, maior será a dimensão da matriz de co-ocorrência. Assim, uma forma encontrada de resolver essa problemática é através da redução de dimensionalidade. O objetivo básico na redução de dimensionalidade é manter apenas as informações relevantes (principalmente eliminando valores iguais a 0) presentes na matriz C, baixando a sua dimensão. Um método tradicional usado para isso, oriundo da álgebra linear é a através de *Singular Value Decomposition* (SVD), ou Decomposição em Valores Singulares.

SVD é uma fatoração de uma matriz complexa ou real dada por:

$$M = U\Sigma V*, \tag{2.6}$$

em que U e V são matrizes unitárias e Σ é uma matriz diagonal. Os elementos na diagonal de Σ são a raiz quadrada dos autovalores de C e U e V são seus autovetores à esquerda e à direita. A técnica de redução consiste em truncar essas matrizes desprezando as linhas de Σ com menores valores, ou seja, cujos autovalores são menos significativos.

Uma abordagem baseada na redução da matriz de co-ocorrência é o algoritmo *Glove* (PENNINGTON; SOCHER; MANNING, 2014). É baseado na probabilidade de um termo co-ocorrer obtida da matriz de co-ocorrência. Com base em cada probabilidade de dois termos co-ocorrerem, o *Glove* cria *Word Embedding* para representar cada termo.

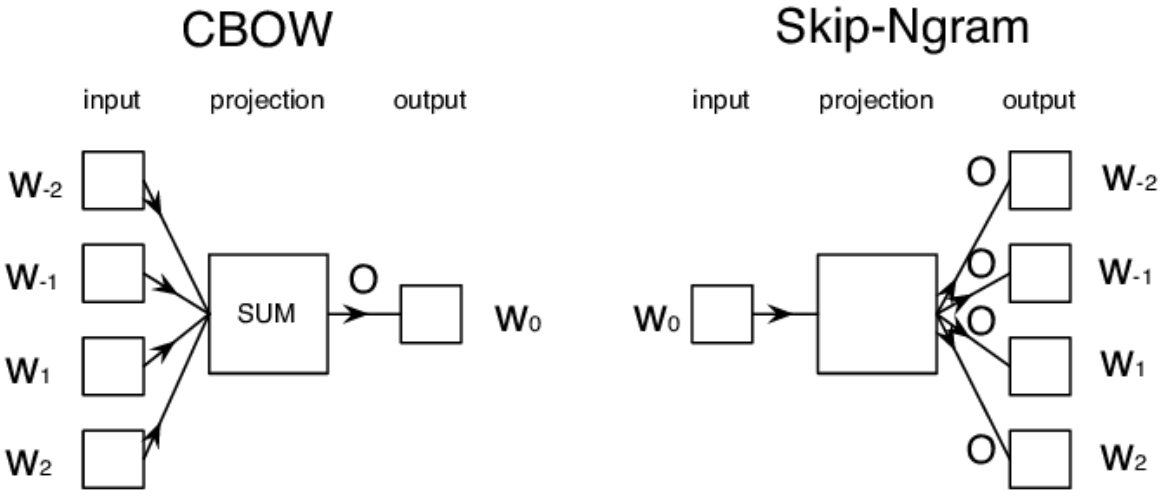
Métodos baseados em Redes Neurais Artificiais

Os modelos baseados em redes neurais, são modelos de linguagem que aprendem a representar de maneira iterativa um termo como um vetor de números reais, denso e de tamanho

definido arbitrariamente. A arquitetura dessas redes neurais é formada por uma entrada, uma camada de pesos e uma camada de saída. Quando a entrada é representada por um termo a saída deseja é uma sequência de termos, já quando a entrada é uma sequência de termos a saída desejada é um termo. À medida que ocorre cada iteração é avaliado o erro entre a saída do modelo e a saída desejada com atualização dos pesos da camada oculta. Essa atualização do erro é uma clássica abordagem da literatura conhecida como *backpropagation* (RUMELHART *et al.*, 1988). Após a convergência do treinamento os vetores de pesos da camada oculta são utilizados como *Word Embedding*.

Existem vários modelos encontrados na literatura para obter *Word Embedding* por meio de redes neurais, sendo o mais conhecido o *Word2Vec* (MIKOLOV *et al.*, 2013), usado neste dissertação. O *Word2Vec* é composto por dois modelos de redes neurais, um denominado *Continuos Bag-of-words* (CBOW) e o outro denominado *Skip-Gram*. A Figura 2 apresenta a arquitetura de rede neural dos modelos CBOW e *Skip-Gram*, respectivamente.

Figura 2 – Arquitetura CBOW e *Skip-Gram*



Fonte – (MIKOLOV *et al.*, 2013) (Adaptado)

CBOW

O CBOW tem por objetivo prever um termo dado um contexto (sequência de termos). Considerando a seguinte sentença presente em uma coleção de documentos *D*:

"O gato pulou a mesa."

Considerando um janela de contexto de tamanho 2, onde uma janela de contexto é definida como os termos que aparecem antes e depois de um determinado termo alvo, pode-se

obter o vetor:

[o, gato, a, mesa]

como sendo o vetor de contexto para o termo **pulou**. Dado esse vetor de contexto o CBOW consegue predizer **pulou** como termo alvo.

No CBOW o vetor de contexto é transformado em uma matriz *one-hot*, que corresponde a uma representação de valores binários para cada termo, onde cada termo é representado como um vetor *one-hot* formado com base nos termos do vetor de contexto. Para esse exemplo, os vetores de contexto são como apresentado no Quadro 3.

Quadro 3 – Matriz *one-hot* para um conjunto de termos

	O	**GATO**	**A**	**MESA**
O	1	0	0	0
GATO	0	1	0	0
A	0	0	1	0
MESA	0	0	0	1

Fonte – Elaborado pelo autor

Essa matriz *one-hot* denominada de X corresponde à entrada do modelo. E o termo alvo corresponde a um vetor Y *one-hot*, formado com base no conjunto de termos da coleção D.

São criadas duas matrizes adicionais V e U. A matriz de entrada V de tamanho n x v, onde n é o tamanho da dimensão da camada oculta (*Word Embedding*) definido arbitrariamente e v a quantidade de termos da coleção D, tal que a $i-$ésima coluna de V é o vetor *Word Embedding* para o termo w_i quando ela é uma entrada para o modelo.

Já a matriz de saída U de tamanho vxn, onde a $j-$ésima linha de U é o vetor *Word Embedding* para o termo w_j quando ela é uma saída para o modelo. Em resumo, cada coluna da matriz V e cada linha da matriz U são os vetores que serão aprendidos para representar cada termo.

Dito isso, o CBOW é desenvolvido nos seguintes passos:

- É gerado um vetor *one-hot* para cada termo do vetor de contexto: $(X^{c-m},\ldots,X^{c-1},X^{c+1},\ldots,X^{c+m})$, onde c corresponde ao índice que possui o valor 1 e m ao tamanho da janela de contexto;
- É obtido os vetores *Word Embedding* por meio da matriz V para cada termo do vetor de contexto : $(VX^{c-m},\ldots,VX^{c-1},VX^{c+1},\ldots,VX^{c+m})$;
- É obtido a média do soma dos vetores *Word Embedding* de cada termo: $\hat{v} = \frac{(VX^{c-m}+\ldots+VX^{c-1}+VX^{c+1}+\ldots+VX^{c+m})}{2m}$;

- É calculado o vetor de score $\hat{z} = U\hat{v}$;
- O vetor de score $\hat{z}$ é transformado em um vetor de probabilidade através de uma função *softmax*, de maneira que $\hat{Y} = softmax(\hat{z})$;
- No final, as probabilidades $\hat{Y}$ devem corresponder com Y, o qual é um um vetor *one-hot* para o termo buscado.

Para que $\hat{Y}$ corresponda a Y é necessário que as duas matrizes V e U sejam bem representadas, para isso é necessário aprender a representação das duas matrizes. Como se trata de duas probabilidades, uma predita e uma verdadeira, a teoria da informação fornece uma medida para comparar as distâncias entre as duas probabilidades, como por exemplo a *Cross-entropy*. *Cross-entropy* é formulado por:

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j log(\hat{y}_j). \tag{2.7}$$

Como Y é um vetor *one-hot* onde apenas um valor é diferente de 0, a fórmula pode ser simplificada como:

$$H(\hat{y}, y) = -y_i log(\hat{y}_i). \tag{2.8}$$

Considerando que a predição foi correta, tem-se $\hat{y}_c = 1$, e calculando $H(\hat{y}, y)$:

$$-1log(1) = 0, \tag{2.9}$$

onde o valor 0 indica que não houve erro. Já em um caso onde a predição foi errada, e $\hat{y}_c = 0.01$, o erro de $H(\hat{y}, y)$ é igual à:

$$1log(0.01) = 4.605. \tag{2.10}$$

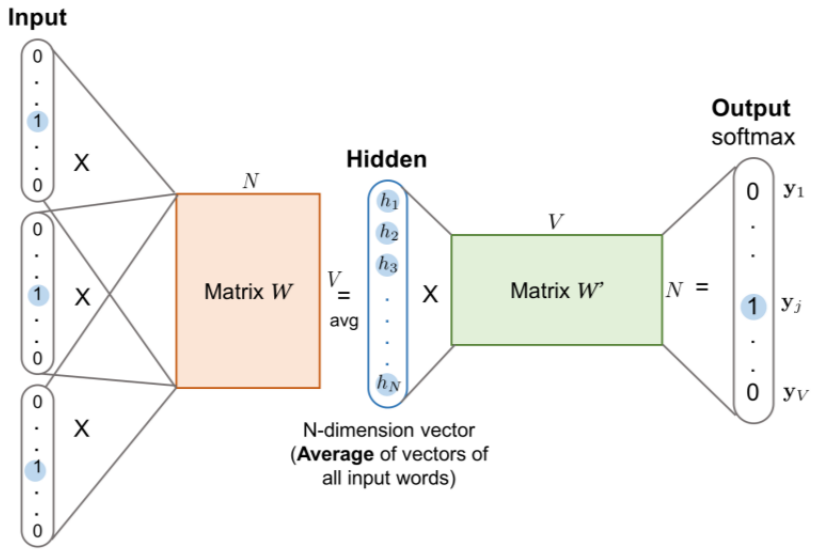
A Figura 3 apresenta a arquitetura do modelo CBOW destacando as dimensões internas e o papel das matrizes U e V.

Skip-Gram

Já o modelo *Skip-Gram* procura realizar a operação inversa do CBOW, uma que ele tem por objetivo prever um contexto dado um termo. Conforme o exemplo citado anteriormente, dado o termo central pulou o *Skip-Gram* prever o contexto:

[o, gato, a, mesa]

Figura 3 – Arquitetura do modelo CBOW



Fonte – (MIKOLOV *et al.*, 2013) (Adaptado)

O funcionamento do *Skip-Gram* é muito similar ao CBOW, contudo, X passa a ser Y e Y passa a ser X. A entrada X é um vetor *one-hot* do termo central, onde o tamanho do vetor corresponde ao tamanho do conjunto de termos da coleção D e a saída Y uma matriz *one-hot* do contexto. As matrizes V e U são as mesmas. E o fluxo é basicamente:

- É gerado um vetor X *one-hot* para o termo de entrada ;
- É obtido o vetor *Word Embedding* para X: $v_c = VX$;
- É gerado os vetores de score: $(u^{c-m}, \ldots, u^{c-1}, u^{c+1}, \ldots, u^{c+m})$ com base em $u = Uv_c$;
- O vetor de score é transformado em probabilidades tal que $\hat{Y} = softmax(\hat{z})$;
- No final, as probabilidades $\hat{Y}$ devem corresponder com Y.

A Figura 4 apresenta a arquitetura do modelo *Skip-Gram* destacando as dimensões internas e o papel das

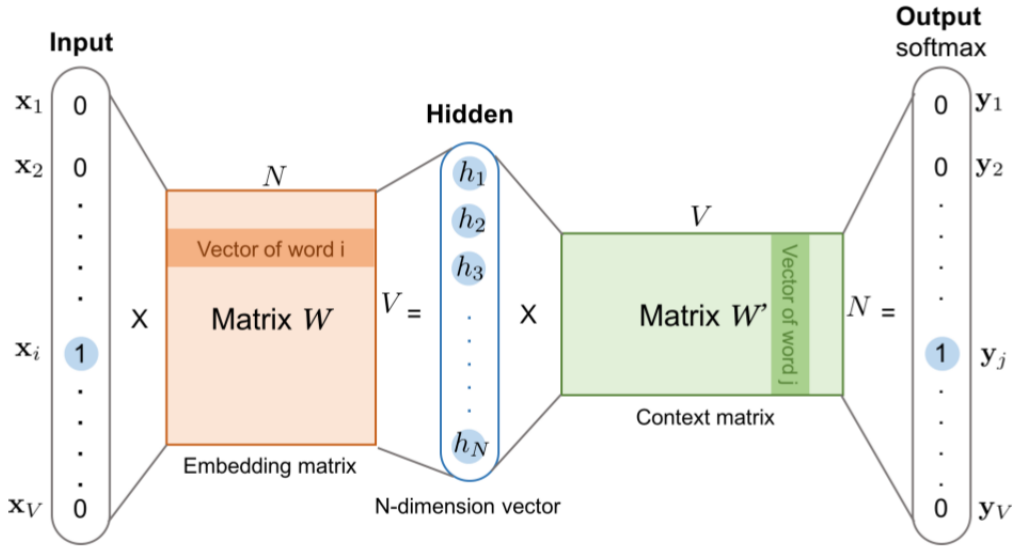
O resultado do CBOW e do *Skip-Gram* são matrizes densas formadas por *Word Embedding* de cada termo. Essa representação permite encontrar relações semânticas, como as apresentas na Figura 5.

Uma vez que *Word Embedding* representa termos, a representação de um documento d por meio dos termos que o compõe pode ser construída de mais de uma forma. A mais simples é dada pela média da soma dos vetores de *Word Embedding* de cada termo, ou seja:

$$v = \frac{(D^1 + \ldots + D^n)}{n}. \tag{2.11}$$

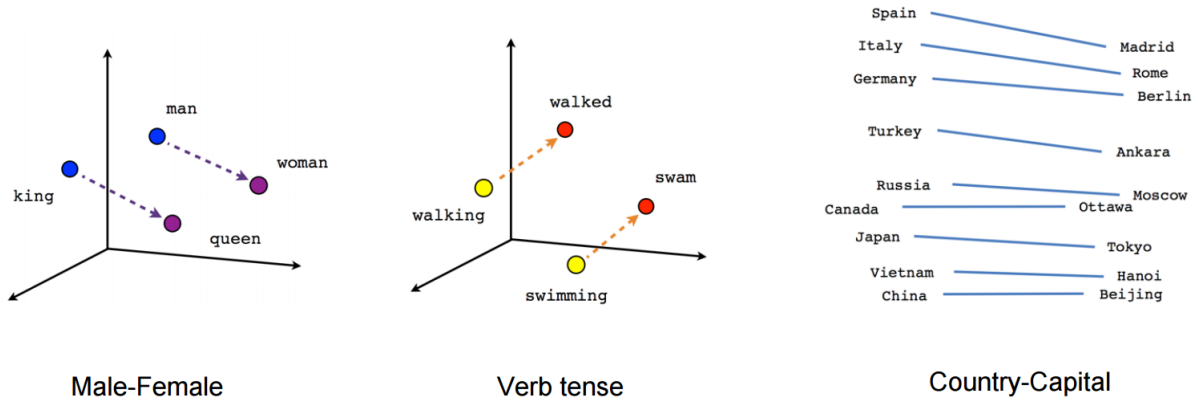
Um outro ponto a ser considerado sobre *Word Embedding* é o domínio de onde o

Figura 4 – Arquitetura do modelo Skip-Gram



Fonte – (MIKOLOV *et al.*, 2013) (Adaptado)

Figura 5 – Relações semânticas entre termos



Fonte – (MIKOLOV *et al.*, 2013) (Adaptado)

contexto dos termos é treinado. Quanto maior for a coleção de documentos e consequentemente maior o dicionário, cada termo apresentará uma quantidade de termos similares dentro de um espaço dimensional maior do que um mesmo termo treinado em uma coleção de documentos menor.

2.1.5 CTE

Uma outra abordagem de representação baseada no nível semântico são modelos baseados em conceitos. A ideia geral desses modelos é extrair conceitos como sendo um conjunto de termos relacionados (através de abordagens não supervisionadas do tipo *clustering*) ou usar informações externas como ontologias ou redes semânticas para expandir e mapear termos do texto em conceitos.

Além de extrair conceitos internamente ou externamente, há uma outra etapa crucial, que diz respeito a representação de um documento em relação ao espaço de conceitos extraídos. Mapear um documento em cada conceito, significa dizer o grau de pertinência em cada conceito. Se n conceitos são extraídos de forma não supervisionada de uma coleção de documentos sobre notícias de jornais, por exemplo, então uma notícia sobre esporte e polícia deverá apresentar uma similaridade maior aos conceitos que abrangem conteúdos sobre esportes e polícia.

A forma tradicional na extração de conceitos de maneira não supervisionada é através de grupos de termos que co-ocorrem ou possuem conceitos similares. Dessa forma um conceito C_i:

$$C_i = \begin{bmatrix} w_1 & w_2 & w_3 & \dots & w_n \end{bmatrix}, \tag{2.12}$$

é composto por um grupo de termos, de modo que cada termo $w_i \in C_1$ tem alguma correlação com os demais termos em C_1. Contudo, tradicionalmente, C_1 é representado de maneira única capturando todo o conjunto de termos em uma única forma, desconsiderando o conteúdo e o contexto de cada termo.

A proposta CTE - Combinação Tópicos *Embedding*

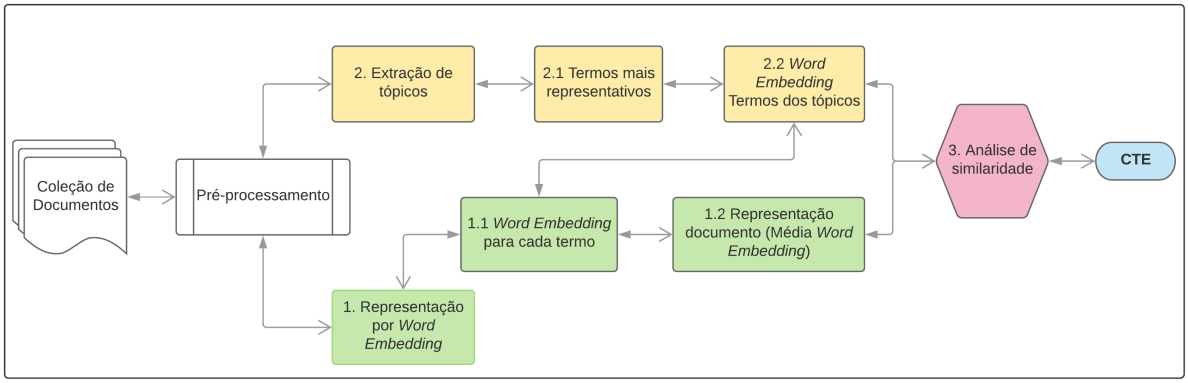
Uma abordagem que leva em consideração o contexto de cada termo é o modelo de Combinação Tópicos *Embeddings* (CTE), uma abordagem que extrai conceitos através de uma abordagem não supervisionada e procura representar cada documento através de medida de similaridade entre o conteúdo do documento e o conteúdo de cada conceito. A ideia básica do CTE é expandir a estrutura de C_1 tratando unicamente cada termo como um conceito e não um conjunto de termos por um conceito, de modo que o conjunto de conceitos C seja:

$$C = \begin{bmatrix} C_{iW1} & C_{iW2} & C_{iW3} & \dots & C_{iWn} \end{bmatrix}, \tag{2.13}$$

onde C_i corresponda a uma conceito extraído e w_j seja o termo presente nesse conceito.

O CTE é dividido em três fases. Para melhor entendimento, a Figura 6 apresenta o fluxo do modelo. Na Fase 1, para uma determinada coleção de documentos, é obtida a representação *Word Embedding* para cada termo e para cada documento da coleção. Na Fase 2 é extraído um conjunto de tópicos, onde os termos com maior probabilidade são utilizados como um conjunto de conceitos. Na Fase 3 é realizada a análise de similaridade entre a representação *Word Embedding* da Fase 1 e a representação *Word Embedding* de cada termo (conceitos) dos tópicos. Segue-se descrevendo cada etapa.

Figura 6 – Combinação Tópicos *Embeddings* - *CTE*



Fonte – Elaborado pelo autor

1. Representação por *Word Embedding*

A representação por *Word Embedding* busca uma representação de termos baseada na hipótese distribucional, representando cada termo como um vetor de n dimensões. No CTE a representação por *Word Embedding* possui duas funções:

1. Representar cada termo (Fase 1.1)
2. Representar cada documento como um vetor de n dimensões (Fase 1.2).

Na Fase 1.1, o *Word2Vec* (MIKOLOV *et al.*, 2013) é treinado com o dicionário da coleção de documentos ou externamente em outro *corpus*. Para o tamanho da dimensão é utilizado $n = 300$, uma vez que esse é tido como parâmetro *default* do algoritmo e utilizado na literatura. Após o treinamento a saída do *Word2Vec* é uma matriz M de tamanho dxn, onde d corresponde ao tamanho do dicionário e n o tamanho da dimensão.

Uma vez obtida a representação *Word Embedding* de cada termo, o objetivo da Fase 1.2 é representar cada documento da coleção com a representação dos termos que o compõe. Formalmente, seja $D = \{d_1,...,d_n\}$ um conjunto de termos de um documento doc_i e seja M a

matriz de representação *Word Embedding* dos termos da coleção de documentos. Então cada documento é representado pelo vetor de cada termo em D obtido da matriz M, formando uma matriz A de tamanho jxk onde j corresponde ao número de termos no documento doc_i e k o tamanho da dimensão da matriz M. Uma vez que as etapas futuras necessitam de representação de mesmo tamanho e mesma dimensão, o vetor f é obtido através média dos termos da matriz A.

2. Extração de tópicos

Tópicos como já descrito anteriormente são modelos que tem o objetivo de descobrir tópicos abstratos em uma coleção de documentos. A extração de tópicos em uma coleção permite obter um conjunto de temas semânticos (conceitos) em cada tópico. Para extração desses tópicos o CTE utiliza o LDA), onde após extrair os tópicos utiliza o conjunto de termos TE formado pelos termos mais representativos e únicos presentes em cada tópico como um conjunto de conceito. E para cada termo de TE é obtida a representação *Word Embedding* através da Fase 1.1.

3. Análise de similaridade

Uma vez obtido um conjunto de conceitos da coleção de documentos, uma medida de similaridade é usada para encontrar uma relação entre um documento e cada conceito. Para isso seja DE a matriz correspondente a representação *Word Embedding* de cada documento obtida na Fase 1.2 e seja TE a matriz correspondente a representação dos termos extraídos via LDA).

Para comparar cada documento com os termos obtidos através do LDA), calcula-se a similaridade cosseno, de acordo com a Equação 2.14 de cada elemento da matriz DE com os elementos da matriz TE obtendo um matriz H de tamanho mxn, onde m representa o número de documentos e n o número de termos (conceitos).

$$cos(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{||\boldsymbol{x}||\cdot||\boldsymbol{y}||}. \tag{2.14}$$

As 3 fases descritas estão dispostas no Algoritmo 1 que apresenta o pseudocódigo para obtenção do CTE.

Algoritmo 1: Pseudocódigo para obtenção do CTE

Entrada: Uma coleção de documentos; um número de tópicos

Saída: Uma representação conceitual para uma coleção de documentos

início

Inicialize uma matriz H;

Treine um modelo de *Word Embedding* (interno ou externo);

Extraia um número de tópicos para a coleção de documentos;

Obtenha os termos tópicos dos tópicos extraídos;

Represente os termos tópicos por *Word Embedding*;

Represente os documentos através da média de *Word Embedding*;

para *cada documento* **faça**

para *cada conceito* **faça**

Calcule a similaridade cosseno entre o documento e o conceito;

Adicione o resultado na matriz H;

fim

fim

Retorne o CTE que corresponde à matriz H;

fim

2.2 CLASSIFICAÇÃO DE DOCUMENTOS

Classificação de documentos é um problema de aprendizagem de máquina baseado em modelos supervisionados. Aprendizagem de máquina pode ser descrita como uma área de inteligência artificial que busca encontrar um padrão com base em um conjunto de amostras tomadas como exemplos.

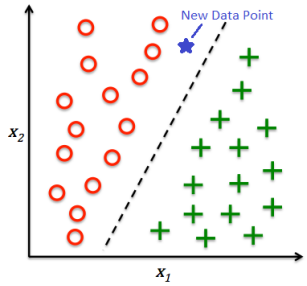
Mais especificamente, modelos de aprendizagem de máquina supervisionados são modelos que aprendem um padrão dado um conjunto de amostras (documentos) e um conjunto de rótulos (a classe a qual cada documento pertence). O conjunto de amostras pode ser descrito com base no conjunto de atributos que pertencem a cada amostra. No caso de classificação de documentos os atributos são as *features* textuais, em que são usadas representação de textos como as descrição na seção anterior.

O processo de aprendizagem pode ser encontrado a partir de uma função discriminante que mapeia cada amostra em sua classe ou estimando uma probabilidade de cada amostra pertencer a uma classe (RUSSELL *et al.*, 2010). Esse processo de aprendizagem é denominado como etapa de treinamento. Depois de aprender esse padrão, o classificador tem a função de

mapear um novo documento não conhecido na fase treinamento, em uma das classes aprendidas.

A Figura 7 apresenta a separação através de um plano de duas classes (uma com elementos em vermelhos e outra com elementos em verdes). O elemento de cor azul exemplifica o processo de classificar um novo elemento, desconhecido no treinamento.

Figura 7 – Separação de classes



Fonte – (SCHOLKOPF; SMOLA, 2001)

Devido a riqueza de classificação de documentos em vários domínios, como classificação de sentimentos, classificação de notícias, classificação de livros, ela pode ser dividia em duas variantes: classificação de documentos mono-rótulo e classificação de documentos multi-rótulos(RUBIN *et al.*, 2012). Classificação mono-rótulo consiste em associar cada documento em apenas uma classe. Um exemplo pode um ser uma avaliação de um produto online com opções de avaliação positiva ou negativa.

Já o problema de classificação multi-rótulo consiste em associar cada documento em uma ou mais classes, por exemplo uma determinada notícia pode ser associada ao tema de esporte, finanças e política. Formalmente seja $X = (x_1, ..., x_n)$ um conjunto finito de instâncias (documentos, notícias) e seja $Y = (y_1, ..., y_m)$ um conjunto finito de rótulos (categorias) de modo que cada instância $x_i \in X$ esteja associada a vários rótulos de classe Y_i, em que $Y_i \subseteq Y$. Dado um conjunto de exemplos de treinamento $S = \{(x_1, Y_1), ..., (x_n, Y_n)\}$ a tarefa de aprendizado de máquina é construir, a partir de S, um classificador f: $X \rightarrow 2^Y$ capaz de estimar uma função de destino desconhecida $\varphi : X \in 2^Y$. Neste contexto, o conjunto 2^Y representa todas as categorizações multi-rótulo possíveis para uma notícia. O problema de multiclasse com um único rótulo é um caso especial do problema de vários rótulos, no qual cada instância recebe apenas um rótulo.

2.2.1 *Naive Bayes*

Classificadores Bayesianos são baseados no teorema de Bayes. O *Naive Bayes* é um modelo que assume a independência entre as *features*. Isso significa adotar que a matriz de covariância das *features* seja uma matriz diagonal. Em classificação de documentos, o *Naive Bayes* tem o objetivo de estimar a probabilidade de um documento pertencer a uma categoria, onde a probabilidade de cada termo em um documento é independente do contexto e da posição do termo no documento. Existem vários modelos que são baseados na hipótese Bayes, como *Bayesian Network*, modelo multivariável de Bernoulli e modelo multinomial. Contudo, todos são construídos a partir do teorema de Bayes que usa o conceito de probabilidade condicional, conforme a equação abaixo:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}, \text{ para } i = 1 \ldots c\ , \tag{2.15}$$

onde c é a quantidade de classes, $P(\omega_i|x)$ é a probabilidade a posteriori da classe ω_i dada a observação do padrão x, $p(x|\omega_i)$ é a função densidade probabilidade condicional da classe, $P(\omega_i)$ representa a probabilidade a priori da classe ω_i e $p(x)$ é dado por:

$$p(x) = \sum_{i=1}^{c} p(x|\omega_i)P(\omega_i), \tag{2.16}$$

onde $p(x)$ é a probabilidade a priori do vetor de treinamento x, e uma constante de normalização, sendo independente da classe.

Uma vez que o classificador de Bayes é determinado pela probabilidade condicional da classe $p(x|\omega_i)$, ela pode ser determinada pela densidade normal ou gaussiana (WEBB; COPSEY, 2011). Nesse caso ele é chamado de Classificador *Naive Bayes* Gaussiano.

A Equação 2.17 representa a distribuição Gaussiana univariada e a Equação 2.18 representa a distribuição Gaussiana multivariada:

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], \tag{2.17}$$

$$p(x|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right], \tag{2.18}$$

onde x é o vetor de atributos de tamanho d para uma determinada amostra, μ a média de x, σ o desvio padrão e Σ a matriz de covariância.

As equações discriminantes de cada classe para o classificador *Naive Bayes* podem ser obtidas calculando o Log da verossimilhança na Equação 2.18 para obter (WEBB; COPSEY, 2011):

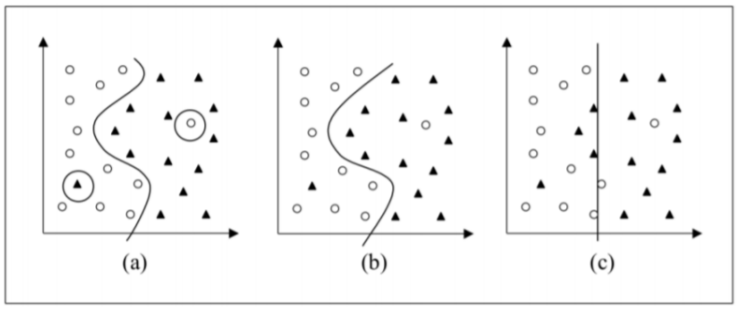
$$g_i = p(x|\omega_i) - \frac{1}{2}Ln(\Sigma|^{-1}) + (x-\mu)^T\Sigma^{-1}(x-\mu). \tag{2.19}$$

2.2.2 *Suport Machine Vector*

Support Vector Machines (SVM) (CORTES; VAPNIK, 1995), ou Máquina de Vetores de Suporte, é um algorítimo de aprendizagem supervisionado utilizado para problemas de classificação e de regressão. O SVM é baseado na Teoria do aprendizado estatístico, que estabelece princípios para que classificadores sejam bons generalizadores para uma predição correta de novos dados com base no domínio de aprendizado.

Tomando como exemplo a Figura 8 (SCHOLKOPF; SMOLA, 2001), a função do classificador no processo de aprendizagem é separar os dados de cada classe.

Figura 8 – Exemplos de hiperplanos



Fonte – (SCHOLKOPF; SMOLA, 2001)

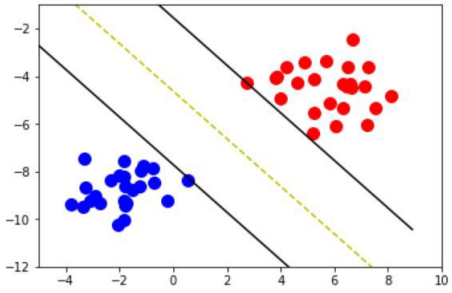
Uma hipótese é representada por uma fronteira de decisão. Para ilustrar esse processo, observa-se nesta figura três hipóteses de separação das classes (circulo e triângulo). Na primeira hipótese o modelo está muito específico (superajustamente) para o conjunto de treinamento, classificando corretamente até os ruídos presentes em cada classe e consequentemente terá uma alta possibilidade de classificar errado novas amostras. Na terceira hipótese, o modelo não está ajustado ao treinamento (sub-ajustamento), desconsiderando os pontos que estão muito próximos

da fronteira de uma das classes, e por isso apresentando muitos erros. Por fim, na segunda hipótese, um meio termo é apresentado com um ajustamento ideal para as duas classes, acertando grande parte dos dados.

É justamente essa separação ótima que é o objetivo do SVM, chamado separador de margem máxima (conhecido também como fronteira de decisão ótima). Inicialmente proposto para classes linearmente separáveis e estendido posteriormente para classes não-linearmente separáveis.

O separador é encontrado onde a distância de separação entre classes é máxima, ou seja, um limite de decisão com a maior distância possível a pontos de exemplos. Essa margem máxima também pode ser entendida como o limite da medida de confiança do classificador. As amostras que estão localizadas nas margens desse separador são as mais informativas para a criação do limite de decisão da classificação e são denominadas vetores de suporte. A Figura 9 apresenta o hiperplano separador com os vetores de suporte de cada classe dispostos na margem máxima do hiperplano separador. A formulação matemática e a resolução deste problema pode ser encontrada em detalhes em (WEBB; COPSEY, 2011).

Figura 9 – Margem de decisão do SVM e Vetores de Suporte



Fonte – (SCHOLKOPF; SMOLA, 2001) (Adaptado)

Em problemas nos quais as classes não são linearmente separáveis, o SVM usa um recurso de mapeamento do espaço original em um outro espaço com uma dimensão em que os dados tornam-se, supostamente, linearmente separáveis. Por exemplo, considerando R^2 um espaço com dimensão de tamanho 2, o SVM utiliza uma função $\Phi : x \rightarrow \phi(x)$ de modo que $R^2 \rightarrow R^3$ seja mapeado em um espaço com dimensão de tamanho 3. À medida que a dimensão do espaço destino aumenta, aumenta também a probabilidade de esse problema se tornar linearmente separável.

Essa função Φ é definida como uma função *kernel*, onde a escolha de uma função *kernel* específica deve ser analisada de acordo com o domínio do problema. A Tabela 2 apresenta

exemplos de *kernels* mais usados:

Tabela 2 – Funções *Kernel* para SVM

Kernel	Função
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$
RBF	$exp(-\sigma \|\|x_i - x_j\|\|^2)$
Sigmoidal	$tanh(\delta(x_i \cdot x_j) + k)$

Fonte – Elaborado pelo autor

O problema de otimização gerado é quadrático com as restrições lineares. A sua solução envolve passos matemáticos com a introdução de uma função Lagrangiana (α) e tornando suas derivadas parciais nulas. Tem-se como resultado o seguinte problema na sua forma dual:

$$Maximizar_\alpha \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \, (x_i \cdot x_j), \tag{2.20}$$

$$\text{Com as restrições: } \begin{cases} 0 \leq \alpha_i \leq C, \quad \forall_i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}, \tag{2.21}$$

onde $S = \{(x_1, Y_1), ..., (x_n, Y_n)\}$ são o conjunto de treinamento.

A escolha dos parâmetros do *kernel* do SVM são críticas para a obtenção de bons resultados. No caso específico de seleção dos parâmetros do *kernel* RBF, a busca é realizada em um espaço formado pelo parâmetro γ e pelo parâmetro de custo C, onde o objetivo é encontrar uma melhor configuração para ambos, que leve à um bom desempenho do classificador. Esse trabalho utilizou $\gamma = 0.01$ e $C = 1.0$. Esses hiper-parâmetros podem ser calibrados utilizando ferramentas tais como o *grid-search* do *scikit-learn*[1].

Não é fácil encontrar um *kernel* ótimo para uma aplicação. As propriedades destes *kernels* listados são quase universais de forma que eles são utilizados em quase todos os domínios de aplicação. Após alguns testes, os experimentos realizados nesta dissertação utilizou o *kernel* RBF. Para que seja possível obter resultados satisfatórios é necessários definir bons valores para os parâmetros do *kernel* RBF. Como o *kernel* RBF possui a forma de um sino, o parâmetro y define a largura da curva de sino. Esse parâmetro também foi definido de forma experimental.

[1] *Grid-search* do *scikit-learn* - Disponível em https://scikit-learn.org/stable/modules/grid_search.html

3 TRABALHOS RELACIONADOS

Categorização de textos em geral e categorização de textos curtos em particular, são tarefas amplamente estudadas na literatura devido a sua importância conceitual e econômica para as aplicações. O leitor pode encontrar bons *surveys* em categorização de textos utilizando as abordagens tradicionais de representação de documentos nas referências (KOWSARI *et al.*, 2019; MANIKANDAN; SIVAKUMAR, 2018; KORDE; MAHENDER, 2012; SEBASTIANI, 2002). Este capítulo irá revisar apenas aqueles trabalhos baseados na ideia de representar um documento como um "saco de conceitos"*Bag of Concept* (BoC) ao invés de BoW e suas derivações, por estarem aqueles conceitualmente mais próximos do método investigado aqui.

Uma tarefa de PLN depende de uma representação de texto, e a representação apropriada depende do tipo de tarefa. Assim uma tarefa de classificação depende de uma representação para executar com maior precisão. Neste capítulo serão listadas algumas abordagens encontradas na literatura que buscaram desenvolver uma representação de documentos por conceitos seja com o uso de recursos externos ou extraindo conceitos internamente. São listados ainda alguns trabalhos que representam documentos por meio de Modelos Tópicos ou *Word Embedding*, e é através da combinação dessas duas abordagens que é desenvolvida uma representação por conceitos nessa dissertação.

3.1 CONCEITOS EXTRAÍDOS INTERNAMENTE

Um dos trabalhos pioneiros em BoC foi o de Sahlgren e Cöster (SAHLGREN; CÖSTER, 2004). Nele os autores desenvolveram um método BoC sem o uso de recursos externos e escalável. Para extrair conceitos, eles usaram o *Random Index* (KANERVA; KRISTOFERSSON; HOLST, 2000; KARLGREN; SAHLGREN, 1996; JOHNSON; LINDENSTRAUSS, 1984; SAHLGREN; KARLGREN, 2005), que é um método para produzir vetores de contexto de palavras baseadas em dados de co-ocorrência. O *Random Index* difere-se por não usar uma matriz de co-ocorrência F de tamanho wc (cada F_w representa as palavras, e cada colunas F_c representa os contextos) e sim uma matriz de contexto G que é construída incrementalmente, toda vez que uma palavra ocorre em um contexto ela é adicionada ao vetor de contexto da palavra especificada. Os conceitos são produzidos somando os vetores de contexto das palavras que ocorrem em um texto criando uma representação por conceitos como entrada em um classificador SVM. Os autores usaram a coleção de notícias *Reuters*-21578 para verificar se alguma categoria de notícias apresentava alguma melhoria com o uso de BoC. Os autores também demonstraram

que a combinação de BoC e TF-IDF melhora o desempenho da SVM em relação a BoC ou TF-IDF sozinhos.

Um outro trabalho que busca extrair conceitos internamente é o de Kim, Kim e Cho (KIM; KIM; CHO, 2017). Nele os autores encontram desvantagens em modelos como de BoW (a alta dimensão apresentada pelo modelo), bem como em *Doc2Vec* (relatam a falta de interpretabilidade no sentido de cada dimensão do *Doc2Vec*). A proposta dos autores é extrair conceitos através de *cluster* de palavras representadas via *Word2Vec*, e então usar a frequência dos *clusters* para representar cada documento. Essa frequência é similar ao TF-IDF, porém utilizando frequência e inverso de franquênia dos conceitos ao invés dos termos. Dessa forma, com o uso de *cluster* de palavras, a abordagem preserva o significado e relações semânticas de palavras além de fornecer uma intuitiva interpretação para cada grupo de conceitos. Após extrair os conceitos e representar cada documento, os autores avaliaram o modelo em tarefas de classificação, comparando-os com os modelos *Doc2Vec*, BoW, *Latent Semantic Analysis* (LSA) e média de *Word Embedding*. Eles alcançaram 82.86% de F1-score com 100 dimensões no conjunto de dados R52 contra X% alcançados em outros trabalhos comparados.

Um outro trabalho baseado em extrair conceitos a partir de *Word Embedding* é o de Chen, Verspoor e Zhai (CHEN; VERSPOOR; ZHAI, 2019). Nele os autores apresentam um forma de extrair conceitos para um determinada palavra em um documento. Usando o *Glove*, treinado em um grande corpus, os autores implementaram o *Word2Concept* com a finalidade de mapear palavras similares para uma palavra tomada como entrada. Dessa forma as palavras similares são consideradas como constituindo um conceito.

Outros trabalhos tem se baseado no uso de redução de dimensionalidade, como nos trabalhos de Deerwester *et al.* (DEERWESTER *et al.*, 2015), Bhalchandra *et al.* (BHALCHANDRA *et al.*, 2016), Kim, Rowland e Park (KIM; ROWLAND; PARK, 2005) que são baseados em LSA, desenvolvidos a partir de SVD. Com base na decomposição da matriz termo-documento em bases ortogonais, o qual podem ser considerados como conceitos artificiais. Em Täckström (TÄCKSTRÖM, 2005) o autor propôs uma comparação entre métodos de redução de dimensionalidade (*Feature Selection*, *Random Mapping*), com o propósito de que cada dimensão do novo espaço dimensional possa ser interpretado como conceitos, uma vez que esse espaço dimensional carrega informações do espaço inicial.

3.2 CONCEITOS EXTRAÍDOS EXTERNAMENTE

Um trabalho que se baseia no uso de recursos externos para extração de conceitos é o trabalho de Mouriño-garcía, Pérez-rodríguez e Anido-rifón (MOURIÑO-GARCÍA; PÉREZ-RODRÍGUEZ; ANIDO-RIFÓN, 2015). A motivação dos autores para desenvolver essa representação BoC é com base na solução do problema de sinonímia e polissemia, uma vez que os conceitos não são ambíguos. Eles usaram a *Wikipedia* como fonte externa de extração de conceitos, implementando o algoritmo proposto por (MILNE; WITTEN, 2013). Esse algoritmo realiza uma busca que leva como entrada cada *n-gram* presente no documento, com a finalidade de encontrar um conjunto de *n-grams* de documentos âncoras presentes na *Wikipedia*. Com esse conjunto de *n-grams* o algoritmo encarrega-se de selecionar os mais relevantes *n-grams* por meio de um processo de desambiguação, onde por meio de um modelo treinado nos artigos da *Wikipedia* que contém os exemplos de desambiguação feito manualmente. Por fim, o algoritmo utiliza um outro modelo treinado para definir o grau de relevância desses conceitos extraídos do texto em exame. No experimento de avaliação, os autores usaram o algoritmo SVM e compararam o desempenho do modelo BoC com o modelo tradicional BoW. O modelo de BoC foi superior ao BoW atingindo 70% de F1-score no conjunto de dados *Reuters*. Entretanto os autores não informaram as dimensões dos modelos avaliados.

Em análise de sentimentos, o trabalho de Chung, Wu e Tsai (CHUNG; WU; TSAI, 2014) propôs o uso de um dicionário denominado *SentiConceptNet* (TSAI *et al.*, 2013), esse dicionário contém conceitos com valores de sentimentos entre -1 e 1. É com base nesse dicionário de conceitos e na abordagem de extração de conceitos via grafos do trabalho de Rajagopal *et al.* (RAJAGOPAL *et al.*, 2013) que eles representaram revisões da coleção *Blitzer* como BoC. Eles obtiveram bons resultados em um experimento de classificação usando SVM.

3.3 COMBINAÇÃO DE TÓPICOS E *WORD EMBEDDING*

A combinação entre tópicos e *Word Embedding* é trabalhada por alguns autores. Porém eles não estão buscando propriamente uma representação por conceitos e sim uma representação de palavras que explore mais as relações semânticas.

Como é o caso do trabalho de Liu *et al.* (LIU *et al.*, 2015), em que os autores propõem uma abordagem chamada *Topical Word Embeddings* (TWE) onde combinam tópicos e *Word Embedding* como forma de melhorar abordagens tradicionais de *Word Embedding* que representam cada palavra com um único vetor. Eles utilizam LDA para extrair tópicos e

atribuí-los para cada palavra da coleção. Uma vez que uma palavra pode apresentar mais de um tópico, correspondendo a mais de um significado na coleção, a intenção deles é criar uma representação para cada significado de uma determinada palavra. Por exemplo, a palavra *apple* pode indicar uma fruta no tópico comida, assim como uma empresa de TI no tópico de tecnologia da informação.

Para criar esses vetores, eles realizaram adaptações no algoritmo *Skip-Gram*(MIKOLOV *et al.*, 2013) e analisam três propostas. Em cada uma eles aumentam a representação de modo que em cada uma está presente tanto a palavra, como uma pseudo palavra criada para o tópico correspondente. Eles representaram documentos utilizando *Word Embedding* de palavras criadas pelo TWE e testaram em uma tarefa de classificação onde alcançaram 80.6% da medição F1 com 400 dimensões no conjunto de dados *20NewsGroup*. É com base nesse trabalho que é explorado nessa dissertação possibilidade de usar palavras extraídas por tópicos como conceitos, e usar representação *Word Embedding* para encontrar uma similaridade entre um documento e cada conceito.

No trabalho de Li *et al.* (LI *et al.*, 2016) os autores combinam tópicos e *Word Embedding* no modelo denominado *TopicVec*. O *TopicVec* é motivado por base em que *cluster* semânticos (palavras relacionadas em um documento que possuem a mesma direção) e tópicos possuem a mesma natureza. Com esta similaridade, a proposta é estender um modelo chamado PSDVec () através da incorporação de tópicos, colocando uma função de associação que modele a distribuição de palavras no tópico, no lugar da distribuição categórica do LDA. A vantagem dessa função link é que o relacionamento semântico entre palavras já é incorporado com o uso da distância cosseno no espaço de *Word Embedding*. Não há melhoria de performance mas os resultados são comparáveis aos anteriores.

3.4 CATEGORIZAÇÃO DE TEXTOS CURTOS

Em relação a textos curtos, alguns trabalhos buscaram encontrar métodos que resolvessem o problema da limitação de informação contextual em textos curtos. Um exemplo é o trabalho de Santos e Gatti (SANTOS; GATTI, 2014), nele os autores trabalharam com classificação de sentimentos em texto muito curto encontrados no *Twitter* (na época com 140 caracteres) e em revisões de filmes. Eles construíram uma rede neural denominada *Character to Sentence Convolutional Neural Network*, onde utilizaram caracteres e sentenças dos textos para extração de *features*. Para representar cada sentenças eles utilizaram *Word Embedding* treinadas

com *Word2Vec* no corpus da Wikipédia, explorando um conhecimento externo de informações contextuais para os textos curtos. Eles realizaram os experimentos nas coleções de documentos *Stanford Sentiment Treebank* (SSTb) e *Stanford Twitter Sentiment* (STS). Eles atingiram uma acurácia de 85.7% para o SSTb e 86.4% para o STS.

No trabalho de Wang *et al.* (WANG *et al.*, 2014), eles utilizaram a ideia de representação por conceitos BoC para representação de textos curtos. Para os autores um conceito é denominado como um conjunto ou classes de entidades pertencentes a um determinado domínio, como palavras similares. Eles propuseram um *framework* para classificação de textos curtos, onde para cada classe de uma coleção de documentos eles expandem os textos através de uma rede semântica denominada *Probase* que contém milhões de conceitos. Para eliminar conceitos similares presentes na *Probase*, eles realizaram um agrupamento de conceitos. Uma vez que esses conceitos foram extraídos eles utilizam uma medida de peso para determinado o grau de cada conceito em um texto. Para avaliação foi proposto um experimento de recomendação de *queries* extraídas de um determinado site, onde *queries* desconhecidas são classificadas e recomendadas a um usuário que busca por algo parecido, onde atingiram 0.66 de F1-score.

Zhang e Wu (ZHANG; WU, 2015) propuseram um método para expandir informações em textos curtos através de *n-gram*. No conjunto de treinamento da própria coleção, eles extraem *n-gram* e para cada para texto calculam a probabilidade de outras palavras que não existem no texto original, e se a probabilidade é maior que um limite atribuído pelos autores então aquela palavra é adicionada no texto. Para avaliação eles utilizaram a coleção de documentos *Sina Weibo* que contém textos de micro-blogs distribuídos em 11 categorias, onde sem expansão dos termos atingiram 0.75% de F1 e com expansão 0.85% de F1.

4 METODOLOGIA

Neste capítulo é apresentada a metodologia detalhada seguida nesta pesquisa. Ela consiste na descrição das coleções de documentos utilizadas, no plano experimental e nas métricas e etapas do processo de avaliação das combinações entre representações e classificadores.

4.1 COLEÇÕES DE DOCUMENTOS

Uma vez que a proposta de avaliação desta dissertação é a classificação de textos curtos, este trabalho utiliza coleções de documentos em inglês (multi-rótulo e mono rótulo) encontradas na literatura para classificação de notícias e classificação de sentimentos, assim como coleções de documentos próprias em inglês e português coletadas durante a pesquisa. Santos e Gatti (SANTOS; GATTI, 2014) definem um texto curto como um texto de até 10 palavras. Essa definição cobre um certo número de aplicações tais como mensagens do *Twitter*, a maioria dos comentários muito curtos e de opiniões sobre produtos além de lides (*lead*) de notícias. Entretanto, para efeito desta dissertação considerou-se texto curto como aqueles com média de até 130 palavras, com o objetivo de englobar notícias curtas que vão além da lide, comentários e opiniões mais longas e resumos em geral de textos maiores tais como resumos de artigos científicos, prontuários médicos, entre outros. Diante disso, foram escolhidas as seguintes coleções:

- ***Reuters* Corpus Volume 1 (RCV-1):** É uma coleção que contém 800.000 notícias do site *Reuters* organizadas em 103 categorias (Economia, Ciência e Tecnologia, Esportes, Corporações, entre outras) (LEWIS *et al.*, 2004). É uma coleção multi-rótulo, onde cada notícia está associada a mais de uma categoria. Devido ao custo computacional de pré-processamento e representação da coleção, foi utilizada apenas uma amostra de 33.149 itens de notícias escolhido de maneira aleatória divididos em 101 categorias. O tamanho médio dos textos neste data set é de 123 palavras;
- ***20Newsgroups*:** É uma coleção que contém 18.821 documentos distribuídos em 20 categorias (Tecnologia, Política, Religião, Esporte entre outros). É um coleção mono-rótulo, onde cada documento é associado a apenas uma categoria. Foi usado todo o conjunto de dados no experimento. O tamanho médio dos textos neste data set é de 102 palavras;
- ***Large Movie View Dataset - Internet Movie Database* (IMDB)**: é uma coleção de documentos binária que contém 50.000 revisões de filmes presentes no site do IMDB, organizadas em duas categorias (positivas e negativas) (MAAS *et al.*, 2011). Para con-

siderar uma revisão como positiva ou negativa, os autores levaram em conta a nota de cada revisão. Uma revisão com nota maior/igual a 7 é rotulada como positiva, e negativa quando a nota for menor/igual a 4. Cada revisão está associada a apenas um rótulo (mono-rótulo) e não há revisões neutras. Devido ao custo computacional de pré-processamento e representação da coleção, foram utilizados apenas uma amostra de 25000 itens escolhido de maneira aleatória. Esta coleção é amplamente utilizada em aplicação de análise de sentimento binária. O tamanho médio dos textos neste data set é de 119 palavras;

- ***Stanford Twitter Sentiment* (STS)**: é uma coleção de documentos (GO; BHAYANI; HUANG, 2009) que contém originalmente 1.6 milhões de *tweets* rotulados em duas categorias de polaridade de sentimentos (negativa e positiva), contudo nos experimentos dessa dissertação foi utilizado apenas uma amostra do conjunto de dados (foram selecionados 19162 *tweets* de maneira aleatória). O tamanho médio dos textos nesta coleção é de 7 palavras. Essa coleção é considerada muito curta enquanto que os demais foi considerado curto;
- **Coleção própria EN - 5 Categorias (Z5News)**: é uma coleção própria criada por (SOUZA, 2019) e contém 43.301 notícias curtas em inglês, devidamente rotuladas e distribuídas em 5 tópicos categorias (esportes, política, tecnologia, finanças e brasil), extraídas do site de notícias: *Reuters* (entre os anos de 2019 e 2011);
- **Coleção própria PT-BR - 5 Categorias (Z5NewsBrasil)**: é uma coleção própria criada por (SOUZA, 2019) e contém 22.833 notícias curtas em português, devidamente rotuladas e distribuídas em 5 categorias (esportes, política, tecnologia, finanças e brasil), extraídas do site de notícias: G1 notícias (entre os anos de 2019 e 2011).

4.2 OBTENÇÃO DE REPRESENTAÇÕES

Antes de representar cada documento foi realizado o pré-processamento de cada coleção de documento. O objetivo do pré-processamento é realizar a limpeza e normalização de dados textuais não-estruturados para que seja possível obter representações estruturadas e compreensíveis para diversos algoritmos de várias tarefas de PLN (BAEZA-YATES; RIBEIRO-NETO, 2013).

O pré-processamento de texto é dividido nas etapas descritas abaixo:

- **Análise léxica de texto**: Essa fase consiste na identificação de palavras (*1-gram*) em um texto. Contudo a análise léxica abrange bem mais que identificar e remover espaços vazios. É considerado detecção de número, hífen, carácter especial, caixa alta.

- **Remoção de *Stopwords***: Consiste em eliminar termos com baixo valor significativo e discriminativo no texto, tais como artigos, preposições, conjunções etc.
- ***Stemming***: Consiste em reduzir palavras flexionadas ou derivadas para a sua raiz. *Stemming* é útil para relacionar palavras com variações de tempo verbal e grau em uma mesma forma.

A Tabela 3 detalha as características de cada coleção de documentos após o pré-processamento, tais como número de documentos, números de classes e média de termos.

Tabela 3 – Características das coleções de documentos após pré-processamento

Coleção	Idioma	Tipo	Docs	Classes	Média/Termos	Dicionário
RCV-1	EN	Multi-rótulo	33.149	101	123	56269
20Newsgroups	EN	Mono-rótulo	18.214	20	102	90211
IMDB	EN	Mono-rótulo	25.000	2	119	74486
STS	EN	Mono-rótulo	19.162	2	7	26983
Z5News	EN	Mono-rótulo	43.301	5	22	26909
Z5NewsBrasil	PT-BR	Mono-rótulo	22.833	5	15	17183

Fonte – Elaborado pelo autor

Depois que o pré-processamento foi realizado, foram obtidas as representações por TF-IDF, e as representações que exploram o nível semântico por tópicos via LDA, *Word Embedding* via *Word2Vec* e *Glove* e CTE para todos os documentos de cada coleção. Abaixo está listada a configuração para cada representação:

- Para o TF-IDF foi obtido uma matriz esparsa para redução de custo computacional.
- Em relação ao LDA, foi variado o número de tópicos definido arbitrariamente (10, 50, 100, 200), onde cada documento é representado pela probabilidade de pertencer a cada tópico. Essa faixa de número de tópicos é comumente encontrada na literatura.
- Em relação a representação via *Word Embedding* obtida via *Word2Vec* com os termos de cada coleção de documento, foi utilizada uma janela de contexto de tamanho 5 aplicado ao algoritmo *Skip-Gram* e uma dimensão de *Word Embedding* com tamanho 300. A dimensão foi escolhida com base em experimentos iniciais com variações (50,100,300,500). O valor desse parâmetro é o mesmo utilizado em quase toda a literatura consultada. Em sua forma *baseline*, cada documento é representado pela média do somatório de cada representação *Word Embedding* que o compõe;
- Em relação a representação via *Word Embedding* através do algoritmo *GLove* com os termos do corpus da *Wikipédia*, também foi utilizado o tamanho 300 na dimensão de *Word Embedding* conforme é prática ampla.

- O modelo CTE investigou a mesma variação entre o mesmo número de tópicos apresentados no experimento com LDA, assim como utiliza a variação de *Word Embedding* treinadas localmente com *Word2Vec* e treinadas externamente com *Glove*, com as respectivas dimensões de *Word Embedding* (300 e 100).

4.3 ESTRATÉGIA DE CLASSIFICAÇÃO

Para as coleções de documentos que apresentam mais de duas classes (RCV-1, *20Newsgroups*) foi aplicada a estratégia *One-Against-All*, que constrói $|C|$ modelos binários de previsão, onde $|C|$ é o número de classes. Cada modelo é treinado para separar uma das classes das outras. Portanto, o modelo i é treinado considerando que todos os exemplos da classe i pertencem à classe positiva, enquanto os exemplos das outras classes pertencem à classe negativa. Quando um exemplo de uma classe desconhecida é apresentado, ele é classificado por cada um dos $|C|$ modelos e recebe a classe relativa ao modelo que obtém o melhor resultado.

4.4 MÉTODO DE AVALIAÇÃO DOS CLASSIFICADORES

Para verificar a eficácia dos classificadores quando recebem as diferentes representações, é necessária uma métrica que consiga quantificar o quão bom é um classificador para a tarefa em que ele está sendo exposto. Este trabalho utiliza a métrica F1-score para avaliar os experimentos de classificação de documentos. F1-score é baseada na combinação de outras duas medidas descritas a seguir: *Precision* e *Recall*.

O Quadro 4 apresenta o exemplo de uma matriz de confusão de duas classes (Classe 1 e Classe 2):

Quadro 4 – Exemplo de matriz de confusão para duas classes

	Classe 1	Classe 2
Classe 1	Verdadeiro Positivo (VP)	Falso Positivo (FP)
Classe 2	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Fonte – Elaborado pelo autor

Precision é definido como a quantidade de documentos classificados como pertencentes à classe C_i que realmente pertencem à classe C_i. O valor de *Precision* é definido como:

$$P = \frac{VP}{(VP + FP)} \tag{4.1}$$

Já o valor de *Recall* é definido como a quantidade de documentos de uma classe C_i identificados corretamente. O valor de *Recall* é definido como:

$$P = \frac{VP}{(VP + FN)} \tag{4.2}$$

F1-score é uma média ponderada de *Precision* e *Recall* definida por:

$$F1\text{-}score = \frac{2 * P * R}{P + R} \tag{4.3}$$

Com base nesses conceitos, observa-se que cada métrica avalia individualmente cada classe, sendo necessária uma medida de avaliação que abranja todo o conjunto de classes. Para isso, tem-se o conceito de micro-média que corresponde a média de cada métrica. Para o *F1-score*, a *F1-micro* é definida como:

$$F1\text{-}micro = \frac{\sum_{i=1}^{n} F1_{c_i}}{n} \tag{4.4}$$

onde n é a quantidade de classes do conjunto de amostras.

Com as métricas de avaliação definidas, é necessário um método de seleção de amostras (documentos) para conjuntos de treino e teste das coleções de documentos e para seleção de modelos.

O procedimento clássico recomendado na avaliação de classificadores em Reconhecimento de Padrões é o seguinte: os dados disponíveis são separados aleatoriamente em conjunto de projeto (*design*) e conjunto de validação. No conjunto de projeto pode-se utilizar validação cruzada para ajuste de parâmetros dos algoritmos, chamado de seleção de modelo. O desempenho do modelo assim selecionado é então avaliado utilizando o conjunto de validação. Esse procedimento é repetido 10 vezes, por exemplo, e assim obtém-se média e variância dos índices avaliados. Entretanto, neste trabalho, a maioria dos parâmetros das representações de documentos e dos classificadores foi seguindo as recomendações da literatura ou realizando uma pesquisa em *grid* com poucos valores. Assim, já tendo esses parâmetros dos modelos selecionados, utilizou-se validação cruzada diretamente na avaliação final.

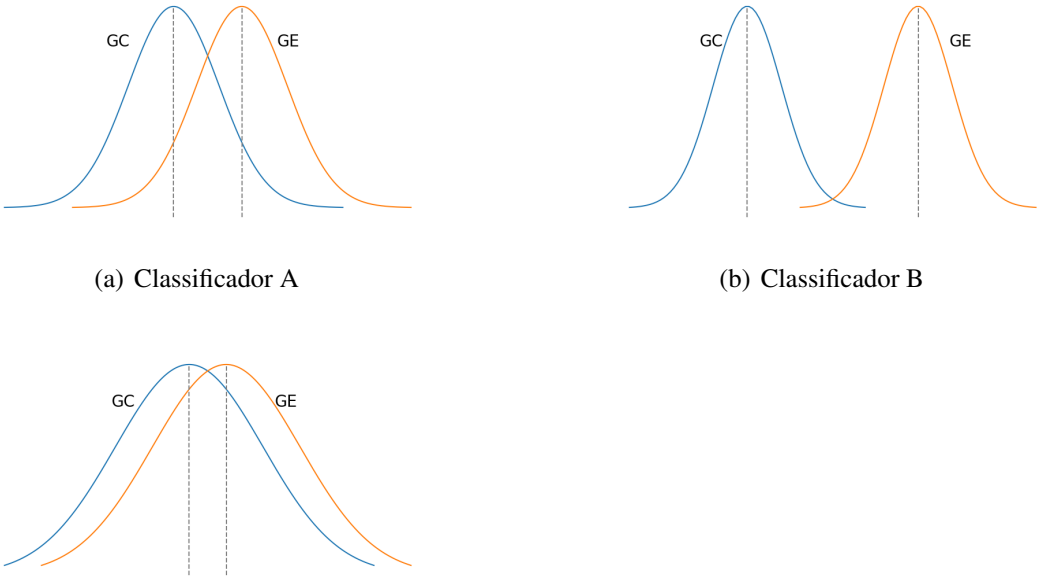
A validação cruzada (*cross-validation*) é uma técnica que visa na validação de modelos com base na capacidade de generalização do classificador. Baseia-se na divisão de um conjunto de amostras, de modo que uma parte seja usada para estimação dos parâmetros na etapa de treinamento, e outra parte seja utilizada para avaliar a etapa de treinamento. Três métodos são utilizados: *holdout*, o *k-fold* e o *leave-one-out*. Nos experimentos realizados foi utilizado o *k-fold*.

O *k-fold* divide o conjunto de amostras em k sub-conjuntos de mesmo tamanho. O classificador é executado k vezes, onde cada um desses k sub-conjuntos é utilizado para teste, enquanto os $k-1$ sub-conjuntos restantes são usados para treinamento. Ao final, toma-se a média e o desvio padrão dos k resultados.

4.5 TESTES ESTATÍSTICOS

O desempenho de algoritmos de classificação é dependente dos dados de entrada. Assim sendo, o desempenho médio é uma variável aleatória. Como tal, experimentos de comparação do desempenho de algoritmos de classificação exigem que testes estatísticos de hipótese sejam verificados. Essa necessidade pode ser entendida observando a Figura 10.

Figura 10 – O efeito da variância dos classificadores na comparação de desempenho



(a) Classificador A

(b) Classificador B

(c) Classificador C

Fonte – Elaborado pelo autor

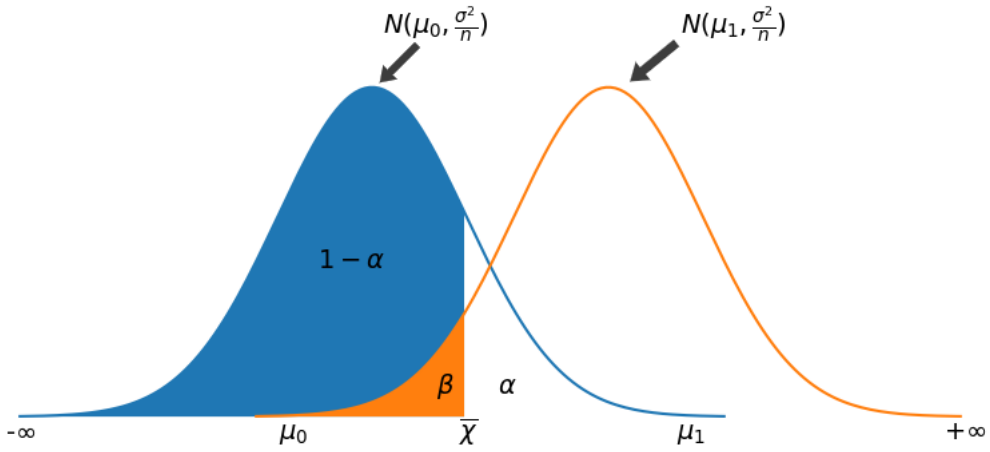
Nela vemos três situações nas quais o classificador GE apresenta média de desempenho maior que a do classificador GC. Entretanto, as variâncias são muito diferentes. Na Figura 10(a) os classificadores tem menor variância enquanto que na Figura 10(b) eles tem variância 10 vezes maior. Na Figura 10(c) GE é claramente superior a GC enquanto que na Figura 10(b) embora as médias sejam diferentes não diferença estatística significativa, e na 10(a) isso não é imediatamente garantido.

Os teste estatísticos de hipótese podem ser paramétricos e não paramétricos. Exem-

plos de testes paramétricos são os testes de hipótese normal padrão e o teste *t-student* e do segundo é o teste de *Wilcoxon*. Neste trabalho adotou-se a hipótese de normalidade e utilizou-se o teste paramétrico com 5% de confiança.

A Figura 11 apresenta duas distribuições com médias μ_0 e μ_1. Tomando como base duas hipótese H_0: $\mu = \mu_0$ e H_1: $\mu > \mu_0$, o chamado erro β consiste na possibilidade de se selecionar uma amostra de uma população com média $\mu_1 (> \mu_0)$ e obter $\overline{\chi}$ de forma que leve a conclusão errada de que H_0 é verdadeira, ou seja P(aceitar H_0/H_1 é verdadeira) = β e P(rejeitar H_0/H_1 é verdadeira) = $1 - \beta$.

Figura 11 – Distribuições com médias μ_0 e μ_1



Fonte – Elaborado pelo autor

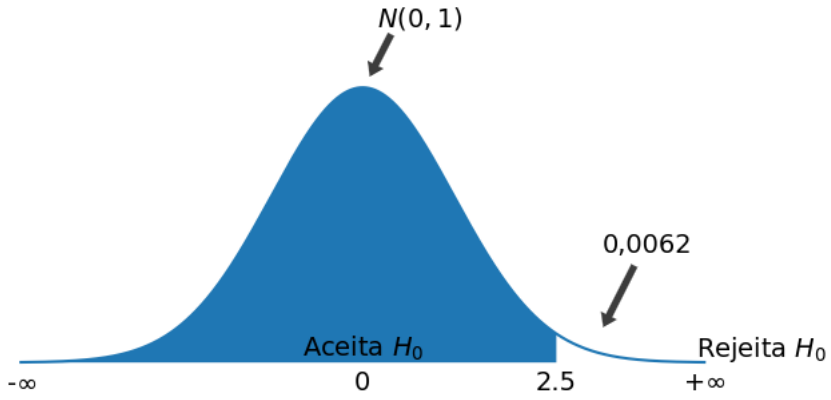
A estatística calculada é o *p-value*. A hipótese nula é a de que as médias são iguais contra a hipótese alternativa de valores diferentes. Sempre que o *p-value* for menor 0.05 rejeita-se a hipótese nula e o algoritmo com maior média de F1-score terá de fato um desempenho estatisticamente superior. O *p-value* mede a probabilidade do resultado ser pelo menos tão extremo quanto teste estatístico, assumindo que a hipótese nula é verdadeira.

Toda conclusão de um teste de hipótese está associada a um nível de significância. A Figura 12 apresenta um nível de significância para *p-value*. Onde, dado um teste z unilateral a 5% de significância, pode-se concluir que a média μ é maior que 20 uma vez que a estatística z obtida foi de 2.5 ($z = 1.645$)

Para $\alpha = 1\%$, ainda assim, rejeita-se H_0, ou seja, μ é maior que 20. Pode-se aceitar

H_0 para qualquer nível de significância (α) menor que 0.0062 uma vez que *p-value* = P(Z>2.5) = 0.0062

Figura 12 – Nível de significância para *p-value*



Fonte – Elaborado pelo autor

4.6 RECURSOS COMPUTACIONAIS

4.6.1 Hardware

- CPU Intel Core i7 (12 CPUs) 2.2 GHz;
- 16 GB de memória RAM.

4.6.2 Software

- Sistema Operacional Windows 10;
- Linguagem de programação Python, versão 3.6.4;
- Biblioteca NumPy, versão 1.15.4, para computação numérica;
- Biblioteca Gensim, versão 3.6.0 para implementação do *Word2Vec* e LDA;
- Biblioteca Scikit-Learn, versão 0.20.2 para implementação de classificadores e métricas de avaliação;
- Biblioteca NLTK, versão 3.4.0 para pré-processamento de texto;
- Biblioteca Scipy, versão 1.2.0 para computação numérica;
- Biblioteca Matplotlib, versão 3.0.2, para geração de gráficos.

5 RESULTADOS

Neste capítulo são apresentados os resultados dos experimentos propostos para avaliação de textos curtos e muito curtos. A notação seguida para denominar cada variante do experimento é REPRESENTAÇÃO_CLASSIFICADOR, como por exemplo: LDA-10_SVM. A notação de cada REPRESENTAÇÃO é dada da seguinte forma. Para representações por *Word Embedding*, **WV** significa o algoritmo *Word2Vec* treinado na coleção e **GLO** significa o algoritmo *Glove* treinado com o corpus da *Wikipedia*. A abordagem **MEAN-X** corresponde à representação de *Word Embedding* pela média, onde *X* significa o algoritmo para obter *Word Embedding*. Para representações por LDA é dado a notação **LDA-X**, onde *X* significa o número de tópicos extraídos. Para a representação CTE a notação é **CTE-X-Y**, onde *X* significa o número de tópicos extraídos e *Y* o método usado para obter *Word Embedding*. Entende-se por variação de representação a abordagem que apresenta o mesmo classificador, mesmo método para obter *Word Embedding* (se houver) contudo apresenta variações no número de tópicos, como por exemplo em CTE e LDA. O TF-IDF passa a ser **TFIDF** para seguir o padrão da notação.

São avaliados 30 métodos (REPRESENTAÇÃO_CLASSIFICADOR) para as 4 coleções de documentos totalizando 120 execuções, além de um pequeno experimento em bases próprias de classificação de notícias muito curtas (título e lide) com 3 representações para cada base totalizando mais 6 experimentos.

São apresentadas tabelas com os resultados para cada coleção de documentos, onde para cada variante é apresentada a média das 10 medidas *F1-micro*, assim como a variância, o desvio padrão e a dimensão vetorial dessas medidas. Abordagens que apresentam variação aparecem de maneira agrupada, onde é destacado em negrito o melhor resultado da variação. Em abordagens que não apresentam variação é destacado em negrito o melhor resultado da abordagem. Quando o melhor resultado aparece mais de uma vez na mesma representação ou variação é destacado o de menor dimensão, e caso a dimensão seja igual, todos são destacados.

5.1 CLASSIFICAÇÃO DE NOTÍCIAS

Os experimentos de classificação de notícias foram realizados sobre as coleções de documentos *20Newsgroups, RCV-1, Z5News e Z5NewsBrasil*.

5.1.1 Resultados e Avaliação *20Newsgroups*

A Tabela 4 apresenta os resultados de *F1-micro* para a coleção de documentos *20Newsgroups*. Dela pode-se extrair as seguintes observações.

Tabela 4 – Resultados da métrica *F1-Score* - (Média, Variância (VAR), Desvio Padrão (STD) e Dimensão) para coleção de documentos *20Newsgroups*

Representação-Classificador	*F1-micro*	VAR	STD	Dimensão
TFIDF_SVM	0.72	0.00077	0.02769	90211
TFIDF_NB	**0.76**	0.00006	0.00759	90211
MEAN-GLO_SVM	**0.55**	0.00012	0.01076	100
MEAN-WV_SVM	0.43	0.00016	0.01268	300
MEAN-GLO_NB	0.49	0.00014	0.01188	100
MEAN-WV_NB	0.20	0.00004	0.00598	300
LDA-10_SVM	0.07	0.00043	0.02082	10
LDA-50_SVM	0.15	0.00008	0.00903	50
LDA-100_SVM	0.17	0.00236	0.04855	100
LDA-200_SVM	**0.19**	0.00012	0.01093	200
LDA-10_NB	0.17	0.00003	0.00569	10
LDA-50_NB	**0.18**	0.00281	0.05303	50
LDA-100_NB	0.15	0.00008	0.00890	100
LDA-200_NB	0.14	0.00003	0.00544	200
CTE-10-GLO_SVM	0.41	0.00012	0.01115	36
CTE-50-GLO_SVM	**0.53**	0.00012	0.01076	169
CTE-100-GLO_SVM	0.53	0.00016	0.01253	383
CTE-200-GLO_SVM	0.53	0.00015	0.01231	969
CTE-10-WV_SVM	0.34	0.00007	0.00860	36
CTE-50-WV_SVM	**0.35**	0.00009	0.00938	180
CTE-100-WV_SVM	0.34	0.00005	0.00700	405
CTE-200-WV_SVM	0.33	0.00028	0.01681	1011
CTE-10-GLO_NB	0.23	0.00016	0.01279	36
CTE-50-GLO_NB	**0.28**	0.00007	0.00817	169
CTE-100-GLO_NB	0.22	0.00004	0.00661	383
CTE-200-GLO_NB	0.19	0.00007	0.00818	969
CTE-10-WV_NB	**0.26**	0.00009	0.00946	36
CTE-50-WV_NB	0.18	0.00007	0.00853	180
CTE-100-WV_NB	0.15	0.00006	0.00748	405
CTE-200-WV_NB	0.14	0.00003	0.00589	1011

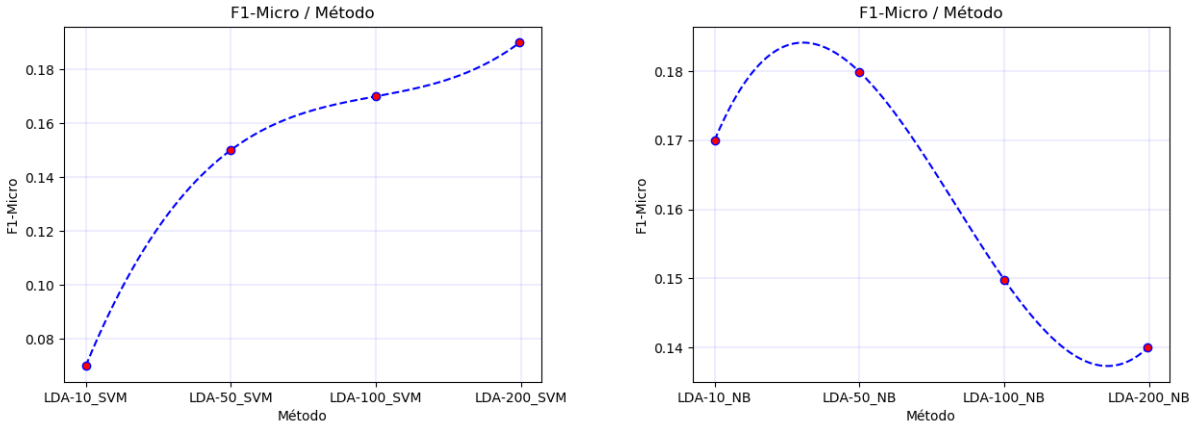
Fonte – Elaborado pelo autor

Primeiramente, observa-se que a representação do TF-IDF (com os dois classificadores) obteve os melhores resultados, destacando-se o TFIDF_NB com *F1-micro* de 0.76% como melhor resultado e TFIDF_SVM com *F1-micro* de 0.72% como segundo melhor valor.

A Figura 13 apresenta graficamente a variação do LDA. Quando o SVM é utilizado como classificador, observa-se uma melhoria crescente no resultado de *F1-micro* quando o

número de tópicos cresce (com 10 tópicos *F1-micro* de 0.07%, 50 tópicos *F1-micro* de 0.15%, 100 tópicos *F1-micro* de 0.17% e 200 tópicos *F1-micro* de 0.19%). Com 200 tópicos extraídos, LDA-200_SVM alcançou 0.19% de *F1-micro*, o melhor resultado das variações da representação via LDA. Já quando o classificador *Naive Bayes* (NB) foi utilizado, observa-se um crescimento quando o número de tópicos passou de 10 para 50 (com 10 tópicos *F1-micro* de 0.17% e 50 tópicos *F1-micro* de 0.18%), e uma queda no resultado quando o número de tópicos passou de 50 para 100 e 200 (com 100 tópicos *F1-micro* de 0.15% e 200 tópicos *F1-micro* de 0.14%).

Figura 13 – Desempenho LDA - *20Newsgroups*



(a) LDA - SVM

(b) LDA - *Naive Bayes*

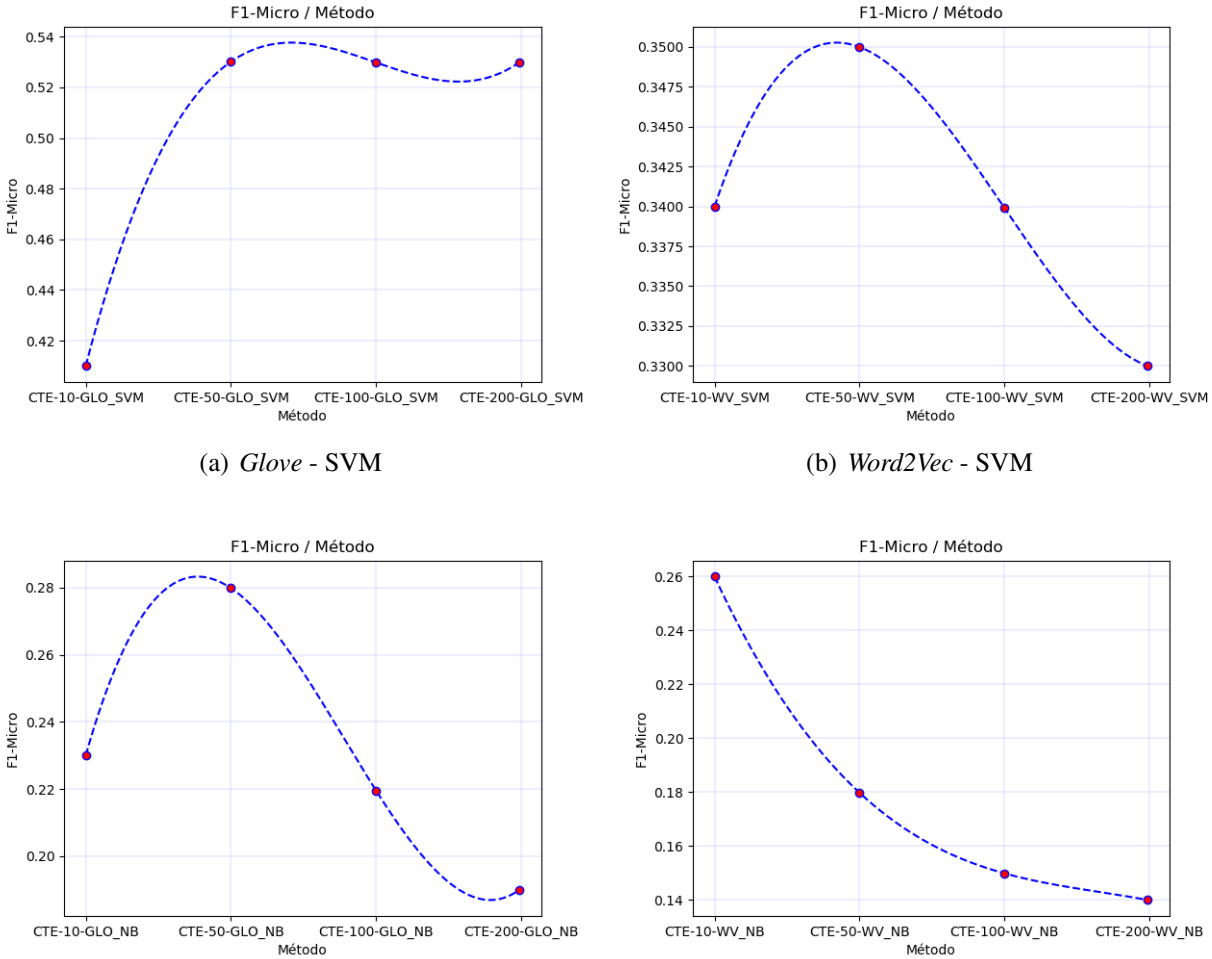
Fonte – Elaborado pelo autor

Em relação a média de *Word Embedding*, a abordagem que utiliza o algoritmo *Glove* treinado no corpus da *Wikipedia* alcançou o melhor resultado em relação ao algoritmo *Word2Vec* treinado no corpus da coleção. MEAN-GLO_SVM atingiu 0.55% de *F1-micro*, e MEAN-GLO_NB atingiu o segundo melhor resultado com 0.49% de *F1-micro*. Com relação aos algoritmos de classificação, observa-se que houve uma melhoria de 115% do resultado do SVM para o *Naive Bayes*, quando a abordagem que usa *Word2Vec* treinado na coleção de documento é utilizada. Já quando o *Glove* é utilizado, a melhoria em usar o SVM em vez do *Naive Bayes* é de aproximadamente 12%.

A Figura 14 apresenta graficamente a variação do CTE. Observa-se que a variação que utiliza *Glove* CTE-50-GLO-SVM obteve o melhor resultado em relação a todos os experimentos do CTE com 0.53% de *F1-micro*. Nessa combinação *Glove* e SVM observa-se uma melhoria no resultado quando o número de tópicos passa de 10 para 50. O resultado mantém-se constante quando o número de tópicos é aumentado para 100 e 200 com 0.53% de *F1-micro* para ambos. Para os resultados que utilizam o *Word2Vec* com o SVM, observa-se uma melhoria

quando o número de tópicos passa de 10 para 50 alcançando 0.35% de *F1-micro* com 50 tópicos, e uma queda no resultado quando o número de tópicos é 100 ou 200 (0.34% e 0.33% de *F1-micro* respectivamente). Quando o *Naive Bayes* é utilizado com o CTE, o melhor resultado é quando o algoritmo *Glove* é utilizado: CTE-50-GLO-NB com 0.28% de *F1-micro*. Detalhando essa combinação *Glove* e *Naive Bayes*, observa-se que ocorre uma melhoria no resultado quando o número de tópicos passa de 10 para 50 (0.23% e 0.28% de *F1-micro* respectivamente), e uma queda nos resultados quando o número de tópicos extraídos é aumentado, 100 e 200 tópicos (0.22% e 0.19% de *F1-micro* respectivamente). Já quando o *Word2Vec* é utilizado com o *Naive Bayes*, observa-se uma queda nos resultados à medida que o número de tópicos cresce, com o melhor resultado com 10 tópicos CTE-10-WV-NB de 0.26% de *F1-micro*.

Figura 14 – Desempenho CTE - *20Newsgroups*



(a) *Glove* - SVM

(b) *Word2Vec* - SVM

(c) *Glove - Naive Bayes*

(d) *Word2Vec - Naive Bayes*

Fonte – Elaborado pelo autor

Em relação as duas representações que utilizam *Word Embedding*, CTE e MEAN,

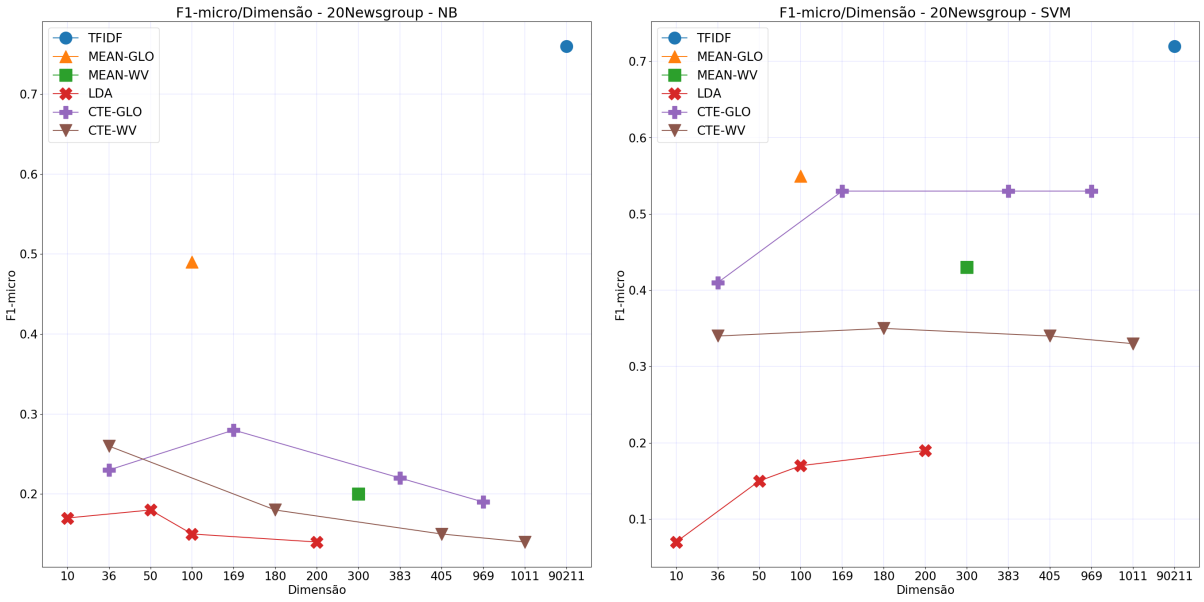
pode-se observar uma coisa comum entre as duas abordagens. O melhor resultado em cada uma é quando o algoritmo *Glove* é utilizado. O melhor resultado para o *Word2Vec* na representação MEAN é de 0.43%, onde o resultado para o *Glove* corresponde à melhoria de 0.27%. Já o melhor resultado para o *Word2Vec* na representação CTE é de 0.35%, onde o resultado para o *Glove* corresponde à uma melhoria de 0.51%. Isso pode ser interpretado como o quanto *Glove* explora mais contextos em ser treinado na *Wikipedia*, enquanto os contextos explorados pelo *Wordvec*, treinado apenas na coleção de documentos, apresentam limitações.

No que diz respeito ao desempenho de cada classificador em relação à dimensão de cada representação, a Figura 15 apresenta a variação entre *F1-micro* (eixo Y) e a dimensão de cada abordagem (eixo X) para os classificadores SVM e *Naive Bayes*. Abordagens que apresentem variações são apresentadas como símbolos interligados (LDA, CTE-GLO, CTE-WV). Enquanto abordagens que não apresentem variações são apresentadas apenas como um símbolo isolado (TF-IDF, MEAN-GLO, MEAN-WV). Uma vez que o CTE apresenta variações com *Glove* e *Word2Vec*, para representar as palavras tópicas do método, a dimensão em alguns casos apresentou uma diferença. Por exemplo, com 10 tópicos o CTE apresentou a mesma dimensão para *Glove* e *Word2Vec* (36), enquanto com 50 tópicos apresentou dimensão (169-*Glove* e 180-*Word2Vec*), com 100 (383-*Glove* e 405-*Word2Vec*) e 200 tópicos (969-*Glove* e 1011-*Word2Vec*). Essa diferença deve-se ao fato de que o *Word2Vec* não encontra uma representação para todas as palavras da coleção de documentos, enquanto o *Glove* contém mais palavras representadas.

Em relação ao SVM, o TF-IDF é o que apresenta melhor resultado de *F1-micro* (0.72%) porém com a dimensão mais alta 90211. MEAN-GLO o segundo melhor *F1-micro* (0.55%) com dimensão baixa de 100, e a abordagem CTE-50-GLO alcançou *F1-micro* (0.53%) com dimensão de tamanho 169. O CTE-50-GLO alcançou este resultado com uma dimensão maior que MEAN-GLO e aproximadamente 533 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o menor resultado com 0.07% de *F1-micro*. Em relação ao *Naive Bayes*, o TF-IDF é o que apresenta melhor resultado (0.76%). MEAN-GLO o segundo melhor *F1-micro* (0.49%) com dimensão de 100, e a abordagem CTE-50-GLO alcançou *F1-micro* (0.28%) com dimensão de tamanho 169. O CTE-50-GLO alcançou este resultado com uma dimensão maior que MEAN-GLO e aproximadamente 533 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.17% de *F1-micro*.

Testes estatísticos foram aplicados quando há dúvidas sobre qual combinação tem melhor performance. Nesta tabela essa situação ocorre quando comparando CTE-50-GLO_SVM com MEAN-GLO_SVM e TFIDF_SVM com TFIDF_NB. Foram aplicados testes *t-student* e

Figura 15 – Relação F1-score/Dimensão - *20Newsgroups*



(a) *Naive Bayes* (b) SVM

Fonte – Elaborado pelo autor

em ambos os casos a hipótese nula é rejeitada com *p-value* = 0.0000304 e *p-value* = 0.0002954, respectivamente.

Síntese do experimento: Nota-se destes resultados que, para estes textos curtos, a representação de alta dimensionalidade TF-IDF com classificador não linear SVM supera todas as demais, superando CTE em pelo menos 0.22% de *F1-micro*. Por outro lado, CTE treinado com *Word Embedding* em corpus externos (GLO) supera as demais representações *Word Embedding* puras, como *Word2Vec* e o próprio *Glove*. Embora com *F1-micro* menor, a redução de dimensionalidade de CTE é da ordem de 1000 vezes.

5.1.2 Resultados e Avaliação RCV-1

A Tabela 5 apresenta os resultados de *F1-micro* para a coleção de documentos RCV-1.

Em relação ao melhor resultado observa-se que a representação do TF-IDF obteve, mais uma vez, o melhor resultado com o SVM: TFIDF_SVM com *F1-micro* de 0.82%.

A Figura 16 apresenta graficamente a variação do LDA. Quando o SVM é utilizado como classificador, observa-se um crescimento quando o número de tópicos passou de 10 para 50 (com 10 tópicos *F1-micro* de 0.4% e 50 tópicos *F1-micro* de 0.55%), e uma queda no resultado quando o número de tópicos passou de 50 para 100 e 200 (com 100 tópicos *F1-micro* de 0.52% e

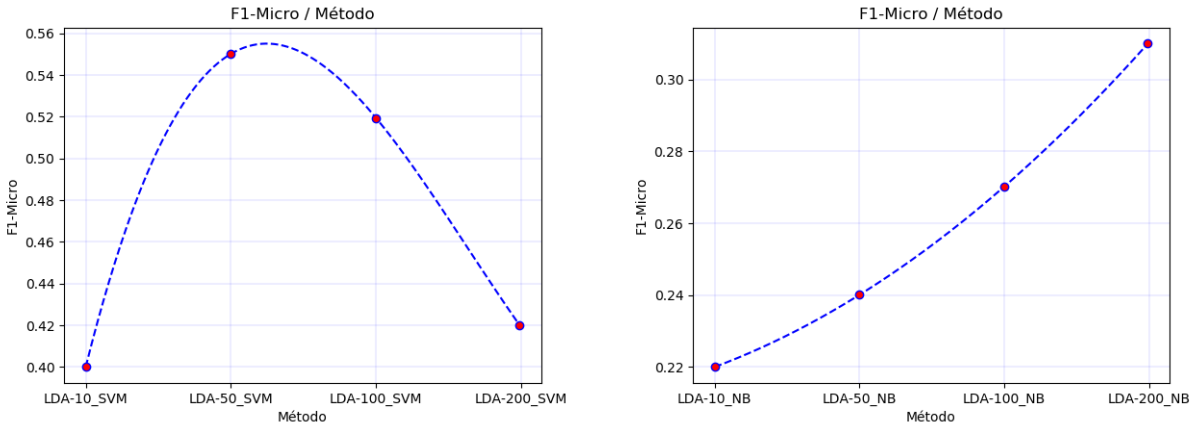
Tabela 5 – Resultados da métrica *F1-Score* - (Média. Variância. Desvio Padrão e Dimensão) para coleção de documentos RCV-1

Representação-Classificador	*F1-micro*	VAR	STD	Dimensão
TFIDF_SVM	**0.82**	0.00005	0.00440	56269
TFIDF_NB	0.51	0.00029	0.01696	56269
MEAN-GLO_SVM	0.60	0.00020	0.01404	100
MEAN-WV_SVM	**0.67**	0.00022	0.01476	300
MEAN-GLO_NB	0.36	0.00008	0.00877	100
MEAN-WV_NB	0.25	0.00022	0.01486	300
LDA-10_SVM	0.40	0.00016	0.01275	10
LDA-50_SVM	**0.55**	0.00013	0.01142	50
LDA-100_SVM	0.52	0.00016	0.01266	100
LDA-200_SVM	0.42	0.00014	0.01201	200
LDA-10_NB	0.22	0.00005	0.00732	10
LDA-50_NB	0.24	0.00003	0.00591	50
LDA-100_NB	0.27	0.00004	0.00624	100
LDA-200_NB	**0.31**	0.00003	0.00560	200
CTE-10-GLO_SVM	0.48	0.00035	0.01861	49
CTE-50-GLO_SVM	**0.49**	0.00028	0.01679	230
CTE-100-GLO_SVM	0.49	0.00025	0.01588	451
CTE-200-GLO_SVM	0.49	0.00026	0.01626	831
CTE-10-WV_SVM	0.51	0.00031	0.01752	57
CTE-50-WV_SVM	0.54	0.00031	0.01757	276
CTE-100-WV_SVM	0.54	0.00014	0.01163	552
CTE-200-WV_SVM	**0.55**	0.00017	0.01308	1038
CTE-10-GLO_NB	**0.20**	0.00003	0.00505	49
CTE-50-GLO_NB	0.19	0.00002	0.00490	230
CTE-100-GLO_NB	0.19	0.00002	0.00473	451
CTE-200-GLO_NB	0.20	0.00003	0.00530	831
CTE-10-WV_NB	**0.23**	0.00005	0.00714	57
CTE-50-WV_NB	0.21	0.00004	0.00640	276
CTE-100-WV_NB	0.21	0.00004	0.00608	552
CTE-200-WV_NB	0.21	0.00004	0.00630	1038

Fonte – Elaborado pelo autor

200 tópicos *F1-micro* de 0.42%). Com 50 tópicos extraídos, LDA-50_SVM alcançou 0.55% de *F1-micro*, o melhor resultado das variações da representação via LDA. Já quando o classificador *Naive Bayes* (NB) foi utilizado, observa-se uma melhoria crescente no resultado de *F1-micro* quando o número de tópicos cresce (com 10 tópicos *F1-micro* de 0.22%, 50 tópicos *F1-micro* de 0.24%, 100 tópicos *F1-micro* de 0.27% e 200 tópicos *F1-micro* de 0.31%).

Figura 16 – Desempenho LDA - RCV-1



(a) *LDA* - SVM

(b) *LDA* - *Naive Bayes*

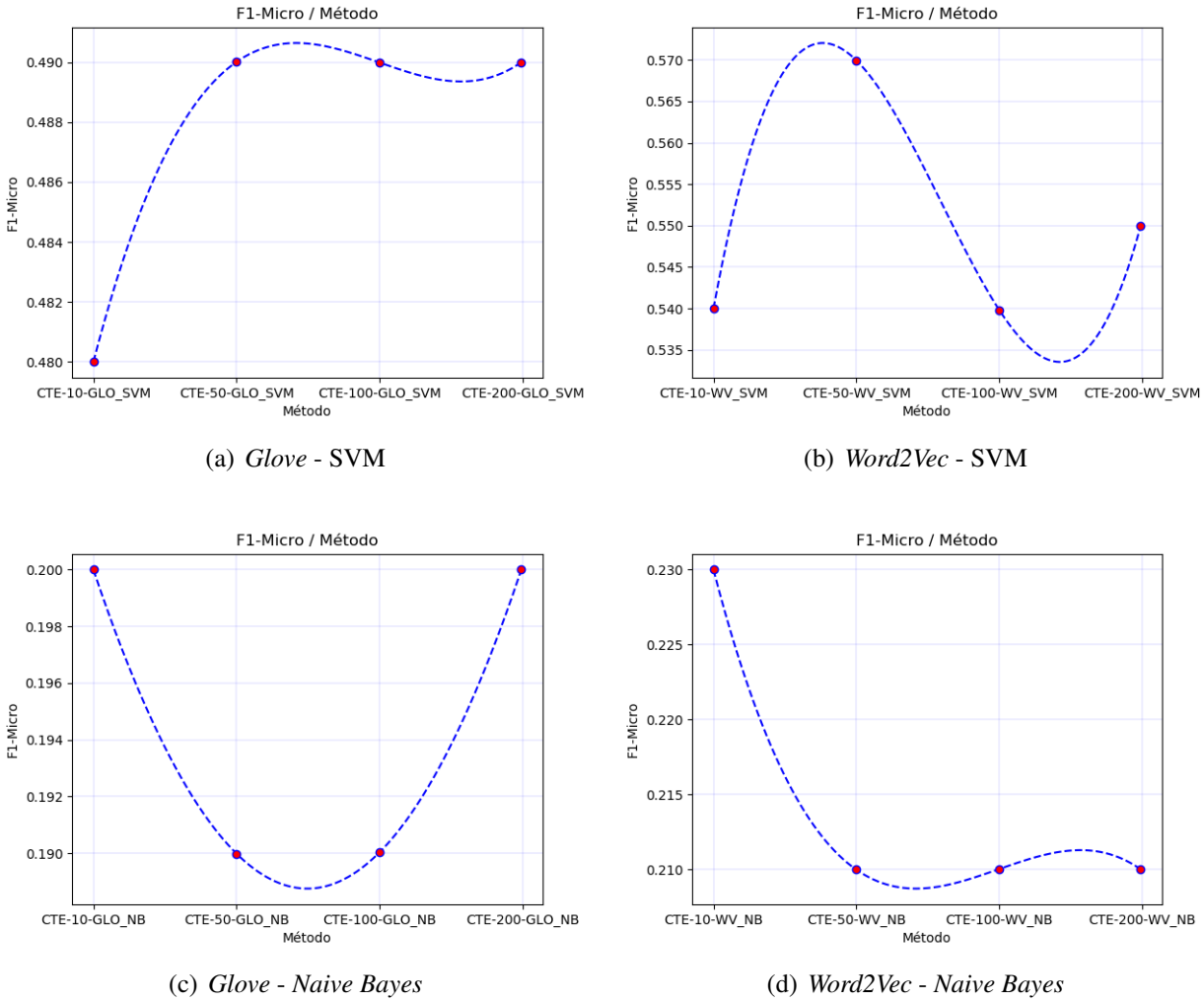
Fonte – Elaborado pelo autor

Em relação a média de *Word Embedding*, a abordagem que utiliza o algoritmo *Word2Vec* treinado na própria coleção alcançou melhor resultado em relação ao algoritmo *Glove* treinado com o corpus da *Wikipedia*. MEAN-WV_SVM atingiu 0.67% de *F1-micro*, e MEAN-GLO_SVM atingiu 0.6% de *F1-micro*. Com relação aos algoritmos de classificação, observa-se que houve uma melhoria com o dobro do resultado do SVM para o *Naive Baye*s, quando a abordagem que usa *Word2Vec* é utilizada. Já quando o *Glove* é utilizado, a melhoria em usar o SVM em vez do *Naive Bayes* é de aproximadamente 67%.

A Figura 17 apresenta graficamente a variação do CTE. Observa-se que a variação que utiliza *Word2Vec* CTE-200-WV_SVM obteve o melhor resultado em relação a todos os experimentos do CTE com 0.55% de *F1-micro*. Nessa combinação *Word2Vec* e SVM observa-se uma melhoria no resultado com o número de tópicos passa de 10 para 50 (0.51% e 0.54% de *F1-micro* respectivamente), da mesma maneira quando o número de tópicos passa de 100 para 200 com (0.54% e 0.55% de *F1-micro* respectivamente). Para os resultado que utilizam o *Glove* com o SVM, observa-se uma melhoria quando o número de tópicos passa de 10 para 50 alcançando 0.49% de *F1-micro* com 50 tópicos, e mantém-se constante quando o número de tópicos é 100 ou 200 com 0.49% de *F1-micro* para ambos. Quando o *Naive Bayes* é utilizado

com o CTE, o melhor resultado é quando o algoritmo *Word2Vec* é utilizado: CTE-10-WV-NB com 0.23% de *F1-micro*. Detalhando essa combinação *Word2Vec* e *Naive Bayes*, observa-se que ocorre uma queda no resultado quando o número de tópicos passa de 10 para 50 (0.23% e 0.21% de *F1-micro* respectivamente), e mantém-se constante quando o número de tópicos é 100 ou 200 com 0.21% de *F1-micro* para ambos. Já quando o *Glove* é utilizado com o *Naive Bayes* observa-se que ocorre uma queda no resultado quando o número de tópicos passa de 10 para 50 (0.20% e 0.19% de *F1-micro* respectivamente), e um crescimento no resultado quando o número de tópicos passa para 100 e 200 (0.19% e 0.20% de *F1-micro* respectivamente).

Figura 17 – Desempenho CTE - RCV-1



(a) *Glove* - SVM

(b) *Word2Vec* - SVM

(c) *Glove - Naive Bayes*

(d) *Word2Vec - Naive Bayes*

Fonte – Elaborado pelo autor

Em relação as duas reapresentações que utilizam *Word Embedding* CTE e MEAN, o melhor resultado em cada uma é quando o algoritmo *Word2Vec* treinado no corpus da coleção. O melhor resultado para o *Glove* na representação MEAN é de 0.6%, onde o resultado para

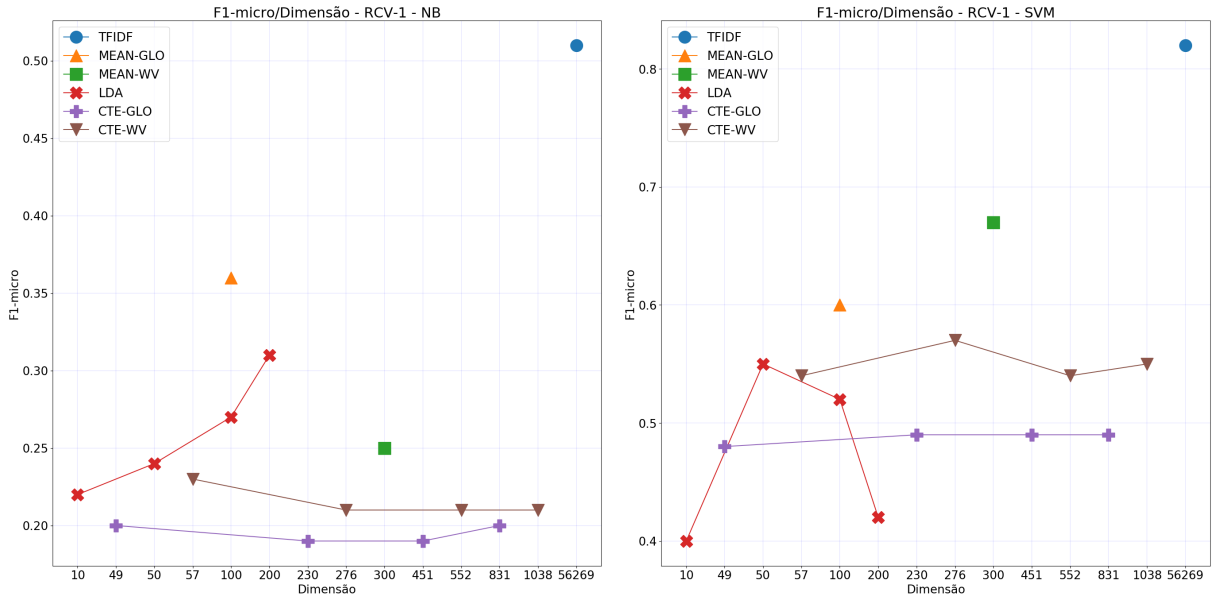
o *Word2Vec* corresponde a melhoria de 0.11%. Já o melhor o resultado para o *Glove* na representação CTE é de 0.49%, onde o resultado para o *Word2Vec* corresponde a uma melhoria de 0.12%. Diferente do que ocorreu na coleção *20Newsgroups*, onde o *Glove* teve um resultado melhor que o *Word2Vec*, essa vantagem do *Word2Vec* em relação ao *Glove* pode ser interpretada como a representatividade da coleção de documentos RCV-1 para definir e explorar os contextos para representar cada palavra.

No que diz respeito ao desempenho de cada classificador em relação à dimensão de cada representação, a Figura 18 apresenta a variação entre *F1-micro* (eixo Y) e a dimensão de cada abordagem (eixo X) para os classificadores SVM e *Naive Bayes*. Abordagens que apresentem variações são apresentadas como símbolos interligados (LDA, CTE-GLO, CTE-WV). Enquanto abordagens que não apresentem variações são apresentadas apenas como um símbolo isolado (TF-IDF, MEAN-GLO, MEAN-WV). Uma vez que o CTE apresenta variações com *Glove* e *Word2Vec* para representar as palavras tópicas do método, a dimensão em todos os casos apresentou uma diferença. Por exemplo, com 10 tópicos apresentou dimensão (49-*Glove* e 57-*Word2Vec*), 50 tópicos (230-*Glove* e 276-*Word2Vec*), 100 tópicos (451-*Glove* e 552-*Word2Vec*) e 200 tópicos (831-*Glove* e 749-*Word2Vec*). Essa diferença deve-se ao fato de que o corpus treinado na *Wikipedia* não contenha todos os termos presentes na coleção de documentos.

Em relação ao SVM, o TF-IDF é o que apresenta melhor resultado de *F1-micro* (0.82%) porém com a dimensão mais alta 56269. MEAN-WV o segundo melhor *F1-micro* (0.67%) com dimensão 300, e a abordagem CTE-200-WV alcançou *F1-micro* (0.55%) com dimensão de tamanho 1038. O CTE-200-WV alcançou este resultado com uma dimensão maior que MEAN-GLO e aproximadamente 54 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.4% de *F1-micro*. Em relação ao *Naive Bayes*, o TF-IDF é o que apresenta melhor resultado (0.51%). MEAN-GLO o segundo melhor *F1-micro* (0.36%) com dimensão de 100, e a abordagem CTE-10-WV alcançou *F1-micro* (0.23%) com dimensão de tamanho 57. O CTE-10-WV alcançou este resultado com uma dimensão menor que MEAN-GLO e aproximadamente 987 vezes menor que o TFIDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.22% de *F1-micro*.

Testes estatísticos foram aplicados quando há dúvidas sobre qual combinação tem melhor performance. Nesta tabela essa situação ocorre quando comparando CTE-10-GLO_SVM com CTE-200-GLO_SVM e CTE-10-WV_SVM com MEAN-GLO_SVM. Foram aplicados testes *t-student* no primeiro caso a hipótese nula não é rejeitada com *p-value* = 0.0843994, no segundo caso a hipótese nula é rejeitada com *p-value* = 0.00000002.

Figura 18 – Relação F1-score/Dimensão - RCV1



(a) *Naive Bayes* (b) SVM

Fonte – Elaborado pelo autor

Síntese do experimento: Nota-se destes resultados que, para estes textos curtos, a representação de alta dimensionalidade TF-IDF com classificador não linear SVM supera todas as demais, superando CTE em pelo menos 0.27% de *F1-micro*. Por outro lado, CTE treinado com *Word Embedding* da própria coleção (interna) (WV) não supera as demais representações *Word Embedding* puras, como *Word2Vec* e *Glove*. Embora com *F1-micro* menor, a redução de dimensionalidade de CTE é da ordem de 1000 vezes.

5.1.3 Z5News e Z5NewsBrasil

Para as coleções próprias foi executado a abordagem CTE com 10 tópicos nas duas variações (*Word2Vec* e *Glove*) com o algoritmo SVM, contudo para a coleção Z5NewsBrasil foi obtido *Word Embedding* em português treinadas previamente com o corpus da *Wikipedia* em PT-BR através do *Word2Vec*. O CTE é comparado com o TF-IDF. A Tabela 6 apresenta a medida de F1-score para a coleção Z5News.

Tabela 6 – Comparativo de combinações para a Coleção Z5News

Método	***F1-micro***
TFIDF_SVM	**0.70**
CTE-10-GLO_SVM	0.53
CTE-10-WV_SVM	0.41

Fonte – Elaborado pelo autor

A Tabela 7 apresenta a medida de F1-score para a coleção Z5NewsBrasil.

Tabela 7 – Comparativo de combinações para a Coleção Z5NewsBrasil

Método	***F1-micro***
TFIDF_SVM	**0.71**
CTE-10-WVPT_SVM	0.60
CTE-10-WV_SVM	0.41

Fonte – Elaborado pelo autor

Síntese destes experimentos: Nota-se destes resultados que, para estes textos curtos, a representação de alta dimensionalidade TF-IDF com classificador não linear SVM supera o CTE em pelo menos 0.17% e 0.11% de *F1-micro*.

5.2 CLASSIFICAÇÃO DE SENTIMENTOS

Nos experimentos de classificação de sentimentos, foram avaliados os resultados dos experimentos para as coleções de documentos IMDB e STS.

5.2.1 Resultados e Avaliação IMDB

A Tabela 8 apresenta os resultados para cada REPRESENTAÇÃO_CLASSIFICADOR da coleção de documentos IMDB.

A Figura 19 apresenta graficamente a variação do LDA. O melhor resultado dentre todos os experimentos para a coleção IMDB é quando a representação do TF-IDF (com os dois classificadores) é utilizada. O TFIDF_SVM com *F1-micro* de 0.89% como melhor resultado e TFIDF_NB com *F1-micro* de 0.87% como segundo melhor valor.

Em relação às variações do LDA, observa-se uma pequena variação nos resultados de *F1-micro* para os dois classificadores (dentre os oito experimentos o menor resultado é 0.6% com 50 tópicos utilizando o *Naive Bayes*, e o melhor resultado é 0.65% em dois cenários: um com 50 tópicos utilizando o SVM e outro com 200 tópicos utilizando o *Naive Bayes*.

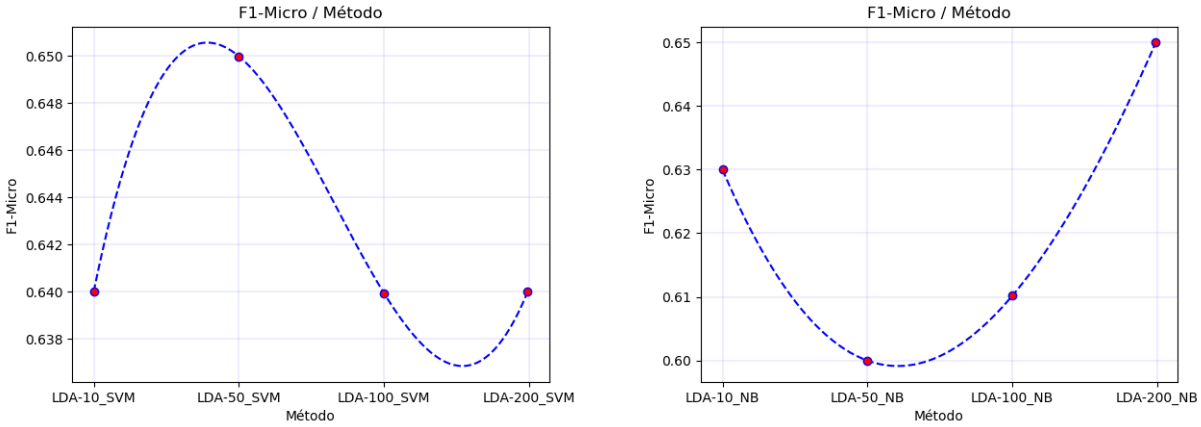
Em relação a média de *Word Embedding*, a abordagem que utiliza o algoritmo *Word2Vec* alcançou um melhor resultado em relação ao algoritmo *Glove*, quando o SVM é utilizado (MEAN-WV_SVM atingiu 0.82% de *F1-micro*). Já quando o *Naive Bayes* é utilizado, a abordagem com o *Glove* é superior ao *Word2Vec* (MEAN-GLO_NB atingiu 0.72% de *F1-micro*). Com relação aos algoritmos de classificação, observa-se que as duas abordagens onde o SVM foi utilizado obtiveram melhores resultados em comparação com a abordagens onde o

Tabela 8 – Resultados da métrica *F1-Score* - (Média. Variância. Desvio Padrão e Dimensão) para coleção de documentos IMDB

Representação-Classificador	*F1-micro*	VAR	STD	Dimensão
TFIDF_SVM	**0.89**	0.00002	0.00440	74486
TFIDF_NB	0.87	0.00008	0.00901	74486
MEAN-GLO_SVM	0.78	0.00015	0.01237	100
MEAN-WV_SVM	**0.82**	0.00009	0.00938	300
MEAN-GLO_NB	0.72	0.00016	0.01280	100
MEAN-WV_NB	0.71	0.00005	0.00680	300
LDA-10_SVM	0.64	0.00008	0.00906	10
LDA-50_SVM	**0.65**	0.00006	0.00796	50
LDA-100_SVM	0.64	0.00008	0.00881	100
LDA-200_SVM	0.64	0.00009	0.00956	200
LDA-10_NB	0.63	0.00010	0.01020	10
LDA-50_NB	0.60	0.00067	0.02583	50
LDA-100_NB	0.61	0.00207	0.04550	100
LDA-200_NB	**0.65**	0.00347	0.05887	200
CTE-10-GLO_SVM	**0.72**	0.00009	0.00952	26
CTE-50-GLO_SVM	0.70	0.00006	0.00795	102
CTE-100-GLO_SVM	0.70	0.00013	0.01122	274
CTE-200-GLO_SVM	0.71	0.00010	0.01015	748
CTE-10-WV_SVM	**0.77**	0.00007	0.00815	26
CTE-50-WV_SVM	0.76	0.00005	0.00727	102
CTE-100-WV_SVM	0.75	0.00005	0.00701	276
CTE-200-WV_SVM	0.75	0.00005	0.00708	749
CTE-10-GLO_NB	**0.65**	0.00015	0.01221	26
CTE-50-GLO_NB	0.64	0.00009	0.00970	102
CTE-100-GLO_NB	0.63	0.00008	0.00891	274
CTE-200-GLO_NB	0.63	0.00010	0.00995	748
CTE-10-WV_NB	**0.66**	0.00008	0.00876	26
CTE-50-WV_NB	0.65	0.00002	0.00436	102
CTE-100-WV_NB	0.62	0.00004	0.00620	276
CTE-200-WV_NB	0.62	0.00004	0.00616	749

Fonte – Elaborado pelo autor

Figura 19 – Desempenho LDA - IMDB



(a) *LDA* - SVM

(b) *LDA - Naive Bayes*

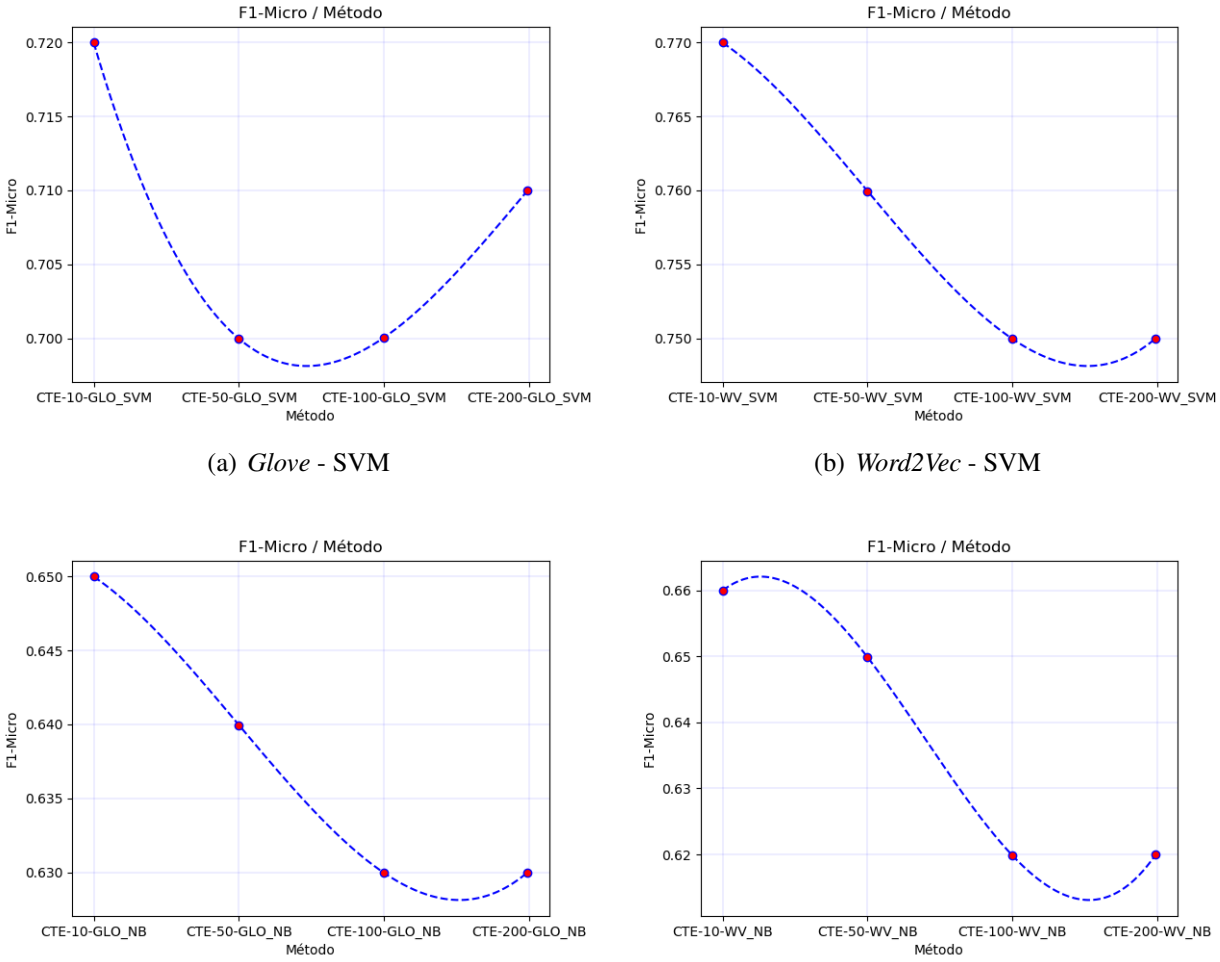
Fonte – Elaborado pelo autor

Naive Bayes foi usado. Houve uma melhoria de 15% do resultado do SVM para o *Naive Bayes*, quando a abordagem que usa *Word2Vec* é utilizada. Já quando o *Glove* é utilizado, a melhoria em usar o SVM em vez do *Naive Bayes* é de aproximadamente 8%.

A Figura 20 apresenta graficamente a variação do CTE. Observa-se que a variação que utiliza *Word2Vec* CTE-10-WV-SVM obteve o melhor resultado em relação a todos os experimentos do CTE com 0.77% de *F1-micro*. Nessa combinação *Word2Vec* e SVM observa-se uma queda no resultado à medida que o número de tópicos cresce de 10 até 100, e mantém-se constante quando o número de tópicos é 100 ou 200 com 0.75% de *F1-micro* para ambos. Para os resultado que utilizam o *Glove* com o SVM, observa-se uma pequena variação nos resultados de 2% de melhoria do melhor resultado para o pior. O menor resultado é com um número de tópicos 50 e 100 (ambos 0.70% de *F1-micro*) e o melhor resultado com 10 tópicos alcançando 0.72% de *F1-micro*. Quando o *Naive Bayes* é utilizado com o CTE, o melhor resultado é quando o algoritmo *Word2Vec* é utilizado: CTE-10-WV-NB com 0.66% de *F1-micro*. Detalhando essa combinação *Word2Vec* e *Naive Bayes*, observa-se que ocorre uma queda nos resultados à medida que o número de tópicos cresce de 10 até 100, e mantém-se constante quando o número de tópicos passa de 100 para 200 (0.62% de *F1-micro* para ambos). Já quando o *Glove* é utilizado com o *Naive Bayes*, também é observado que ocorre uma queda nos resultados à medida que o número de tópicos cresce de 10 até 100 (onde o melhor resultados é com 10 tópicos alcançando 0.66% de *F1-micro*), e mantém-se constante quando o número de tópicos passa de 100 para 200 (0.63% de *F1-micro* para ambos).

Em relação as duas reapresentações que utilizam *Word Embedding* CTE e MEAN. O

Figura 20 – Desempenho CTE - IMDB



(a) *Glove* - SVM

(b) *Word2Vec* - SVM

(c) *Glove - Naive Bayes*

(d) *Word2Vec - Naive Bayes*

Fonte – Elaborado pelo autor

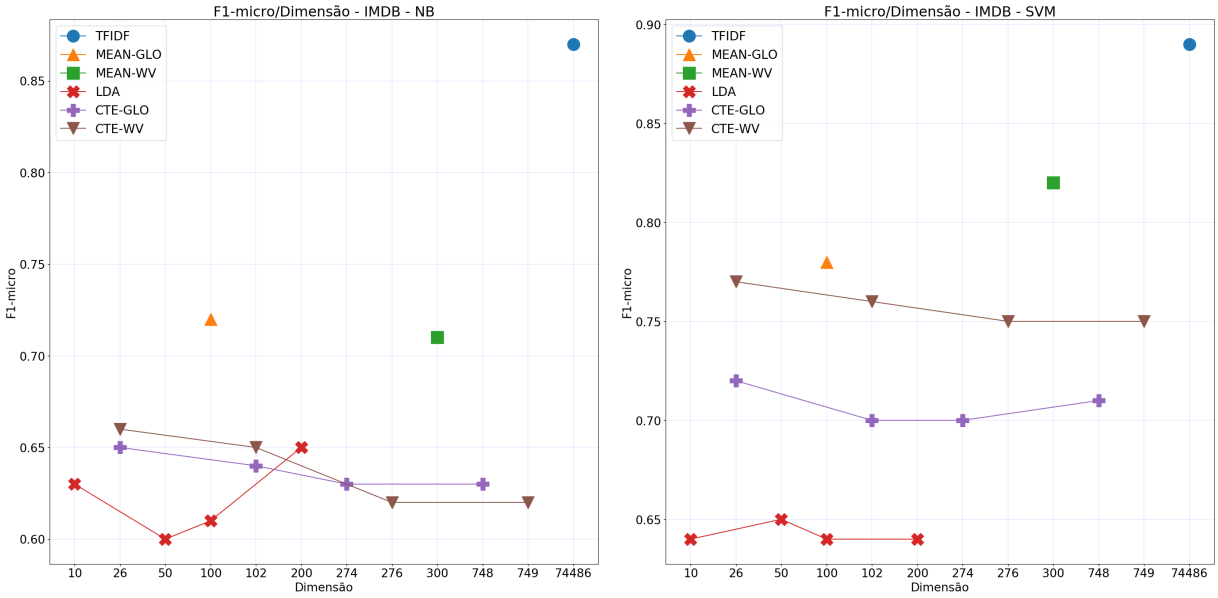
melhor resultado em cada uma é quando o algoritmo *Word2Vec* é usado. O melhor resultado para o *Glove* na representação MEAN é de 0.78%, onde o resultado para o *Word2Vec* corresponde a melhoria de 0.05%. Já o melhor o resultado para o *Glove* na representação CTE é de 0.72%, onde o resultado para o *Word2Vec* corresponde a uma melhoria de 0.06%. Nessa coleção o *Word2Vec* foi superior. Essa vantagem do *Word2Vec* em relação ao *Glove* pode ser interpretada como a representatividade do corpus IMDB para definir e explorar os contextos para representar cada palavra.

No que diz respeito ao desempenho de cada classificador em relação à dimensão de cada representação, a Figura 21 apresenta a variação entre *F1-micro* (eixo Y) e a dimensão de cada abordagem (eixo X) para os classificadores SVM e *Naive Bayes*. Abordagens que apresentem variações são apresentadas como símbolos interligados (LDA, CTE-GLO, CTE-WV). Enquanto abordagens que não apresentem variações são apresentadas apenas como um

símbolo isolado (TF-IDF, MEAN-GLO, MEAN-WV). Uma vez que o CTE apresenta variações com *Glove* e *Word2Vec* para representar as palavras tópicas do método, a dimensão em alguns casos apresentou uma diferença. Por exemplo, com 10 e 50 tópicos o CTE apresentou a mesma dimensão para *Glove* e *Word2Vec* (26 e 162), enquanto com 100 tópicos apresentou dimensão (274-*Glove* e 278-*Word2Vec*) e 200 tópicos (748-*Glove* e 749-*Word2Vec*). Essa diferença deve-se ao fato de que o corpus treinado na *Wikipedia* não contenha todos os termos presentes na coleção de documentos.

Em relação ao SVM, o TF-IDF é o que apresenta melhor resultado (0.89%) porém com a dimensão mais alta 74486. MEAN-WV o segundo melhor *F1-micro* (0.82%) com dimensão baixa de 300, e a abordagem O CTE-10-WV alcançou *F1-micro* (0.77%) com dimensão de tamanho 26. O CTE-10-WV alcançou este resultado com uma dimensão 11 vezes menor que MEAN-WV e aproximadamente 2800 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.64% de *F1-micro*. Em relação ao *Naive Bayes*, o TF-IDF é o que apresenta melhor resultado (0.87%). MEAN-GLO o segundo melhor *F1-micro* (0.72%) com dimensão de 100, e a abordagem CTE-10-WV alcançou *F1-micro* (0.66%) com dimensão de tamanho 26. O CTE-10-WV alcançou este resultado com uma dimensão 3 vezes menor que MEAN-GLO e aproximadamente 2800 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.63% de *F1-micro*.

Figura 21 – Relação F1-score/Dimensão - IMDB



(a) *Naive Bayes* - SVM (b) SVM

Fonte – Elaborado pelo autor

Testes estatísticos foram aplicados quando há dúvidas sobre qual combinação tem melhor performance. Nesta tabela essa situação ocorre quando comparando CTE-50-GLO_SVM com CTE-100-GLO_SVM, MEAN-GLO_NB com MEAN-WV_NB e TFIDF_SVM com TFIDF_NB. Foram aplicados testes *t-student* no primeiro caso a hipótese nula não é rejeitada com *p-value* = 0.36, no segundo caso a hipótese nula não é rejeitada com *p-value* = 0.4 e no terceiro caso a hipótese nula é rejeitada com *p-value* = 0.0000009.

Síntese do experimento: Nota-se destes resultados que, para estes textos curtos, a representação de alta dimensionalidade TF-IDF com classificador não linear SVM supera todas as demais, superando CTE em pelo menos 0.12% de *F1-micro*. Por outro lado, CTE treinado com *Word Embedding* da própria coleção (interna) (WV) não supera as demais representações *Word Embedding* puras, como *Word2Vec* e *Glove*. Embora com *F1-micro* menor, a redução de dimensionalidade de CTE é da ordem de 1000 vezes.

5.2.2 Resultados e Avaliação STS

A Tabela 9 apresenta os resultados para cada REPRESENTAÇÃO_CLASSIFICADOR da coleção de documentos STS.

Mais uma vez o melhor resultado dentre todos os experimentos foi obtido com a representação do TF-IDF (com os dois classificadores) para *F1-micro*. O TFIDF_SVM ficou com *F1-micro* de 0.73% e TFIDF_NB com *F1-micro* de 0.73%.

A Figura 22 apresenta graficamente a variação do LDA. Observa-se que quando o *Naive Bayes* é utilizado como classificador, observa-se uma melhoria crescente no resultado de *F1-micro* quando o número de tópicos cresce (com 10 tópicos *F1-micro* de 0.58%, 50 tópicos *F1-micro* de 0.59%, 100 tópicos *F1-micro* de 0.68% e 200 tópicos *F1-micro* de 0.71%). Com 200 tópicos extraídos, LDA-200_NB alcançou 0.71% de *F1-micro*, o melhor resultado das variações da representação via LDA. Quando o SVM é utilizado pode-se observar o contrário, o resultado de *F1-micro* cai a medida que o número de tópicos cresce (com 10 tópicos *F1-micro* de 0.62%, 50 tópicos *F1-micro* de 0.56%, 100 tópicos *F1-micro* de 0.5% e 200 tópicos *F1-micro* de 0.5%).

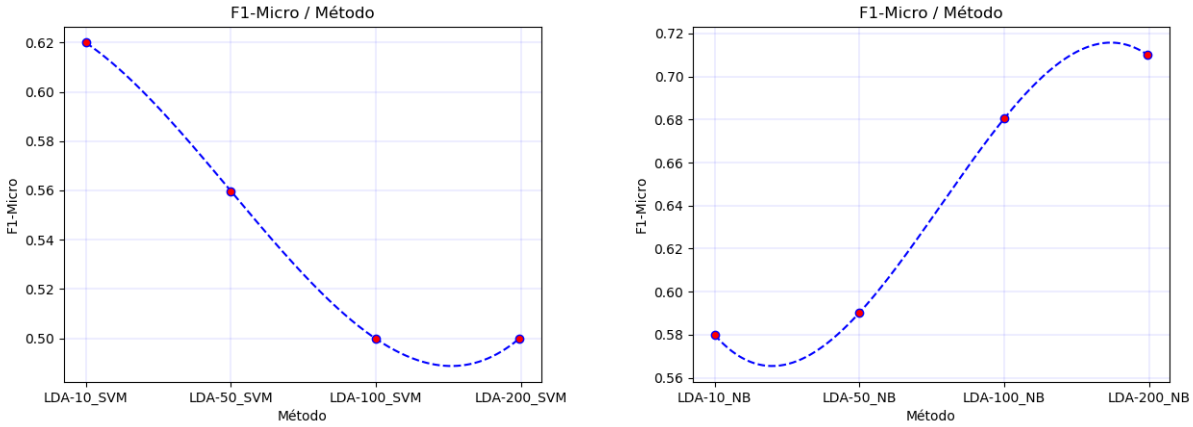
Em relação a média de *Word Embedding*, a abordagem que utiliza o algoritmo *Glove* alcançou um melhor resultado em relação ao algoritmo *Word2Vec*, quando o SVM é utilizado (MEAN-WV_SVM atingiu 0.69% de *F1-micro* e MEAN-WV_SVM atingiu 0.52% de *F1-micro*), da mesma forma quando o *Naive Bayes* é utilizado (MEAN-GLO_NB atingiu 0.61% de *F1-micro* e MEAN-WV_NB atingiu 0.53% de *F1-micro*).

Tabela 9 – Resultados da métrica *F1-Score* - (Média. Variância. Desvio Padrão e Dimensão) para coleção de documentos STS

Representação-Classificador	*F1-micro*	VAR	STD	Dimensão
TFIDF_SVM	**0.73**	0.00006	0.00780	26983
TFIDF_NB	**0.73**	0.00006	0.00793	26983
MEAN-GLO_SVM	**0.69**	0.00018	0.01348	100
MEAN-WV_SVM	0.52	0.00002	0.00447	300
MEAN-GLO_NB	0.61	0.00004	0.00659	100
MEAN-WV_NB	0.53	0.00004	0.00632	300
LDA-10_SVM	**0.62**	0.00071	0.02662	10
LDA-50_SVM	0.56	0.00011	0.01028	50
LDA-100_SVM	0.50	0.00000	0.00103	100
LDA-200_SVM	0.50	0.00000	0.00014	200
LDA-10_NB	0.58	0.00019	0.01373	10
LDA-50_NB	0.59	0.00029	0.01704	50
LDA-100_NB	0.68	0.00062	0.02484	100
LDA-200_NB	**0.71**	0.00101	0.03185	200
CTE-10-GLO_SVM	0.64	0.00012	0.01082	44
CTE-50-GLO_SVM	0.65	0.00014	0.01185	156
CTE-100-GLO_SVM	**0.66**	0.00013	0.01125	346
CTE-200-GLO_SVM	0.66	0.00008	0.00898	810
CTE-10-WV_SVM	**0.52**	0.00002	0.00479	44
CTE-50-WV_SVM	0.51	0.00001	0.00278	161
CTE-100-WV_SVM	0.51	0.00001	0.00309	354
CTE-200-WV_SVM	0.51	0.00001	0.00270	829
CTE-10-GLO_NB	0.52	0.00017	0.01298	44
CTE-50-GLO_NB	0.52	0.00014	0.01178	156
CTE-100-GLO_NB	**0.53**	0.00017	0.01315	346
CTE-200-GLO_NB	0.53	0.00021	0.01447	810
CTE-10-WV_NB	**0.51**	0.00001	0.00322	44
CTE-50-WV_NB	0.51	0.00001	0.00303	161
CTE-100-WV_NB	0.51	0.00001	0.00305	354
CTE-200-WV_NB	0.51	0.00001	0.00307	829

Fonte – Elaborado pelo autor

Figura 22 – Desempenho LDA - STS



(a) *LDA* - SVM

(b) *LDA - Naive Bayes*

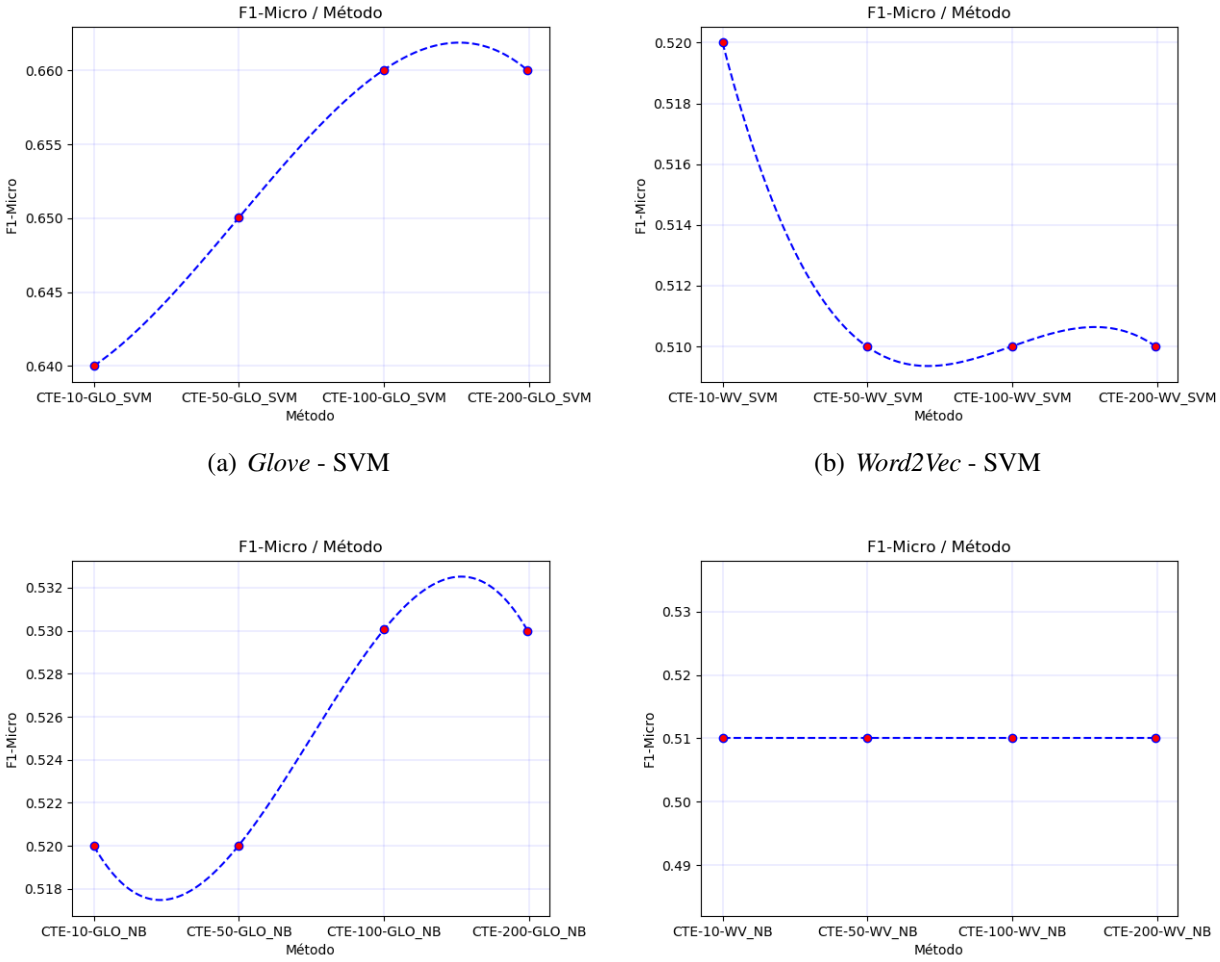
Fonte – Elaborado pelo autor

Com relação aos algoritmos de classificação, observa-se que o SVM superior ao *Naive Bayes* quando o algoritmo *Glove* foi utilizado, houve uma melhoria de 13% do resultado do SVM para o *Naive Bayes*. Já o *Naive Bayes* foi superior ao SVM quando o algoritmo *Word2Vec* foi utilizado, houve uma melhoria de 2% do resultado do *Naive Bayes* para o SVM.

A Figura 23 apresenta graficamente a variação do CTE. Observa-se que a variação que utiliza *Glove* CTE-100-GLO-SVM (da mesma forma que CTE-200-GLO-SVM) obteve o melhor resultado em relação a todos os experimentos do CTE com 0.66% de *F1-micro*. Nessa combinação *Word2Vec* e SVM observa-se um crescimento nos resultados à medida que o número de tópicos cresce de 10 até 100 (0.64% com 10 tópicos e 0.65% com 50 tópicos), e mantém-se constante quando o número de tópicos é incrementado para 100 e 200 com 0.66% de *F1-micro* para ambos. Para os resultados que utilizam o *Word2Vec* com o SVM, observa-se uma queda no resultado quando o número de tópicos passa de 10 para 50, 0.52% para 10 tópicos e 0.51% para 50 tópicos, após isso o resultado mantém-se com 0.51% de *F1-micro*, Quando o *Naive Bayes* é utilizado com o CTE, o melhor resultado é quando o algoritmo *Glove* é utilizado: CTE-100-GLO-NB (da mesma forma que CTE-200-GLO-NB) com 0.53% de *F1-micro*. Detalhando essa combinação Glove e *Naive Bayes*, observa-se que o resultado só cresce quando o número de tópicos passa de 50 para 100 (0.52% para 10-50 tópicos e 0.53% para 100-200 tópicos). Já quando o *Word2Vec* é utilizado com o *Naive Bayes*, o resultado é o mesmo desde 10 tópicos até 200 tópicos (0.51% de *F1-micro* para todos).

Em relação as duas representações que utilizam *Word Embedding* CTE e MEAN, o melhor resultado em cada uma é quando é usado o algoritmo *Glove*. O melhor resultado para o

Figura 23 – Desempenho CTE - STS



(a) *Glove* - SVM

(b) *Word2Vec* - SVM

(c) *Glove - Naive Bayes*

(d) *Word2Vec - Naive Bayes*

Fonte – Elaborado pelo autor

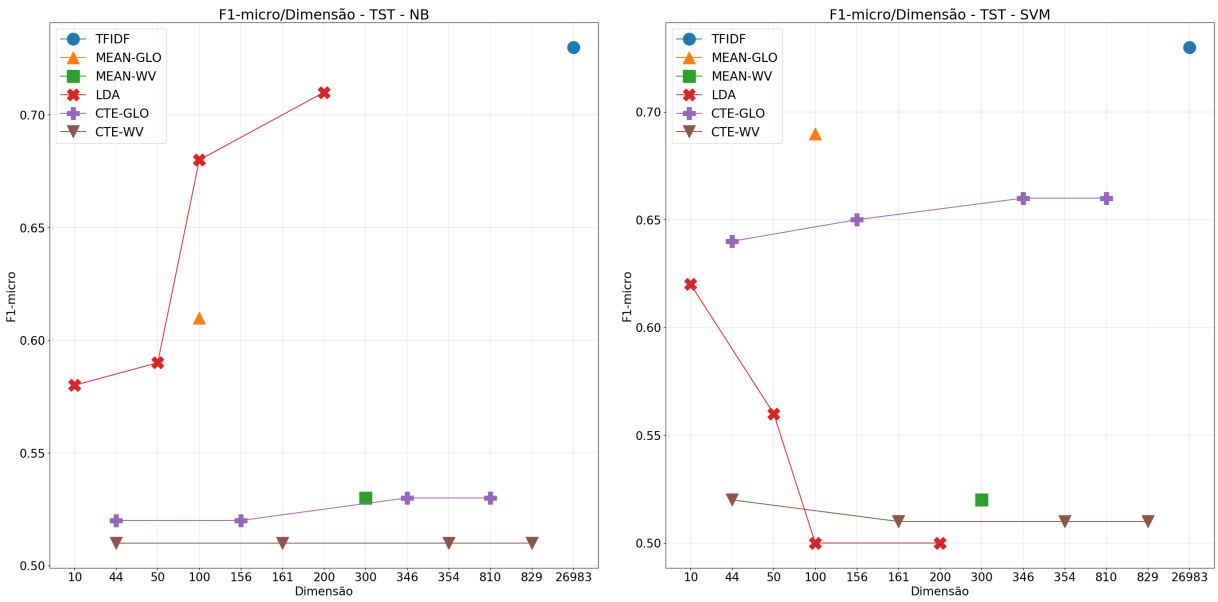
Word2Vec na representação MEAN é de 0.53%, onde para o melhor resultado com *Glove* (0.69%) corresponde à melhoria de 0.3%. Já o melhor o resultado para o *Word2Vec* na representação CTE é de 0.52%, onde o resultado para o melhor resultado com *Glove* (0.66%) corresponde a uma melhoria de 0.26%.

No que diz respeito ao desempenho de cada classificador em relação à dimensão de cada representação, a Figura 24 apresenta a variação entre *F1-micro* (eixo Y) e a dimensão de cada abordagem (eixo X) para os classificadores SVM e *Naive Bayes*. Abordagens que apresentem variações são apresentadas como símbolos interligados (LDA, CTE-GLO, CTE-WV). Enquanto abordagens que não apresentem variações são apresentadas apenas como um símbolo isolado (TF-IDF, MEAN-GLO, MEAN-WV). Uma vez que o CTE apresenta variações com *Glove* e *Word2Vec* para representar as palavras tópicas do método, a dimensão em alguns casos apresentou uma diferença. Por exemplo, com 10 tópicos o CTE apresentou a mesma

dimensão para *Glove* e *Word2Vec* (44), enquanto com 50 tópicos apresentou dimensão (156-*Glove* e 161-*Word2Vec*) com 100 tópicos (346-*Glove* e 354-*Word2Vec*) e 200 tópicos (810-*Glove* e 829-*Word2Vec*). Essa diferença deve-se ao fato de que o corpus treinado na *Wikipedia* não contenha todos os termos presentes na coleção de documentos.

Em relação ao SVM, o TF-IDF é o que apresenta melhor resultado (0.73%) porém com a dimensão mais alta 26983. MEAN-GLO o segundo melhor *F1-micro* (0.69%) com dimensão baixa de 100, e a abordagem O CTE-100-GLO alcançou *F1-micro* (0.66%) com dimensão de tamanho 346. O CTE-100-GLO alcançou este resultado com uma dimensão maior que MEAN-WV e aproximadamente 77 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.62% de *F1-micro*. Em relação ao *Naive Bayes*, o TF-IDF é o que apresenta melhor resultado (0.73%). LDA-200 o segundo melhor *F1-micro* (0.71%) com dimensão de 200, e a abordagem CTE-100-GLO alcançou *F1-micro* (0.53%) com dimensão de tamanho 346. O CTE-100-GLO alcançou este resultado com uma dimensão maior que LDA-200 e aproximadamente 77 vezes menor que o TF-IDF. A menor dimensão de tamanho 10 (LDA) alcançou o resultado com 0.58% de *F1-micro*.

Figura 24 – Relação F1-score/Dimensão - STS



(a) *Naive Bayes* (b) SVM

Fonte – Elaborado pelo autor

Testes estatísticos foram aplicados quando há dúvidas sobre qual combinação tem melhor performance. Nesta tabela essa situação ocorre quando comparando CTE-10-GLO_SVM com CTE-200-GLO_SVM, TFIDF_SVM com TFIDF_NB e MEAN-GLO_SVM e TFIDF_SVM.

Foram aplicados testes *t-student* no primeiro caso a hipótese nula é rejeitada com *p-value* = 0.004, no segundo caso a hipótese nula é rejeitada com *p-value* = 0.04 e no terceiro caso a hipótese nula é rejeitada com *p-value* = 0.0000001.

Síntese do experimento: Nota-se destes resultados que, para estes textos curtos, a representação de alta dimensionalidade TF-IDF com classificador não linear SVM supera todas as demais, superando CTE em pelo menos 0.07% de *F1-micro*. Por outro lado, CTE treinado com *Word Embedding* em corpus externos (GLO) não supera o *Glove*, porém supera *Word2Vec*. Embora com *F1-micro* menor, a redução de dimensionalidade de CTE é da ordem de 1000 vezes.

6 CONCLUSÃO

A questão de pesquisa posta neste trabalho foi verificar até que ponto conceitos construídos a partir de agrupamentos semânticos (tópicos, por exemplo) possibilitam ganhos significativos em tarefas de categorização de textos curtos ou muito curtos, sabendo-se que o contexto tem papel fundamental na construção destes conceitos e que um texto curto carrega um contexto muito limitado. Para respondê-la o objetivo geral foi declarado como sendo "Avaliar o desempenho de combinações de classificador versus representação por conceito em categorização de textos curtos".

O trabalho de Cabrera, Escalante e Gómez (CABRERA; ESCALANTE; GÓMEZ, 2013) mostra com exemplos que a métrica *F1-micro* pode decrescer de 0.64% quando categorizando com o documento todo (notícia curta), para 0.18% quando categorizando apenas com o título (texto muito curto) o que representa uma queda de 72.74%, no pior caso. No melhor caso, *F1-micro* cai de 0.90% para 0.70%, com SVM Linear. A coleção de documentos usada foi o de notícias R8.

Como conclusão geral, a literatura e os resultados deste trabalho mostram que a categorização de textos curtos e muito curtos ainda apresenta resultados baixos e que é um problema com amplo espaço para pesquisa exploratória. Os melhores resultados são obtidos em testes com data sets de domínios específicos mas que não generalizam bem como algoritmos gerais.

6.1 CONTRIBUIÇÕES DO TRABALHO

As principais contribuições deste trabalho podem ser assim resumidas:

- Quando a tarefa é categorização de textos curtos as representações avaliadas que exploram um nível semântico (LDA, Word2vec, *Glove*, CTE) apresentaram desempenho menor do que representações baseadas em contagem de palavras (TF-IDF). Sabendo-se que a dimensionalidade das representações conceituais é de duas a três ordens de vezes menor, já seria um ganho significativo obter desempenho comparável com as representações baseadas em frequência de palavras.
- No experimento de classificação binária de sentimentos a representação proposta CTE alcançou um resultado próximo ao obtido por TF-IDF e MEAN-WV, porém com uma dimensão de representação muito menor. Experimentos exaustivos são necessários para levar a uma conclusão definitiva mas esses resultados apontam que essa proposta pode ser

promissora.

- Embora essa dissertação não tenha explorado o tempo de execução para obter representações de texto e o tempo de execução para treinar cada classificador com a representação de texto definida, o custo em utilizar uma representação com menor dimensão é consideravelmente menor do que representações como o TF-IDF.

6.2 LIMITAÇÕES

Devido ao grande volume de experimentos que se pretendeu realizar, algumas limitações foram estabelecidas já no início deste trabalho. As mais importantes foram:

- Foram calculadas poucas métricas de desempenho não permitindo que outras interpretações possam ser extraídas dos resultados.
- Foi utilizado um número pequeno de bases de dados não permitindo que as conclusões sejam definitivas.

6.3 TRABALHOS FUTUROS

As principais linhas de investigação futura pensadas para a continuidade deste trabalho são:

- Trabalhar com coleções de documentos na língua portuguesa.
- Investigar outras técnicas de Modelos Tópicos como LSA, *Latent Semantic Indexing* (LSI) para construção do CTE, assim como outros algoritmos de *Word Embedding* como o *fastText*, além do uso do *Glove* treinado na própria coleção e *Word2Vec* treinado em um corpus externo como a Wikipédia.
- Investigar se algoritmos de *Deep Learning* podem alterar os resultados obtidos.

REFERÊNCIAS

ADENIYI, D. A.; WEI, Z.; YONGQUAN, Y. Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. **Applied Computing and Informatics**, Elsevier, v. 12, n. 1, p. 90–108, 2016.

ALLAHYARI, M.; POURIYEH, S.; ASSEFI, M.; SAFAEI, S.; TRIPPE, E. D.; GUTIERREZ, J. B.; KOCHUT, K. A brief survey of text mining: Classification, clustering and extraction techniques. **arXiv preprint arXiv:1707.02919**, 2017.

ASSENT, I. Clustering high dimensional data. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 2, n. 4, p. 340–350, 2012.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação - 2ed: Conceitos e Tecnologia das Máquinas de Busca**. Bookman Editora, 2013. ISBN 9788582600498. Disponível em: <https://books.google.com.br/books?id=YWk3AgAAQBAJ>.

BATES, M. Models of natural language understanding. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 92, n. 22, p. 9977–9982, 1995.

BHALCHANDRA, P.; MULEY, A.; JOSHI, M.; KHAMITKAR, S.; DARKUNDE, N.; LOKHANDE, S.; WASNIK, P. Prognostication of student's performance: An hierarchical clustering strategy for educational dataset. **Advances in Intelligent Systems and Computing**, v. 410, p. 149–157, 2016. ISSN 21945357.

BLEI, D. M. Probabilistic topic models. **Commun. ACM**, ACM, New York, NY, USA, v. 55, n. 4, p. 77–84, abr. 2012. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/2133806.2133826>.

BLEI, D. M.; EDU, B. B.; NG, A. Y.; EDU, A. S.; JORDAN, M. I.; EDU, J. B. Latent Dirichlet Allocation. **Journal of Machine Learning Research**, v. 3, p. 993–1022, 2003.

BULLINARIA, J. A.; LEVY, J. P. Extracting semantic representations from word co-occurrence statistics: A computational study. **Behavior Research Methods**, v. 39, n. 3, p. 510–526, Aug 2007. ISSN 1554-3528. Disponível em: <https://doi.org/10.3758/BF03193020>.

CABRERA, J. M.; ESCALANTE, H. J.; GÓMEZ, M. Montes-y. Distributional term representations for short-text categorization. In: SPRINGER. **International Conference on Intelligent Text Processing and Computational Linguistics**. [S.l.], 2013. p. 335–346.

CHEN, J.; VERSPOOR, K.; ZHAI, Z. A Bag-of-concepts Model Improves Relation Extraction in a Narrow Knowledge Domain with Limited Data. p. 43–52, 2019.

CHUNG, J.; WU, C.; TSAI, R. Improve Polarity Detection of Online Reviews with Bag-of-Sentimental-Concepts. **2014.Eswc-Conferences.Org**, p. 1–4, 2014. Disponível em: <http://2014.eswc-conferences.org/sites/default/files/eswc2014-challenges{_}cl{_}submission{}.>

COLLOBERT, R. Word embeddings through hellinger pca. In: CITESEER. **in Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics**. [S.l.], 2014.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, Sep 1995. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00994018>.

DEERWESTER, S.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Cara Hidup Susah.pdf. **Kehidupan**, v. 3, n. 12, p. 34, 2015.

GO, A.; BHAYANI, R.; HUANG, L. Twitter sentiment classification using distant supervision. **CS224N Project Report, Stanford**, v. 1, n. 12, p. 2009, 2009.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

HASAN, K. S.; NG, V. Automatic keyphrase extraction: A survey of the state of the art. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2014. p. 1262–1273.

JAYAWARDANA, V.; LAKMAL, D.; SILVA, N. de; PERERA, A. S.; SUGATHADASA, K.; AYESHA, B.; PERERA, M. Semi-supervised instance population of an ontology using word vector embedding. In: IEEE. **2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)**. [S.l.], 2017. p. 1–7.

JOHNSON, W. B.; LINDENSTRAUSS, J. Extensions of Lipschitz mappings into a Hilbert space. n. January 1984, p. 189–206, 1984.

KANERVA, P.; KRISTOFERSSON, J.; HOLST, A. Random indexing of text samples for latent semantic analysis. **Proceedings of the 22nd Annual Conference of the Cognitive Science Society**, v. 1036, n. 2, p. 16429–16429, 2000. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.6523{&}rep=rep1{&}ty>.

KARLGREN, J.; SAHLGREN, M. 26 From Words to Understanding. n. January 2001, p. 294–311, 1996.

KIM, H.; ROWLAND, P.; PARK, H. Dimension reduction in text classification with support vector machines. **Journal of Machine Learning Research**, v. 6, p. 37–53, 2005. ISSN 15337928.

KIM, H. K.; KIM, H.; CHO, S. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. **Neurocomputing**, Elsevier B.V., v. 266, p. 336–352, 2017. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2017.05.046>.

KIM, K.; CHOI, K.-S. Dimension-reduced estimation of word co-occurrence probability. In: **Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2000.

KORDE, V.; MAHENDER, C. N. Text classification and classifiers: A survey. **International Journal of Artificial Intelligence & Applications**, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 2, p. 85, 2012.

KOWSARI, K.; MEIMANDI, K. J.; HEIDARYSAFA, M.; MENDU, S.; BARNES, L.; BROWN, D. Text classification algorithms: A survey. **Information**, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 150, 2019.

LEWIS, D. D.; YANG, Y.; ROSE, T. G.; LI, F. Rcv1: A new benchmark collection for text categorization research. **Journal of machine learning research**, v. 5, n. Apr, p. 361–397, 2004.

LI, S.; CHUA, T.-S.; ZHU, J.; MIAO, C. Generative Topic Embedding: a Continuous Representation of Documents (Extended Version with Proofs). 2016. Disponível em: <http://arxiv.org/abs/1606.02979>.

LIN, H.; NG, V. Abstractive summarization: A survey of the state of the art. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, p. 9815–9822.

LIU, B. Sentiment analysis and opinion mining. **Synthesis lectures on human language technologies**, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012.

LIU, L.; TANG, L.; DONG, W.; YAO, S.; ZHOU, W. An overview of topic modeling and its current applications in bioinformatics. **SpringerPlus**, Springer, v. 5, n. 1, p. 1608, 2016.

LIU, Y.; LIU, Z.; CHUA, T.-S.; SUN, M. Topical Word Embeddings. **Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)**, v. 2, n. C, p. 2418–2424, 2015.

MAAS, A. L.; DALY, R. E.; PHAM, P. T.; HUANG, D.; NG, A. Y.; POTTS, C. Learning word vectors for sentiment analysis. In: **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**. Portland, Oregon, USA: Association for Computational Linguistics, 2011. p. 142–150. Disponível em: <http://www.aclweb.org/anthology/P11-1015>.

MANIKANDAN, R.; SIVAKUMAR, R. Machine learning algorithms for text-documents classification: A review. **Machine learning**, v. 3, n. 2, 2018.

MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. Introduction to information retrieval. **Natural Language Engineering**, Cambridge university press, v. 16, n. 1, p. 100–103, 2010.

MERROUNI, Z. A.; FRIKH, B.; OUHBI, B. Automatic keyphrase extraction: a survey and trends. **Journal of Intelligent Information Systems**, Springer, p. 1–34, 2019.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed Representations of Words and Phrases and their Compositionality. In: BURGES, C. J. C.; BOTTOU, L.; WELLING, M.; GHAHRAMANI, Z.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 26**. Curran Associates, Inc., 2013. p. 3111–3119. Disponível em: <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>.

MILNE, D.; WITTEN, I. H. An open-source toolkit for mining Wikipedia. **Artificial Intelligence**, Elsevier B.V., v. 194, p. 222–239, 2013. ISSN 00043702. Disponível em: <http://dx.doi.org/10.1016/j.artint.2012.06.007>.

MOURIÑO-GARCÍA, M.; PÉREZ-RODRÍGUEZ, R.; ANIDO-RIFÓN, L. Bag-of-Concepts Document Representation for Textual News Classification (PDF Download Available).pdf. v. 6, n. 1, p. 173–188, 2015.

NASAR, Z.; JAFFRY, S. W.; MALIK, M. K. Information extraction from scientific articles: a survey. **Scientometrics**, Springer, v. 117, n. 3, p. 1931–1990, 2018.

PANG, B.; LEE, L. *et al.* Opinion mining and sentiment analysis. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 2, n. 1–2, p. 1–135, 2008.

PAPADIMITRIOU, C. H.; RAGHAVAN, P.; TAMAKI, H.; VEMPALA, S. Latent semantic indexing: A probabilistic analysis. **Journal of Computer and System Sciences**, Elsevier, v. 61, n. 2, p. 217–235, 2000.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **In EMNLP**. [S.l.: s.n.], 2014.

RAJAGOPAL, D.; CAMBRIA, E.; OLSHER, D.; KWOK, K. A graph-based approach to commonsense concept extraction and semantic similarity detection. **WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web**, p. 565–570, 2013.

RUBIN, T. N.; CHAMBERS, A.; SMYTH, P.; STEYVERS, M. Statistical topic models for multi-label document classification. **Machine learning**, Springer, v. 88, n. 1-2, p. 157–208, 2012.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *et al.* Learning representations by back-propagating errors. **Cognitive modeling**, v. 5, n. 3, p. 1, 1988.

RUSSELL, S.; RUSSELL, S.; NORVIG, P.; DAVIS, E. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2010. (Prentice Hall series in artificial intelligence). ISBN 9780136042594. Disponível em: <https://books.google.com.br/books?id=8jZBksh-bUMC>.

SAHLGREN, M.; CÖSTER, R. Using bag-of-concepts to improve the performance of support vector machines in text categorization. p. 487–es, 2004.

SAHLGREN, M.; KARLGREN, J. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. **Natural Language Engineering**, v. 11, n. 3, p. 327–341, 2005. ISSN 13513249.

SANTOS, C. D.; GATTI, M. Deep convolutional neural networks for sentiment analysis of short texts. In: **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers**. [S.l.: s.n.], 2014. p. 69–78.

SCHOLKOPF, B.; SMOLA, A. J. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. Cambridge, MA, USA: MIT Press, 2001. ISBN 0262194759.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM computing surveys (CSUR)**, ACM, v. 34, n. 1, p. 1–47, 2002.

SILVA, F. T. d. Luppar: Um sistema de recuperação de informação para coleções fechadas de documentos [dissertação de mestrado]. **Fortaleza: MA Ciência da Computação, Universidade Estadual do Ceará - UECE**, 2018.

SONG, G.; YE, Y.; DU, X.; HUANG, X.; BIE, S. Short text classification: A survey. **Journal of multimedia**, Academy Publisher, v. 9, n. 5, p. 635, 2014.

SOUZA, A. A. d. Luppar news-rec: Um recomendador inteligente de notícias [dissertação de mestrado]. **Fortaleza: MA Ciência da Computação, Universidade Estadual do Ceará - UECE**, 2019.

TÄCKSTRÖM, O. An Evaluation of Bag-of-Concepts Representations in Automatic Text Classification. **ReCALL**, p. 1–72, 2005.

TSAI, A. C.-r.; WU, C.-e.; TSAI, R. T.-h.; HSU, J. Y.-j. Building a Concept- Level Sentiment on Commonsense Knowledge. **Ieee Intelligent Systems**, v. 28, n. 2, p. 22–30, 2013.

WANG, F.; WANG, Z.; LI, Z.; WEN, J.-R. Concept-based short text classification and ranking. In: ACM. **Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management**. [S.l.], 2014. p. 1069–1078.

WEBB, A.; COPSEY, K. **Statistical Pattern Recognition**. Wiley, 2011. ISBN 9781119952961. Disponível em: <https://books.google.com.br/books?id=WpV9Xt-h3O0C>.

ZHANG, W.; YOSHIDA, T.; TANG, X. A comparative study of tf* idf, lsi and multi-words for text classification. **Expert Systems with Applications**, Elsevier, v. 38, n. 3, p. 2758–2765, 2011.

ZHANG, X.; WU, B. Short text classification based on feature extension using the n-gram model. In: IEEE. **2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)**. [S.l.], 2015. p. 710–716.