Elaine Cristina Moreira Marques

Redução de características baseada em grupos semânticos aplicados à classificação de textos

Recife - PE/ Brasil

Julho - 2018

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA

Redução de características baseadas em grupos semânticos aplicados à classificação de textos

Dissertação julgada adequada para obtenção do título de Mestre em Biometria e Estatística Aplicada, defendida e aprovada por unanimidade em 17/07/2018 pela comissão examinadora

Área de concentração: Biometria e Estatística Aplicada

Orientador: Dr. Rafael Ferreira Leite de Mello

Coorientador: Dr. Wilson Rosa de Oliveira Junior

Recife

Julho/2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

M357r Marques, Elaine Cristina Moreira.
Redução de características baseada em grupos semânticos aplicados à classificação de textos / Elaine Cristina Moreira Marques . – Recife, 2018.
105 f.: il.

Orientador(a): Rafael Ferreira Leite de Mello.
Coorientador(a): Wilson Rosa de Oliveira Junior.
Dissertação (Mestrado) Universidade Federal Rural de Pernambuco, Pós-Graduação em Biometria e Estatística Aplicada, Recife, 2018.
Inclui referências.

1. Texto - Agrupamento. 2. Texto – Classificação 3. Texto – Redução 4. Word embeddings I. Mello, Rafael Ferreira Leite de, orient. II. Oliveira Junior, Wilson Rosa de, coorient. III. Título.

CDD 310

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA

Redução de características baseadas em grupos semânticos aplicados à classificação de textos

Elaine Cristina Moreira Marques

Dissertação julgada adequada para obtenção do título de Mestre em Biometria e Estatística Aplicada, defendida e aprovada por unanimidade em 17/07/2018 pela comissão examinadora

Orientador:

_____

Dr. Rafael Ferreira Leite de Mello
Orientador

Banca examinadora:

_____

Dr. Wilson Rosa de Oliveira Junior
UFRPE

_____

Dr. Adenilton José da Silva
UFRPE

_____

Dr. Evandro de Barros Costa
UFAL

Dedico este trabalho ao meus pais Edgar e Cássia, os quais tanto amo.

"Eu sinto que sei que sou um tanto bem maior".

Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida, companhia, saúde e benção de acordar todos os dias e poder lutar pelos meus sonhos. "Se Deus fizer ele é Deus, se não fizer ele é Deus".

Aos meus pais Edgar e Cássia, pela paciência inesgotável, força e dedicação e por nunca me deixarem desistir. Pelo amor incondicional, e pelo suporte para aguentar as turbulências da vida e seguir em frente sempre, com muita honestidade, respeito e responsabilidade. Por isso, dedico este trabalho aos que me deram a vida e batalharam, junto comigo.

Ao meu irmão Ederson, por me ajudar em tudo que preciso, ao meus amados sobrinhos Kaleb e Emanuel, por me proporcionar os melhores sorrisos, naqueles momentos em que mais precisei e pensei em desistir, EU AMO vocês, titia.

À minha amada e humilde família, recheada de pessoas simples e coração puro, ensinando-me a respeitar o próximo, a ouvi-los e com estes, aprender, para assim conquistar meus objetivos, ainda que o trabalho seja longo e árduo, obrigada por entenderem minha ausência nesse período e sempre me colocarem em suas orações, por acreditarem no meu potencial, passando energias positivas, apoiando e confiando sempre nas minhas decisões.

A minha prima Gêlda Karla, que sempre será minha fonte de inspiração, de luta, dedicação e muita conquista em sua vida acadêmica e profissional.

Ao meu orientador Rafael Ferreira, por toda paciência, toda disponibilidade em qualquer dia e horário, que mesmo distante esteve presente em todos os passos da construção desta dissertação, obrigada prof. por ser tão solicito não só nesta orientação, mas a todos os membros do nosso grupo de estudos.

Aos meus amigos, Laércio Cerqueira, João Agnaldo, Danielson Lima e Andressa Liberato por sempre estarem ao meu lado. Aos amigos que tanto pertubei e por todo apoio computacional Máverick André e Mayrton Dias. Aos amigos construídos Juan, Susana, Glauce, Zarzar, Rodrigo, Carla Patríca, Álbaro, Wesley e a todos os colegas do DEINFO que estiveram presente nesta caminhada.

Ao secretário Marco Antônio dos Santos pela sua dedicação e competência em seu trabalho, obrigada por toda ajuda. Aos professores do DEINFO Moacyr Cunha, Cristiane, Adenilton e a todos que compõem o departamento. À Capes e a UFRPE, pelo suporte físico e financeiro concedido.

"Pedi, e vos será concedido; buscai, e encontrareis; batei, e a porta será aberta para vós. Pois todo o que pede recebe; o que busca encontra; e a quem bate, se lhe abrirá.
(Bíblia Sagrada, Mateus 7)

Resumo

A classificação de textos é uma técnica que tem como objetivo organizar e categorizar informações, a partir de documentos textuais presentes nas plataformas digitais. Usualmente cada uma das palavras que constituem os documentos são consideradas como uma característica. Esta abordagem para representações textuais simples é chamada *Bag of Words*. Embora estas características sejam importantes para classificar documentos, a maioria delas são irrelevantes e/ou redundantes o que provoca problemas como alta dimensionalidade, tornando a categorização custosa em termos de memória e execução. Para reduzir a grande demanda de recursos computacionais, técnicas de redução de dimensionalidade são aplicadas, como a seleção e a transformação de características. A seleção de características é bastante utilizada na literatura, pelo fato desta possuir um custo computacional mais baixo em relação as outras técnicas. Nesta técnica, características são selecionadas sem apresentar modificações nas características originais, ou seja, ocorre a seleção de um subconjunto que contém apenas as características mais relevantes do conjunto original. Na transformação de características ocorre a formação de um novo conjunto de características, sendo este novo conjunto menor que o conjunto original, contendo novas palavras ocasionadas por meio da combinação ou transformação das palavras originais. É importante frisar que ambos os métodos possuem algum tipo de perda de informação. O objetivo deste trabalho é propor um novo método de redução de dimensionalidade que minimize a perda de informação das características a partir da criação de grupos de palavras semanticamente relacionadas utilizando algoritmos de agrupamento e *Word Embeddings*. Com isso, é possível reduzir a quantidade de características mantendo a semântica de cada palavra. Neste trabalho a redução ocorreu por meio da criação de grupos semânticos. Inicialmente as palavras das bases de dados passaram por uma vetorização, utilizando os métodos Word2Vec e o Glove. Após a vetorização das palavras, foram aplicados os algoritmos de agrupamento, criando grupos menores de características em relação aos grupos originais. O método foi aplicado em bases de dados bastante utilizadas na literatura, alcançando bons resultados, principalmente em dados mais desestruturados, como páginas da Web, notícias, postagens feitas em Twitter, entre outras.

Palavras-chave: *Agrupamento de texto, Classificação de texto, Redução de demensionalidade, Word embeddings.*

Abstract

The classification of texts is a technique that aims to organize and categorize information, from textual documents present on digital platforms. Usually each of the words that constitute the documents are considered as a characteristic. This approach to simple textual representations is called Bag of Words. Although these characteristics are important for classifying documents, most of them are irrelevant and/or redundant, which causes problems such as high dimensionality, making categorization costly in terms of memory and execution. In order to reduce the large demand for computational resources, dimensionality reduction techniques are applied, such as the selection and transformation of characteristics. Characteristic selection is widely used in the literature because it has a lower computational cost compared to other techniques . In this technique, characteristics are selected without presenting modifications in the original characteristics, that is, the selection of a subset that contains only the most relevant characteristics of the original set occurs. In the transformation of characteristics occurs the formation of a new set of characteristics, this new set being smaller than the original set, containing new words caused by the combination or transformation of the original words. It is important to stress that both methods have some kind of loss of information. The objective of this work is to propose a new dimensionality reduction method that minimizes the loss of characteristic information from the creation of semantically related groups of words using clustering algorithms and Word Embeddings. With this, it is possible to reduce the amount of characteristics maintaining the semantics of each word. In this work the reduction occurred through the creation of semantic groups. Initially, the words in the databases were vectorized using Word2Vec and Glove methods. After the words were vectorized, the clustering algorithms were applied, creating smaller groups of characteristics in relation to the original groups. The method was applied in widely used databases in the literature, reaching good results, especially in more unstructured data, such as Web pages, news, Twitter posts, among others.

Keywords: *Text classification, Dimensionality reduction, Text Clustering, Word embeddings.*

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AFSA	*Automatic Features Subsets Analyzer*
ALOFT	*At Least One FeaTure*
Bow	*Bag of Words*
CHI	*Qui-Quadrado*
cMFDR	*Category-dependent Maximum f Features per Document-Reduced*
DF	*Document Frequency*
DR	*Dimensionality Reduction*
DT	*Decision Tree*
FEF	*Feature Evaluation Function*
IC	*Intervalo de Confiança*
IGFSS	*Improved Global Feature Selection Scheme*
KNN	*K-Nearest Neighboors*
ME	*Maximum Entropy*
MFD	*Maximum f Features per Document*
MFDR	*Maximum f Features per Document - Reduced*
ML	*Machine Learning*
MLP	*Multiplayer Perceptron*
MNB	*Multinomial Naïve Bayes*
MRDC	*Multivariate Relative Discrimination Criterion*
NB	*Naïve Bayes*
PCA	*Principal Component Analysis*
RF	*Random Forest*
SVD	*Singular Value Decomposition*

SVM	*Support Vector Machine*
TC	*Text Classification*
TM	*Text Mining*
TV	*Term Variance*

Sumário

1 Introdução

O crescimento pela busca de informação é um fator evidente nos dias atuais, principalmente devido à grande quantidade de dados dispostos na Internet. Fontes de informações como livros, revistas, jornais, blogs, redes sociais, entre outros, produzem grandes quantidades de dados ao longo do tempo (SANTANA, 2008). A Internet tem proporcionado diversas fontes para busca de informação, blogs, perfis de *Twitter*, *Facebook*, as redes sociais em geral são grandes veículos de comunicação entre os indivíduos e são a partir delas que encontra-se, surgindo um cenário de mudanças na forma que se busca, informações.

Atualmente a internet vem proporcionando diversos benefícios, entres eles pode-se citar o setor publicitário por meio da divulgação de produtos, marcas, notícias como política, saúde, educação, fóruns de discussões, postagens em Twitter, índice de satisfação de determinado produto, entre outras informações, vem sendo cada vez mais procurada pelos indivíduos nas redes sociais. Como muitas informações são inseridas a todo momento na internet, torna-se cada vez mais difícil encontrar informação relevante sem ajuda de técnicas computacionais, como recuperação de informação (BÜTTCHER; CLARKE; CORMACK, 2016), classificação de texto (UYSAL, 2016), sumarização de texto (FERREIRA et al., 2013), entre outros.

Essas técnicas estão inseridas na mineração de texto que é uma área que consiste em extrair e analisar dados a partir de textos com diferentes estruturas (SERAPIÃO; SUZUKI; MARQUES, 2010). Além das técnicas já citadas a mineração envolve métodos computacionais como a utilização de algoritmos de agrupamento, que serve para agrupar palavras, como também a área estatística que é vista na mineração como a área que lida com a análise das informações (MACHADO et al., 2010).

Com tantos documentos e informações armazenadas nas plataformas digitais, surgiu a necessidade de facilitar a organização dos conteúdos destes documentos de acordo com o assunto desejado. Uma maneira de se organizar os conteúdos é utilizando a classificação de textos (TC, do inglês *Text classification* ou *Text categorization*) cujo objetivo consiste em indicar documentos de textos a suas respectivas categorias por meio de um conjunto predefinido de categorias (SEBASTIANI, 2002).

A classificação de texto pode ser representada por meio da utilização de diferentes modelos, o mais utilizado na literatura é o modelo de representação vetorial baseada em texto, ou seja, representações textuais simples, conhecido como saco de palavras (BoW, do inglês, *Bag of Words*). A representação BoW mostra bons resultados quando empregada em métodos de análise que procuram uma medida de similaridade entre documentos (como

clustering) (JOACHIMS, 1996). Por outro lado este modelo apresenta um problema de alta dimensionalidade, já que, a dimensionalidade dos vetores é igual ao tamanho do vocabulário de palavras das bases de dados (JOACHIMS, 1998).

Na literatura existem diversas técnicas para solucionar o problema de alta dimensionalidade, algumas técnicas de redução de características são empregadas como a transformação/extração de características e seleção (SEBASTIANI, 2002).

Na técnica de seleção, como o próprio nome informa, algumas características são selecionadas, onde não são apresentadas modificações nas características originais, o que ocorre nesta técnica é a seleção de um subconjunto que contém apenas as características mais relevantes do conjunto original. Na técnica de transformação estas características são transformadas em outras características (SEBASTIANI, 2002).

Nesta dissertação adotamos a técnica de transformação de características, onde, inicialmente são criados os conjuntos de palavras, por exemplo, tem-se dois grupos, A e B que são representados pelas palavras $A_1, \cdots, A_n$ e $B_1, \cdots, B_n$ em um documento que contém 6 palavras $D_1 = A_1, A_2, A_3, A_4, B_2, B_3$ e estas palavras são transformadas em apenas 2 características, A, A, A, A, B, B.

A técnica de seleção de características é uma das técnicas mais utilizadas na literatura quando se trata de classificação de texto, pelo fato desta possuir um custo computacional mais baixo em relação as outras técnicas de redução de dimensionalidade, tornando-a mais rápida e mais utilizada (FRAGOSO, 2016).

Por exemplo, na seleção de características algumas palavras do texto são eliminadas, tendo como consequência a perda de informação em seus resultados. As perdas apresentadas no processo de seleção de características, são consideradas como um problema de busca onde cada estado do espaço de busca representa um subconjunto de características (PINHEIRO; CAVALCANTI; REN, 2015).

O método de transformação de características realizam a transformação por meio da junção de uma ou mais colunas, transformando-as em uma só coluna como pode ser visto no método *Singular Value Decomposition* (SVD), mas conhecida como Decomposição Singular de Valores, baseado em redes neurais artificiais, este método é usado para aprender as relações entre um amplo número de palavras no texto, que não só reduz as dimensões das palavras, mas também procura as relações associativas entre os termos. (LI; PARK, 2007).

O método de transformação também apresenta perda de informações em seus resultados, por exemplo com a mistura das frequências das palavras em diferentes colunas. Esta mistura contribui negativamente nos resultados, fazendo que esta técnica perca o sentido das palavras. Mesmo a transformação apresentando alguns problemas, esta técnica foi adotada por formar um novo conjunto de característica menor que o conjunto

original, onde neste novo conjunto, novos termos são gerados por meio da combinação ou transformação dos termos originais (FRAGOSO, 2016).

Diante o problema de alta dimensionalidade, esta dissertação tem como proposta indicar um novo algoritmo para transformação de características baseado em algoritmos de agrupamento e *Word Embedding*. Mais especificamente, foi proposto a criação de grupos de palavras semanticamente relacionadas utilizados para reduzir o número de características, mantendo a descritividade do conjunto inicial, ou seja, quando os grupos de características são criados, estes não perderão informações por estarem agrupados em grupos menores semanticamente relacionados.

Esta proposta procura resolver o problema da alta dimensionalidade das características, a partir da criação de grupos de palavras, que visam facilitar a busca por determinado conteúdo, classificando todas as informações que pertencem a um determinado assunto a um grupo específico. A partir do método proposto tem-se o intuito de aplicá-lo em blogs, perfis de *Twitter*, onde, por meio de uma palavra específica o conteúdo buscado pelo usuário possa ser identificado e ser apresentado de forma relevante para sua busca.

Para isso avaliou-se diferentes tipos de *Word Embedding* como o Word2Vec (MIKOLOV et al., 2013) e Glove (PENNINGTON; SOCHER; MANNING, 2014) com os algoritmos de agrupamento, K-means, DBSCAN, Birch e Agglomerative Clustering (LI; PARK, 2007).

Logo, a técnica de transformação de características foi aplicada para criar grupos de características e depois tranformá-las em outras. Também foram realizadas comparações de cada um destes algoritmos de agrupamento e as técnicas de *Word Embeddings* em relação aos algoritmos de classificação de texto *Support Vector Machine* e *Random Forest* (FERNÁNDEZ-DELGADO et al., 2014).

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo principal deste trabalho é propor um método de criação de grupos semânticos utilizando algoritmos de agrupamentos e *Word Embeddings* para reduzir a dimensionalidade das características utilizadas para classificação de textos.

1.1.2 Objetivos específicos

A fim de atingir o objetivo geral definiu-se os seguintes objetivos específicos:

- Realizar a revisão da literatura sobre redução de dimensionalidade para classificação de texto;
- Comparar a performance do métodos de *Word Embeddings* para seleção de características aplicadas a classificação de texto;
- Analisar o desempenho dos métodos de agrupamento de texto utilizando *Word Embeddings*;
- Propor métodos de redução de características para classificação de texto;
- Executar experimentos para validar os métodos propostos comparando-os com o do estado da arte.

1.2 Organização da dissertação

Esta dissertação encontra-se dividida em seis capítulos. O Capítulo 1, introduz a problemática em relação a redução de dimensionalidade de características encontrada na classificação de texto e os objetivos do trabalho proposto. No Capítulo 2, são apresentados os conceitos em relação ao tema deste trabalho, como teorias de mineração de texto, classificação de texto, agrupamento e o conceito de *Word Embeddings* formas as quais ocorreu a vetorização das palavras. O Capítulo 3, são apresentados os trabalhos relacionados em relação ao método proposto juntamente com uma tabela que mostra as principais diferenças dos trabalhos com os do estado da arte. O Capítulo 4, apresenta o método utilizado para realizar os experimentos, aqui são explicados as etapas de treinamento e aplicação dos métodos abordados. No Capítulo 5, são mostrados os experimentos realizados. No Capítulo 6 é apresentada a conclusão, algumas contribuições e limitações encontradas no trabalho, como também os trabalhos futuros.

2 Embasamento Teórico

Este capítulo apresenta os principais conceitos de mineração de textos, suas técnicas como a classificação de textos a partir da utilização dos algoritmos de classificação *Support Vector Machine* (SVM) e *Random Forest* (RF) e a técnica de agrupamento de dados, onde se fez uso dos algoritmos de agrupamento K-Means, Agglomerative Clustering, DBSCAN e Birch, as palavras (termos) foram representadas vetorialmente através de *Word Embeddings*, o Word2Vec e Glove, os conceitos de todos os assuntos abordados estão descritos aqui.

2.1 Mineração de texto

Mineração de texto (*Text Mining*) é uma área que extrai e analisa dados a partir de textos dos mais variados tipos, fornecendo informações de interesse dos usuários a partir de documentos de textos não-estruturados (SERAPIÃO; SUZUKI; MARQUES, 2010), a mineração de texto envolve várias técnicas computacionais, como a classificação de texto, sumarização, extração de informações, agrupamento entre outros e além da computação a área estatística torna-se presente, pelo fato desta lidar com a análise das informações (MACHADO et al., 2010).

As técnicas computacionais utilizadas na mineração de texto servem para processar/recuperar documentos de textos e identificar a partir dos documentos quais destes são relevantes para a busca que foi realizada pelo indivíduo (BAEZA-YATES; RIBEIRO-NETO, 2013).

Algumas vantagens podem ser observadas na mineração de texto, como encontrar informações específicas em um documento, ou seja, encontrar detalhes de informações relevantes sem que o usuário leia todo o texto, além desta vantagem a mineração de texto pode ser utilizada em quaisquer meios que façam uso de textos (LANDAUER; FOLTZ; LAHAM, 1998). A mineração é a principal ferramenta para buscas específicas em documentos textuais, ou seja, recupera informações, encontra padrões, realiza análises tanto qualitativas quanto quantitativas em grandes volumes de textos, além de realizar o melhor entendimento dos conteúdos dispostos nos documentos textuais utilizados, a mineração de texto identifica documentos que sejam similares entre si, como também busca por informações relevantes dentro do documento.

Conclui-se então, que a mineração de texto é definida como processo que fornece uma colaboração na descoberta de conhecimento, por meio dos documentos de textos, que podem ser utilizadas nas inúmeras áreas de conhecimento (BARION; LAGO, 2015).

2.2 Classificação de texto

Diante do grande volume de dados, diversos documentos em formato de texto são armazenados para recuperação e posterior utilização, por exemplo, uma leitura de determinado documento. No entanto, existem diversas formas desses documentos serem armazenados, mas para facilitar este armazenamento algumas medidas foram tomadas afim de melhorar o processo de busca de um determinado documento, como a sua organização.

A classificação ou categorização de textos (TC, do inglês *Text classification* ou *Text categorization*) é uma técnica cujo objetivo é organizar documentos, indicando os textos destes documentos a suas respectivas categorias por meio de um conjunto predefinido de palavras que são destinadas as classes ou categorias que mais possuem similaridade (SEBASTIANI, 2002).

De maneira mais formal, o processo de classificação de textos recebe como entrada um documento $\boldsymbol{D}$ e um conjunto fixo de classes $C = \{c_1, c_2, \cdots, c_q\}$. A saída apresentada por este processo, consiste em determinar a classe em que o documento $\boldsymbol{D}$ esteja relacionado semanticamente.

Buscas rápidas de determinados conteúdos que encontram-se organizados são realizadas pela classificação, que facilita a recuperação de informações (IR, do inglês *information retrieval*), que tem como objetivo recuperar documentos que sejam relevantes para a busca do usuário. A categorização das palavras é realizada através de uma rotulação, onde cada documento possui um identificador único (BAEZA-YATES; RIBEIRO-NETO, 2013).

Logo, a partir dos mais variados documentos existentes, grupos são criados e posteriormente rotulados, ou seja, são classificados através de conjuntos e são inseridos conteúdos de acordo com o rótulo definido. Neste caso o rótulo serve para referenciar os tópicos dos documentos, nos tópicos podem ser observadas as características dos documentos, como sua qualidade, gênero, entre outras. Este processo de rotulação dos documentos é conhecido como classificação ou categorização de texto.

Na classificação de texto inicialmente é realizado a representação vetorial, ou seja, transforma-se os documentos de textos em uma representação vetorial apropriada para o algoritmo de aprendizagem. Nesta representação cada elemento do vetor indica uma determinada característica do texto, ou seja, os termos presentes nos documentos são descritos como características e os documentos textuais representam as instâncias.

A classificação de texto pode ser representada por meio da utilização de diferentes modelos, o mais utilizado na literatura é o modelo de representação vetorial baseada em texto, conhecido como saco de palavras (BoW, do inglês, *Bag of Words*). No modelo BoW, cada um dos documentos de textos são representados por um vetor de características, aqui as palavras do vocabulário são associadas a frequência das palavras presentes nos

documentos.

Ao utilizar a representação BoW os métodos apresentam bons resultados quando empregada em métodos de análise, que procuram uma medida de similaridade entre documentos (como *clustering*) (JOACHIMS, 1996). Por outro lado, este modelo apresenta um problema de alta dimensionalidade, uma vez que, a dimensionalidade dos vetores é igual ao tamanho do vocabulário de palavras das bases de dados. Muitas das características presentes nos documentos são irrelevantes ou redundantes para o experimento, o que torna a classificação de texto custosa, em relação ao tempo de execução dos experimentos e memória consumida. E estas características irrelevantes são designadas como um problema para a classificação de textos.

Problemas de alta dimensionalidade podem ser solucionados por algumas técnicas de redução de características (DR, do inglês *Dimensionality Reduction*) que tem como objetivo realizar o aperfeiçoamento do desempenho da classificação contribuindo assim para a redução do esforço computacional (FRAGOSO, 2016). Estas técnicas de redução são divididas em seleção e transformação, onde na seleção, como o próprio nome informa, algumas características são selecionadas e na transformação estas características são transformadas em outras características.

A técnica adotada no presente trabalho é a transformação de características, onde inicialmente são criados os conjuntos de palavras e esse conjunto é transformado em um único grupo, ou seja, após o agrupamento estas características são transformadas em outras.

A técnica de seleção de características é uma das técnicas mais utilizadas na classificação de texto, por possuir um custo computacional mais baixo em relação as outras técnicas de redução de dimensionalidade, tornando-a mais rápida e mais utilizada (YU; LIU, 2003). Por exemplo na seleção de características algumas palavras do texto são eliminadas, tendo como consequência a perda de informação em seus resultados.

O método de transformação também é uma técnica que possui alto custo computacional, contribuindo assim para um alto consumo de memória. Esta por sua vez também apresenta perda de informações em seus resultados. Por exemplo, na transformação os métodos tradicionais realizam uma transformação por meio da junção de uma ou mais colunas, transformando-as em uma só coluna, isto causa a mistura das frequências das palavras, como pode ser visto no método SVD, então esta mistura contribui negativamente nos resultados, fazendo que esta técnica perda o sentido das palavras.

2.2.1 Algoritmos de Classificação de texto

2.2.1.1 Máquinas de Vetores de Suporte (SVM)

Máquinas de Vetores de Suporte (SVM, do inglês, *Support Vector Machine*) (RAKOTOMAMONJY, 2003) (ZHANG; YOSHIDA; TANG, 2008), criado por Vapnik em 1995 (CORTES; VAPNIK, 1995) é um dos algoritmos de indução de classificadores de texto utilizados primeiramente por Joachims (1998) como também é empregado nesta dissertação pelo fato de possuir habilidades para resolver problemas de alta dimensionalidade (JOACHIMS, 1998) (LEOPOLD; KINDERMANN, 2002).

SVM, é uma técnica que vem sendo bastante utilizada na classificação de textos, como também pode ser observada no aprendizado de problemas frequentemente vistos em reconhecimento de padrão. O SVM é uma família de classificadores que procura de forma não aleatória o melhor ponto de divisão entre as classes, por meio da utilização de hiperplanos (BAEZA-YATES; RIBEIRO-NETO, 2013).

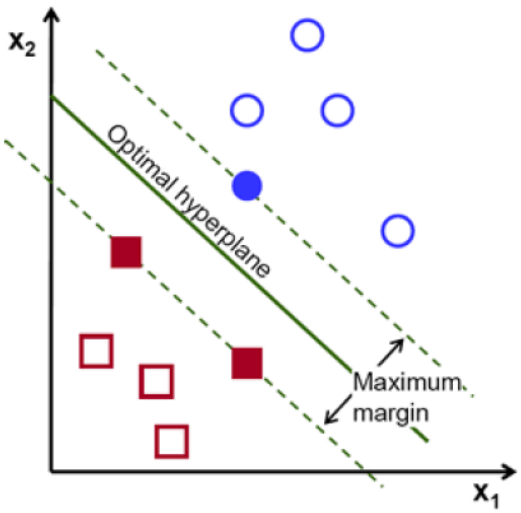


Figura 1 – Hiperplano ideal com distância máxima entre limites de classe em um SVM.
Fonte: Documentação online do OpenCV

Como as máquinas de vetores de suporte são formadas por um conjunto de algoritmos de aprendizado automático supervisionado, o seu treinamento é realizado em função da localização das regiões de fronteiras entre as classes, encontrando assim a melhor separação do hiperplano. As amostras de formação presentes nas fronteiras são denominadas vetores de suporte e são utilizadas para encontrar o hiperplano ideal que satisfaz as propriedades da margem máxima, como apresentado na Figura 1 (WU et al., 2008).

Em um problema de classificação binária o SVM forma um método espaço vetorial, por exemplo, aqui os documentos são representados como vetores ou pontos em um espaço t-dimensional, dada as representações a ideia é encontrar uma superfície de decisão por meio do hiperplano. Logo, no processo de classificação o hiperplano irá separar as duas

classes a partir dos dados de treinamento. Esta separação resulta na maximização das distância entre as classes, ou seja, após a divisão das duas regiões, todos os documentos que compõem a classe C_a encontram-se em uma região e todos os documentos da classe C_b pertencerão a outra região. Quando o espaço é representado por duas dimensões, esse hiperplano apresentará uma linha. Em um espaço tridimensional o hiperplano será um plano (BAEZA-YATES; RIBEIRO-NETO, 2013). Um vez que o hiperplano é aprendido e realiza a separação das classes, a classificação de um novo ponto é trivial (WU et al., 2008).

Por definição, cada um dos textos do conjunto de treinamento é visto como um ponto x_i no espaço $\Re^M$ e o aprendizado do classificador consiste em encontrar a melhor decisão e "separa" os elementos positivos dos negativos no espaço vetorial (ZHANG; YOSHIDA; TANG, 2008).

Para espaço linearmente separável o hiperplano pode ser escrito como:

$$wx + b = 0, \tag{2.1}$$

onde x é um valor arbitrário dos objetos classificados; o vetor w e o vetor b são constantes aprendidas a partir do conjunto de treinamento dos objetos linearmente separáveis.

Afim de resolver problemas linearmente restrito de programação quadrática o SVM deve ser sempre globalmente ideal e para isso é necessário que satisfaça a equação 2.2:

$$\min_{\omega} \frac{1}{2}||\omega||^2 + C\sum_i \xi_i \tag{2.2}$$

quando há restrição utiliza-se:

$$y_i(x_i w + b) \geq 1 - \xi_i \qquad \xi_i \geq 0, \forall_i \tag{2.3}$$

Quando os objetos não são separáveis, têm-se como entrada os dados originais e estes são transformados em um espaço de dimensão superior utilizando o hiperplano linear e o mapeamento não linear. A separação também pode ser realizada em um novo espaço sem que a complexidade computacional do problema de programação da função quadrática aumente, esse tipo de problema pode ser visto na função Kernel que é usada para quando são originadas as semelhanças do espaço dimensional original inferior (ZHANG; YOSHIDA; TANG, 2008).

A função Kernel é determinante em aplicações de máquinas de vetores de suporte, o Kernel utilizado neste trabalho é denominado Kernel polinomial que tem como característica analisar as combinações de amostras utilizadas como entrada para determinar a similaridade entre as palavras.

A função Kernel é dada por:

$$K(x, y) = (x \cdot y + 1)^p \tag{2.4}$$

onde, $p = \text{n}$, ou seja, é um espaço projetado com n-dimensões. Um único parâmetro é dito p, caso contrário, se existe dois parâmetros deve-se utilizar $(x \cdot y + b)^p$.

Temos que $p = 1$ é uma progressão linear, ou seja, sem progressões. Quando $p = 2$ têm-se um progressão quadrática.

Logo, o classificador SVM mostra que a partir do hiperplano procura-se por uma superfície de decisão que apresente maior distância entre qualquer ponto de dados. A distância apresentada pela superfície de decisão ao ponto de dados mais próximo deve determinar a margem do classificador. E o ponto essencial do classificador é o conceito de maximização de margem (JOACHIMS, 1998).

2.2.1.2 *Random Forest* (RF)

Random Forest é um algoritmo de classificação que utiliza métodos de árvores de decisão, foi proposto por Breiman em 2001 (BREIMAN, 2001), também é um algoritmo clássico, ou seja, classificador bastante utilizado para resolver problemas de TC. No entanto, este se difere dos outros algoritmos de árvores de decisão, o objetivo deste classificador é realizar a construção de várias árvores de decisão, que utiliza subconjuntos aleatórios de atributos extraídos do conjunto original de todos os atributos. Este conjunto original dispõe de um tipo de amostragem com reposição conhecida também como *bootstrap*, que facilita a análise dos dados.

O classificador RF, após a construção dos subconjuntos gera diversas árvores de decisão, as quais são construídas de forma simultânea, onde são consideradas todas as variáveis selecionadas para a análise. As árvores são utilizadas de forma conjunta na classificação de novos objetos, tendo como consequência um conjunto de resultados. Logo cada uma das árvores fornecerá um voto em relação a escolha da classe em que o objeto pertence. Daí o algoritmo de classificação define as classes dos objetos de acordo com o maior número de votos para cada uma das classes, "sendo que quanto menor a similaridade entre duas árvores melhor, e pela força que cada árvore tem individualmente, ou seja, quanto mais precisa uma árvore for, melhor será sua nota."

Algumas características tornam a utilização do classificador *Random Forest* mais vantajosa em relação as outras técnicas, onde este é:

- Algoritmo de classificação mais poderoso quando comparado a uma árvore de decisão;
- Apresenta boa taxa de acertos obtida ao testar o classificador em diferentes conjuntos de dados;
- Evita sobre ajustes (overfitting);
- O RF é menos sensível a ruídos;
- Classifica aleatoriamente as árvores sem que ocorra intervenção humana.

O funcionamento do classificador *Random Forest*, é dado da seguinte maneira: a partir de um elemento X, ou seja uma base de dados gera-se várias árvores, onde cada uma delas gera regras e a partir destas regras, novos padrões são descobertos. Estas descobertas contribui para que a decisão mais adequada aproxime-se o máximo dos padrões encontrados.

Após a criação das árvores, ou como são conhecidas, florestas aleatórias, o próximo passo é calcular qual das árvores apresentam regras mais exatas. Após a escolha da árvore com maior exatidão, esta é aplicada na base de dados e apresenta um resultado denominado Y.

2.3 Agrupamento

Para se extrair conhecimento de algumas base de dados é fundamental separar estes dados em forma de grupos (*clusters*) que possuam algum significado para análise. A criação destes grupos é necessária para que se realize uma melhor investigação da grande massa de dados gerada pela web (TAN et al., 2006).

O agrupamento consiste em aproveitar as informações obtidas a partir dos dados e assim dividi-las em grupos, onde, os elementos semelhantes ou relacionados entre si pertençam aos mesmos, esse agrupamento ocorre também para grupos que possuem pouca relação. Obtém-se o melhor agrupamento pela grande semelhança entre os elementos do grupo e a maior diferença entre eles.

A técnica de agrupamento é comum quando deseja-se realizar uma análise estatística ou a generalização dos dados de forma exploratória. Logo, esta é uma técnica necessária, para que informações de mesma características sejam analisadas sem a presença de informações irrelevantes. Tanto a similaridade entre os termos dos grupos quanto a dissimilaridade devem ser investigadas.

Na Figura 2, são apresentados exemplos de agrupamento de dados. Nestes exemplos é visto a semelhança de pontos do mesmo grupo por meio da distância euclidiana entre os mesmos. Pontos próximos concentram-se em um mesmo grupo, pontos mais distantes pertencem a grupos separados.

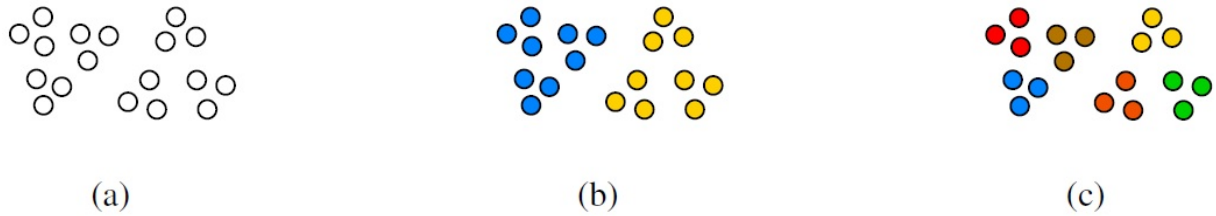


Figura 2 – Exemplos (a) conjunto de dados originais, (b) dois grupos, (c) seis grupos. Fonte: (COLYER, 2016)

Nos exemplos apresentados anteriormente, é possível observar o exemplo **(a)**, onde o grupo apresenta o conjunto de dados originais como entrada sua representação é dada por pontos cartesianos, o exemplo **(b)** formado por dois grupos e o exemplo **(c)** formado por seis grupos de acordo com os critérios estabelecidos e as distâncias entre os termos.

2.3.1 Etapas do processo de Agrupamento

Para realizar o agrupamento de dados são necessárias quatro etapas fundamentais: (XU; WUNSCH, 2008).

1. **Seleção e extração de atributos:** Após a obtenção da base de dados, é fundamental encontrar um subconjunto de atributos, adequado para se realizar o agrupamento. Os atributos em relação a base são coletados e depois estes são transformados tendo como resultado novos atributos, provenientes dos dados originais. Esta etapa de seleção deve ser bem realizada, onde uma boa seleção dos atributos contribui para um menor custo computacional dos cálculos das medidas, armazenamento e agrupamento dos dados.

2. **Seleção de algoritmos de agrupamento:** Aqui os algoritmos de agrupamento são selecionados de acordo com as características dos dados que se deseja trabalhar. Os algoritmos são escolhidos baseado na sua função de similaridade e na função objetivo. Para um bom agrupamento dos dados é necessário escolher um bom algoritmo e adequar da melhor possível seu parâmetro para se obter resultados satisfatórios.

3. **Validação dos grupos:** Os resultados dos agrupamentos devem ser passados por uma avaliação, logo, medidas que visam avaliar estes parâmetros são aplicadas para se obter um resultado de qualidade e confiável para a análise dos dados. Estas medidas de avaliação são aplicadas da mesma forma em todos os algoritmos de agrupamento, onde seu objetivo é avaliar apenas os resultados obtidos por cada um destes, as medidas devem fornecer informações sobre a quantidade de grupos existentes nos dados e se os resultados apresentados pelos algoritmos são significativos.

4. **Interpretação dos resultados:** Nesta última etapa é possível realizar uma análise dos resultados por meio dos agrupamentos obtidos, podendo assim obter resultados significativos que expliquem bem as informações extraídas dos dados.

Na Figura 3, mostra-se o processo de agrupamento dos dados, algumas etapas são realizadas apenas uma única vez outras são repetidas até encontrar o melhor agrupamento, para se obter o melhor agrupamento dos dados as tarefas de mudanças de parâmetros ou técnicas são repetidas n vezes, desta forma pode-se garantir o melhor agrupamento dos dados.

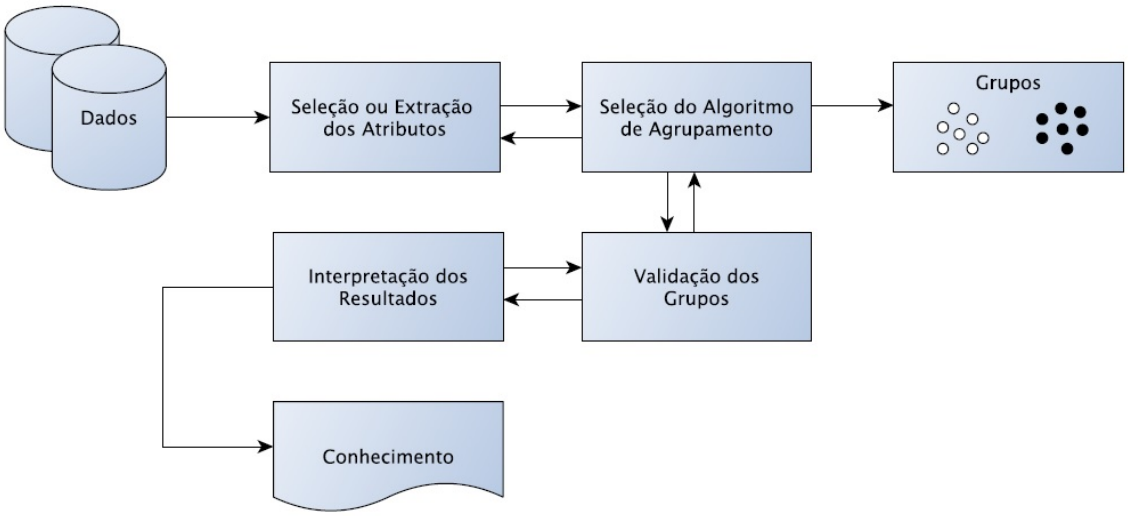


Figura 3 – Exemplo da realização das etapas do processo de agrupamento de dados.
Fonte: (XU; WUNSCH, 2008)

2.3.2 Algoritmos de Agrupamento

Existem diversos tipos de algoritmos de agrupamento presentes na literatura, cada um deles pertecem a uma determinada categoria, com suas próprias características (FAHAD et al., 2014). Os algoritmos de agrupamentos podem ser classificados em:

- **Algoritmos baseado em particionamento:** O número de grupos a ser criados é determinado inicialmente. Os grupos formados devem conter pelo menos um objeto e cada um destes, deve pertecer a exatamente um grupo.

  Por exemplo o algoritmo K-means possui um centro, onde neste centro estão as médias aritméticas de todos os pontos e coordenadas.

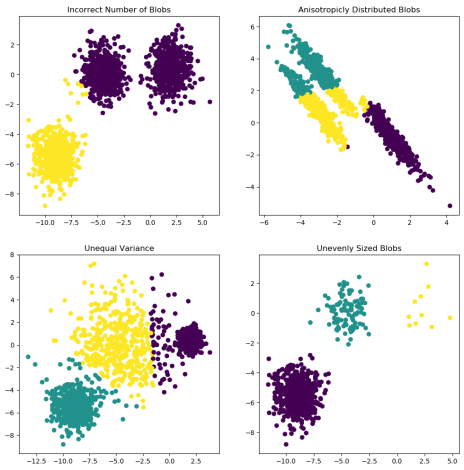


Figura 4 – Exemplo suposições de agrupamento formado pelo K-means.
Fonte: (SCIKIT-LEARN, 2018)

A Figura 4, apresenta um exemplo ilustrativo da criação de grupos não intuitivos. Estes foram gerados pelo algoritmo agrupamento K-means, os primeiros 3 (três) gráficos são apresentados com falhas nas suposições do algoritmo, apenas o último retorna grupos intuitivos.

- **Algoritmos baseado em Hierarquia:** As hierarquias são dadas por meio das proximidades obtidas pelos nós intermediários, de acordo coma organização dos dados. Estes são representados por um dendograma, os dados peculiares são representados pelos nós das folhas e o grupo inicial é dividido em vários outros grupos. Os grupos baseados em hierarquias podem ser aglomerativos (*bottom-up*), onde se inicia com um obejto para cada grupo, posteriormente mescla recursivamente dois ou mais grupos, ou divisível (*top-down*), onde é iniciado com um grupo do conjunto de dados e depois é dividido até obter-se o grupo mais apropriado, de acordo com o número de grupos pré definido. A Figura 5 é um exemplo da utilização do algoritmo BIRCH e MiniBatchKmean, uma comparação entre eles é realizada, pois este quando não se é escolhido a quantidade de grupos seus dados são apresentados de forma reduzida, ocorrendo assim um pré-processamento antes da etapa final, reduzindo cada vez mais os tamanhos dos grupos. Já a Figura 6 presenta o tipo de agrupamento formado pelo algoritmo Agglomerative Clustering utilizando a distância euclidiana.

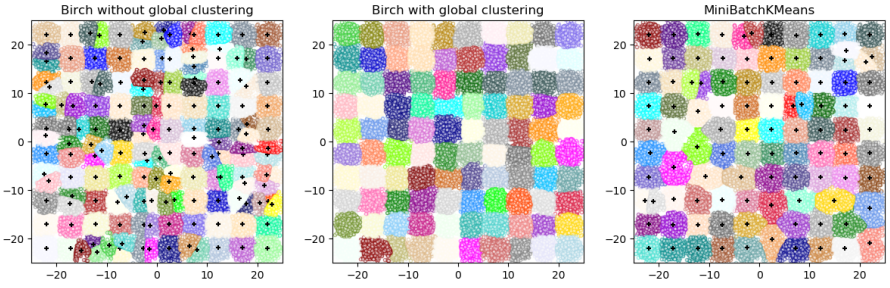


Figura 5 – Exemplo comparação dos grupos criados pelo algoritmo BIRCH.
Fonte: (SCIKIT-LEARN, 2018)

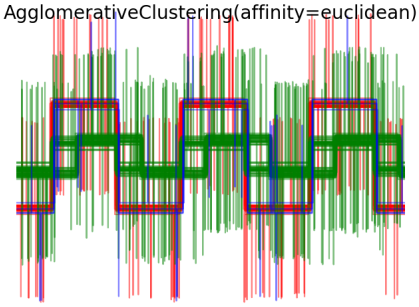


Figura 6 – Exemplo suposição de agrupamento formado pelo *Agglomerative Clustering*.
Fonte: (SCIKIT-LEARN, 2018)

- **Algoritmos baseado em Densidade:** Os objetos são separados de acordo com as suas regiões de densidade, conectividade e limite. Estes objetos estão relacionados aos vizinhos mais próximos do ponto. Um grupo cresce em qualquer direção por ser um componente denso. Os algoritmos baseados em densidade possuem a característica de descobrir aglomerados de forma arbitrária e proteger-se de *outliers*.

  Por exemplo o algoritmo DBSCAN é um dos algoritmos que utilizam o método para filtrar o ruído (*outliers*) e descobrir aglomerados de forma arbitrária. Na Figura 7, é apresentada as principais densidades de uma amostra de grupos realizada pelo algoritmo DBSCAN.

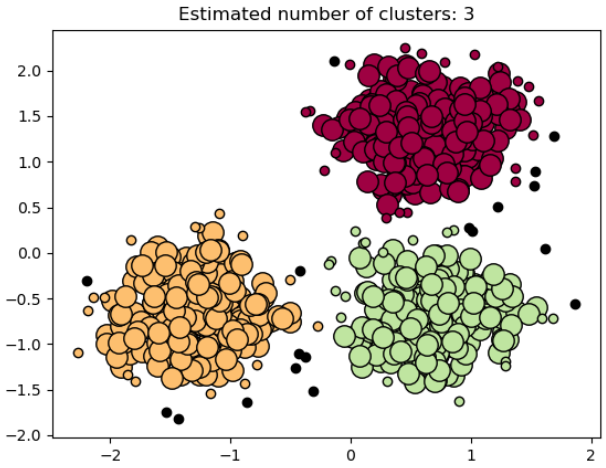


Figura 7 – Visão geral das categorias de agrupamentos.
Fonte: (SCIKIT-LEARN, 2018)

2.3.2.1 K - Means

Os grupos criados pelo algoritmo K-means (também conhecido por K-médias) agrupa dados buscando separar amostras em $\boldsymbol{n}$ grupos de variâncias iguais, tal separação minimiza o critério da **inércia** ou soma de quadrados dentro dos *clusters*. Este algoritmo apresenta-se de forma eficiente para um número grande de amostras e exige que o número de grupos seja especificado inicialmente.

O algoritmo divide o conjunto de **N** amostras **X** em **K** *clusters* disjuntos **C**, onde cada um destes são descritos pela média μ_j das amostras nos *clusters*. A parte central dos *clusters* são denominadas centróides, em geral, estes não são pontos de **X**, emboram pertençam ao mesmo espaço. Os centróides escolhidos pelo algoritmo visam dominuir a inércia ou a soma de *clusters* do critério ao quadrado (MACQUEEN et al., 1967).

$$\sum_{i=0}^{n} min_{\mu_j \epsilon C}(||x_j - \mu_j||^2) \tag{2.5}$$

A inércia ou também como conhecida, soma dos quadrados dentro do *cluster*, pode ser vista como uma medida que responde quão coerente são os *clusters*. A inércia possui alguns problemas como:

- Pressupõe que os *clusters* são convexos e isotrópicos, onde nem sempre estas características são observadas. Não aplica-se de forma satisfatória a aglomerados alongados, ou de formas irregulares.

- Não possui métrica normalizada, o único pressuposto que se sabe é que os melhores valores são baixos, ou seja, próximos a zero e o valor ideal é zero. Quando se possui espaços com altas dimensões, as distâncias euclidianas não são satisfatórias, ou seja, ocorre o processo de "maldição da dimensionalidade". Se aplicar o PCA, algoritmo de redução de dimensionalidade antes do agrupamento k-means, os problemas podem ser resolvidos e assim acelerar os cálculos.

O **K**, de K-means, representa a quantidade de centroides, ou seja, os pontos centrais dos grupos que serão criados afim de encontrar a similaridade dos dados. O número das classes deve ser passada para o algoritmo.

Passos para se realizar o agrupamento pelo algoritmo K-Means

Uma das maneiras de se iniciar o processo de agrupamento, é o algoritmo inserir o **K** pontos (centroides) aleatórios iniciais. Desta forma pode-se começar as iterações e encontrar os possíveis resultados.

1. **Fornecer valores para os centroides:** No primeiro passo os **K** centroides recebem os valores iniciais. No início do algoritmo geralmente escolhe-se os primeiros pontos da tabela. É importante inserir todos os pontos em um centroide para que o processamento do algoritmo seja iniciado.

2. **Gerar uma matriz de distância entre cada ponto e os centroides:** Nesta etapa, são calculadas a distância entre cada ponto e os centroides. Esta é considerada a parte mais custosa dos cálculos, por exemplo, se existirem **N** pontos e **K** centroides, deve-se calcular N x K distância nesta etapa.

3. **Colocar cada ponto nas classes de acordo com a sua distância do centroide e da classe:** Os pontos aqui são classificados de acordo com sua distância em relação aos centroides de cada classe.

   A classificação acontece da seguinte forma: o ponto irá pertencer à classe representada pelo centroide que está mais próximo do ponto. O algoritmo finaliza sua execução caso não exista a mudança de classe de nenhum ponto.

4. **Calcular os novos centroides para cada classe:** Aqui acontece o refinamento dos valores das coordenadas dos centroides. As classes que possuem mais de um ponto passa por um novo cálculo, onde obtêm-se novo valor dos centroides, fazendo-se a média de cada um dos atributos de todos os pontos que pertencem a esta classe.

5. **Repetir até a convergência:** Neste último passo o algoritmo volta ao segundo passo, onde repete-se interativamente o cálculo das coordenadas dos centroides.

Logo, o **K-means** é um tipo de agrupamento que utiliza uma abordagem *hard*, o agrupamento é definido pelo centro da massa de seus membros. Para a sua execução é necessário se ter um conjunto inicial de agrupamento, com ações de sequências iterativas, para que o algoritmo venha convergir é necessário se realizar diversas iterações.

2.3.2.2 DBSCAN

Nos agrupamentos realizados pelo algoritmo DBSCAN, os grupos são separados em duas áreas, uma de baixa densidade e outra de alta densidade. Este procedimento diferencia os grupos formados pelo DBSCAN do K-means.

O DBSCAN trabalha apenas com as amostras principais, ou seja, amostras que estão presentes nas áreas de alta densidade. Aqui o grupo é dito ser uma amostra central, onde estas encontram-se próximas umas das outras de acordo com uma determinada distância, um outro grupo representa as amostras não centrais (ESTER et al., 1996).

Os parâmetros ***min-samples*** e ***eps*** são os parâmetros os quais definem a densidade dos grupos, estes parâmetros são inversamente proporcionais, quando deseja-se indicar a maior densidade necessária para formar os grupos.

Passos para se realizar o agrupamento pelo algoritmo DBSCAN

Uma das maneiras de se iniciar o processo de agrupamento, é o algoritmo inserir o **K** pontos (centroides) aleatórios iniciais. Desta forma pode-se começar as iterações e encontrar os possíveis resultados.

1. Primeiramente é necessário definir os parâmetros de ***min-samples*** e ***eps***.

2. O DBSCAN procura pelos grupos observando a vizinhança de cada ponto no conjunto de dados.

3. Se o parâmetro *eps* é maior que o parâmetro *min-sample*, um novo grupo é criado com objetos núcleos.

4. O DBSCAN iterativamente encontra os objetos que podem ser diretamente atingidos pela densidade a partir dos objetos núcleos.

5. O processo termina quando nenhum novo ponto pode ser adicionado ao algum grupo.

Logo, o **DBSCAN** os *cluster* observados por este algoritmo são de alta densidade separado por áreas de baixa densidade, os *clusters* encontrados pelo presente algoritmo podem ser representados de diversas formas diferentemente do k-means que diz que os *cluster* são vistos de forma convexa. Este algoritmo utiliza os conceitos de amostras de núcleo, que são amostras presentes em áreas de alta densidade.

2.3.2.3 Birch

Neste algoritmo de agrupamento uma árvore é construída, tal árvore é denominada árvore de características (CFT). Os dados utilizados neste algoritmo tem suas perdas compactadas a um conjunto de nós denominado *Characteristic Feature nodes* (CF Nodes). Estes nós apresentam um valor de subgrupos e estes subgrupos quando estão localizados nos nós podem ter CF nodes filhos (SCIKIT-LEARN, 2018).

Os subgrupos possuem a característica de armazenação de informações nos grupos, as informações possuem as seguintes características.

- Número de amostras em um subgrupo;
- Soma Linear - Um vetor n-dimensional que contém a soma de todas as amostras;
- Soma quadrada - Soma da norma L2 quadrada de todas as amostras;
- Centróides - Para evitar o recálculo de soma linear /n-amostras;
- Norma quadrada dos centróides.

Este algoritmo possui dois parâmetros, sendo eles o limiar que limita a distância entre a amostra inserida e os subgrupos existentes e o fator de ramificação que limita os subgrupos em um nó.

Logo, o **BIRCH** (do inglês, *Balanced Iterative Reducing and Clustering Using Hierarchies*) é um algoritmo de agrupamento que resume-se em três valores ou parâmetros sendo eles o número de pontos, a soma linear dos pontos e a soma quadrada dos pontos, desta forma é possível reduzir o espaço de memória RAM ocupado pelo funcionamento do algoritmo, tornando o cálculo das medidas mais rápido.

2.3.2.4 *Agglomerative Clustering*

O ***Agglomerative Clustering*** cria grupos aninhados, incorporando ou dividindo sucessivamente. A hierarquia do grupo é representada em formato de árvore ou dendograma.

A raiz da árvore representa um único conjunto que reúne todas as amostras, sendo que as folhas da árvore representam os grupos com apenas uma amostra.

Nesta etapa de utilização de algoritmos de agrupamento, necessitou-se analisar cada uma das características dos algoritmos abordados. Um breve resumo dos algoritmos pode ser observado na Tabela 1.

Tabela 1 – Uma comparação dos algoritmos de agrupamento em *Sckit -learn.*

Método	Parâmetros e Escala	Uso	Métrica
K-Means	Número de Cluster Escala: Muito grande n_samples, Médio: n_clusters com código Minibatch	Uso geral, grupos de mesmo tamanho, geometria plana, poucos grupos	Distância entre pontos
DBSCAN	Tamanho da vizinhança Escala: Muito grande n_samples, Médio:n_clusters	Geometria não plana. Tamanhos de agrupamentos desiguais	Distância entre os pontos mais próximos
Birch	Fator de ramificação, limiar, cluster global (opcional) Escala: Grande:n_clusters e n_samples	Grande conjunto de dados, remoção de valores aberrantes, redução dos dados	Distância euclidiana entre os pontos
Agglomerative Clustering	Número de Cluster, tipo de ligação, distância Escala: Grande: n_clusters e n_samples	Muitos grupos, possivelmente restrições de conectividade, distâncias não euclidianas	Qualquer distância pairwise

Os algoritmos de agrupamento têm o objetivo de dividir um conjunto de objetos em grupos de acordo com sua similaridade, ou seja, objetos similares pertencem a um mesmo grupo, objetos não similares pertencem a um outro grupo (MANNING; SCHÜTZE, 1999).

Os objetos são representados e agrupados por um conjunto de atributos e de valores, sem informação de sua classe ou categoria. Estes objetos são:

- Agrupamento de documentos de textos ou similares;
- Identificação de grupos em redes sociais;
- Segmentação de cliente;
- Identificação de plantas com características similares.

2.4 Word Embeddings

Recentemente utiliza-se a técnica do Word2vec (MIKOLOV et al., 2013) composto por uma arquitetura de redes neurais, implantado inicialmente por Tomas Mikolov do Google, para aprender *embeddings* de palavras (BENGIO et al., 2003), que são representadas por um vetor denso.

Word2vec tem como objetivo construir uma representação vetorial para palavras em um texto, de forma não supervisionada, onde o algoritmo associa cada palavra do texto a um vetor correspondente chamado de Wordvec. A ideia desta técnica é que palavras que aparecem em contextos similares dentro de uma coleção de documentos, ou seja, dentro dos textos, sejam representadas por vetores próximos (MIKOLOV et al., 2013).

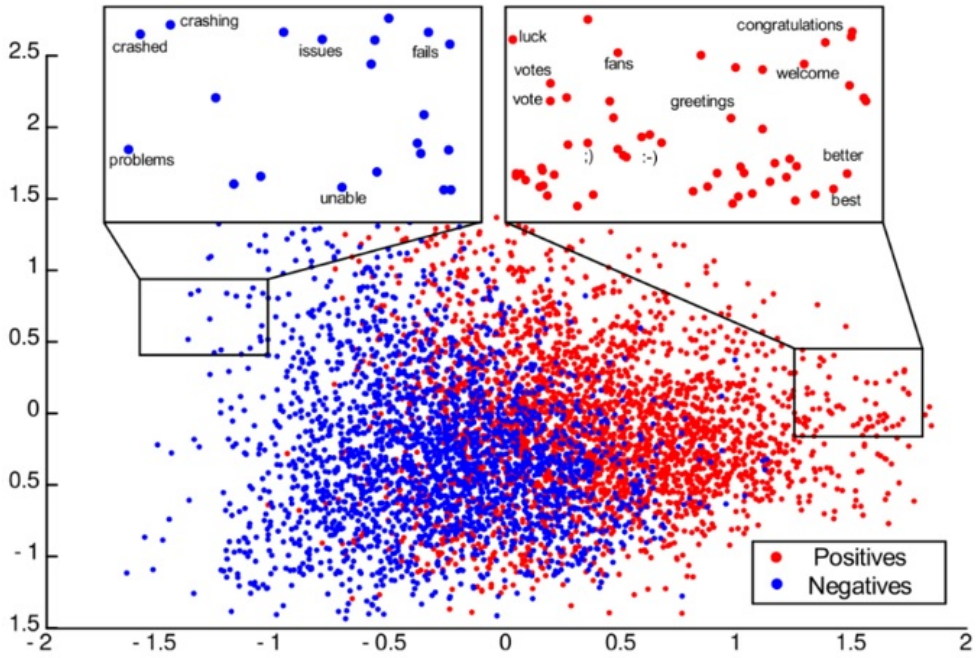


Figura 8 – Exemplo de palavras positivas e negativas no espaço dos Wordvecs.
Fonte: (COLYER, 2016)

A Figura 8, apresenta a vetorização das palavras utilizando o Word2Vec. Aqui palavras semanticamente semelhantes ficam mais próximas umas das outras no espaço de representação vetorial, em contrapartida, palavras semanticamente diferentes ficam mais distantes.

Assim é possível observar que a partir do *corpus* de entrada (os textos), é produzido um espaço vetorial com "n" dimensões, os vetores são posicionados de acordo com a aproximação das palavras e da semelhança entre os contextos. Algumas relações entre palavras podem ser observadas por meio do Word2vec, onde a relação entre duas palavras fornece informações sobre a relação entre duas outras palavras. Como exemplo temos que a distância entre "homem" e "mulher" possui uma distância próxima a "rei" e "rainha".

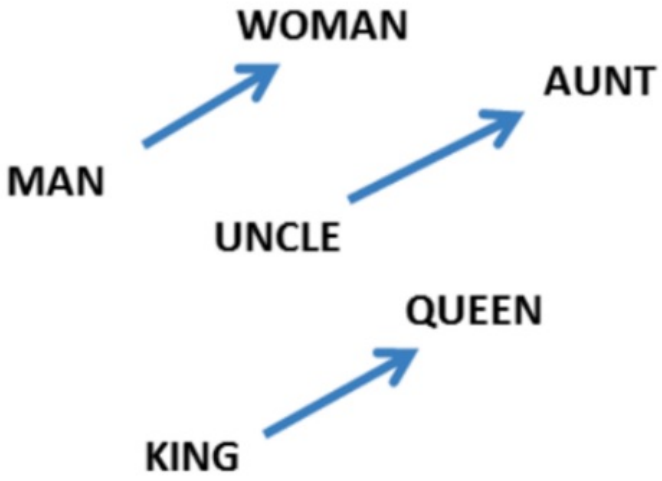


Figura 9 – Exemplo de relação entre duas palavras no Word2vec.
Fonte: (COLYER, 2016)

A Figura 9, apresenta um exemplo da relação entre duas palavras onde o vetor ("*King*") – vetor ("*Man*") + vetor ("*Woman*") = vetor ("*Queen*") (MIKOLOV et al., 2013). Aqui ocorre a subtração do vetor para a palavra homem da palavra rei e adiciona o vetor da palavra mulher. Este é o vetor que mais se aproxima da representação vetorial da palavra rainha, além destas, relações de formas automáticas de tradução entre as palavras podem ser realizadas, como é o caso vetores das palavras em inglês serem similares ou próximos aos vetores das palavras em espanhol, desta forma pode-se obter sequências de eventos e suas similaridades, como pode ser observado na Figura 10.

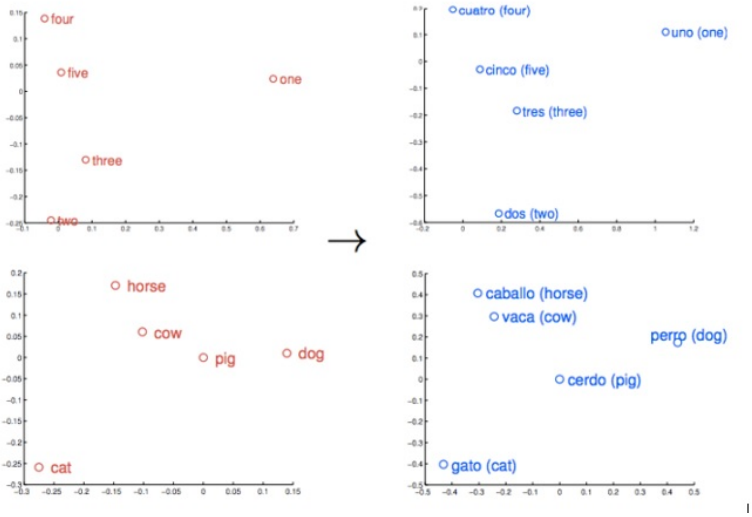


Figura 10 – Exemplo de relação de forma automática de tradução no Word2vec.
Fonte: (COLYER, 2016)

Existem dois modelos de rede neural que exemplificam os procedimentos realizados pelo o word2vec, onde são capazes de extrair padrões a partir de sequências sendo eles o Skip-gram, que estima a probabilidade de um determinado contexto dada uma palavra, e o CBOW que prevê a palavra atual com base no contexto. Esses modelos utilizam uma camada oculta que representa os *embeddings*. Um exemplo do modelo Skip-gram será demonstrado pelo fato deste produzir resultados mais precisos em grandes coleções de documentos.

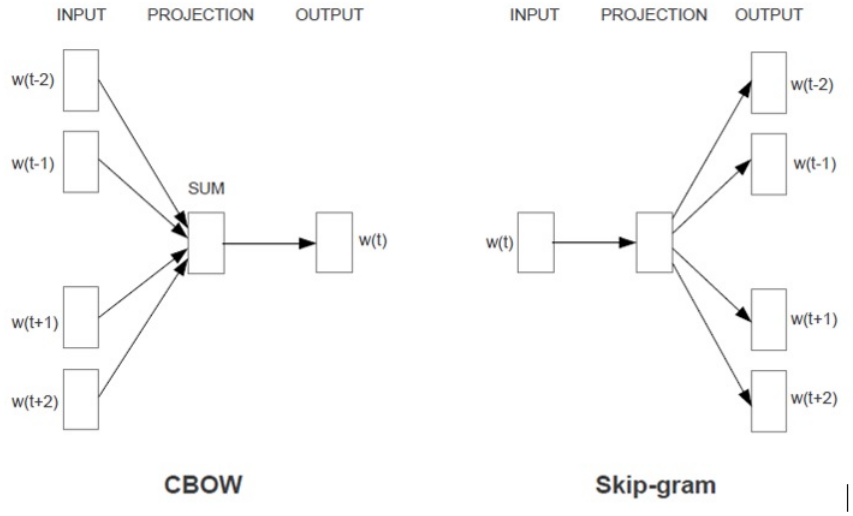


Figura 11 – Arquitetura CBOW prevê a palavra atual com base no contexto, e o modelo Skip-gram prevê palavras circundantes dada a palavra atual.
Fonte: (MIKOLOV et al., 2013)

Na Figura 11, observa-se então um exemplo do modelo contínuo Skip-gram, onde são apresentadas algumas das amostras de treinamento (pares de palavras) e contexto definidas inicialmente na base de dados. Desta forma como exemplo utilizou-se a frase "Fast brown fox jumps over the lazy dog".

A camada oculta é representada vetorialmente pelo *input* como pode ser observado na Figura 12, ou seja, por uma "palavra de entrada" (a palavra destacada em azul), que também será usada na camada de saída (*output*), para prevê uma palavra do contexto. Esta "palavra de entrada" é escolhida ao acaso, a rede neural irá mostrar a probabilidade de cada palavra do vocabulário ser a palavra próxima que se quer encontrar. Na Figura 12 apresenta-se um exemplo, o tamanho de janela utilizado é 2, o tamanho da janela serve como parâmetro de distância para mostrar quais palavras serão próximas a palavra de entrada, como a janela escolhida é de tamanho 2, serão observadas 2 palavras atrás e 2 palavras à frente da palavra de entrada.

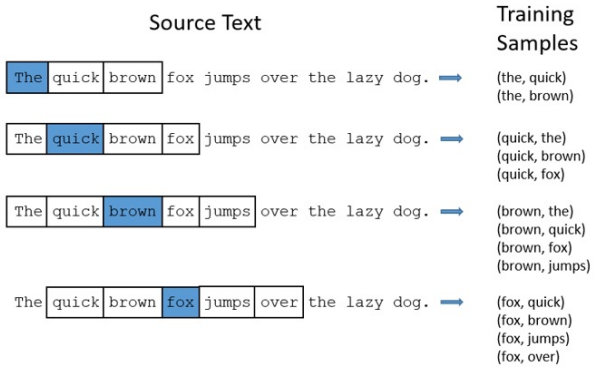


Figura 12 – Exemplos de tamanho de janelas.
Fonte: (MCCORMICK, 2016)

As probabilidades de saída serão relacionadas com a probabilidade de encontrar cada palavra do vocabulário próxima à palavra de entrada, ou seja, a rede aprende os valores das estatísticas a partir da frequência de cada emparelhamento, quando o treinamento é concluído pode-se observar que dada a palavra de entrada, outras palavras surgirão sendo algumas delas mais relacionadas ou de maior probabilidade com a palavra de entrada do que outras.

A Figura 13, apresenta uma rede neural que é construída a partir de um vocabulário de 10.000 palavras únicas. Uma palavra é escolhida como "palavra de entrada", por exemplo "ants" (formigas). O vetor desta rede terá 10.000 componentes, ou seja, um vetor para cada palavra do vocabulário, coloca-se "1" na posição correspondente à palavra "ants" (palavra de entrada) e "0" para todas as outras posições. A saída da rede é um vetor único com 10.000 componentes, que possui uma probabilidade para cada palavra escolhida aleatoriamente do vocabulário. A rede neural treinada é representada pela seguinte arquitetura:

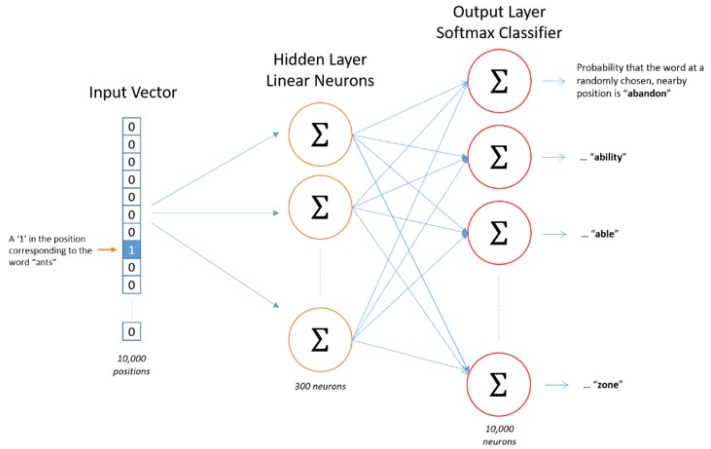


Figura 13 – Arquitetura da Rede Neural.
Fonte:(MCCORMICK, 2016)

Para encontrar esta probabilidade é necessário utilizar a função de ativação Softmax

utilizada em redes neurais de classificação, isto corre apenas nos neurônios de saída (*output*), onde a função força a rede neural apresentar a probabilidade dos dados das classes definidas.

Ao executar o treinamento desta rede, observa-se que o *input* e o *output* são simplesmente um vetor *one-hot-enconding*, onde o *input* é a palavra que se quer inferir o contexto, e o *output* é uma distribuição de probabilidade, com diversos valores um para cada contexto. Logo, esta rede é vista como um acerto do contexto a partir das palavras, ou seja, o contexto é próximo a "palavra de entrada", utilizando os métodos usuais de redes neurais como (*SGD, Dropout, etc.*) e técnicas usadas no contexto de Word2vec como (*hierarchical softmax, negative sampling*, etc.) os word2vec encontrados na camada oculta é visto quando este processo é finalizado, onde na matriz de pesos desta camada, cada uma das linhas inclui a representação vetorial de uma palavra, isto se dá pelo fato da entrada ser um vetor *one-hot*.

Para o exemplo desta dissertação utilizou-se uma camada oculta como pode ser visto na Figura 14, a que representa os *embeddings* com 300 recursos o mesmo utilizado no conjunto de dados de notícias do Google, desta forma a camada oculta será representada por uma matriz de pesos (vetores de palavras aprendidos) com 10.000 linhas e 300 colunas, uma para cada neurônio escondido. O número de recursos e um "hiper-parâmetro" são necessários para testar diversos valores diferentes para se obter o melhor resultado.

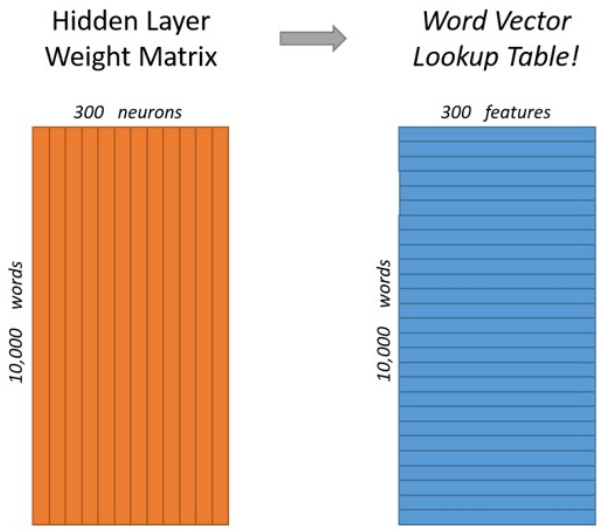


Figura 14 – Matriz de peso da camada oculta.
Fonte: (MCCORMICK, 2016)

Então o objetivo deste exemplo é aprender essa matriz de peso da camada oculta.

Ao multiplicar 1 x 10.000 vetores *one-hot* por uma matriz de 10.000 x 300 este processo irá efetivamente apenas selecionar a linha da matriz correspondente ao "1", ou seja, selecionará apenas a linha da "palavra de entrada". Um pequeno exemplo disto é apresentado na Figura 15:

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Figura 15 – Matriz de peso da camada
Fonte:(MCCORMICK, 2016)

Desta forma é possível observar que a camada oculta do modelo está funcionando corretamente como uma tabela de pesquisa, onde a saída da camada oculta é apenas o "vetor de palavras" para a palavra de entrada escolhida.

Os neurônios de saída apresentam um vetor de peso que é multiplicado pelo vetor de palavras da camada oculta, que emprega a função *exp(x)* ao resultado. Os resultados obtidos em 1 é realizado a partir da divisão dos resultados pela soma dos resultados de todos os 10.000 nós de saída.

O cálculo da saída do neurônio de saída para a palavra "carro" é dada da seguinte forma como pode ser visto na Figura 16:

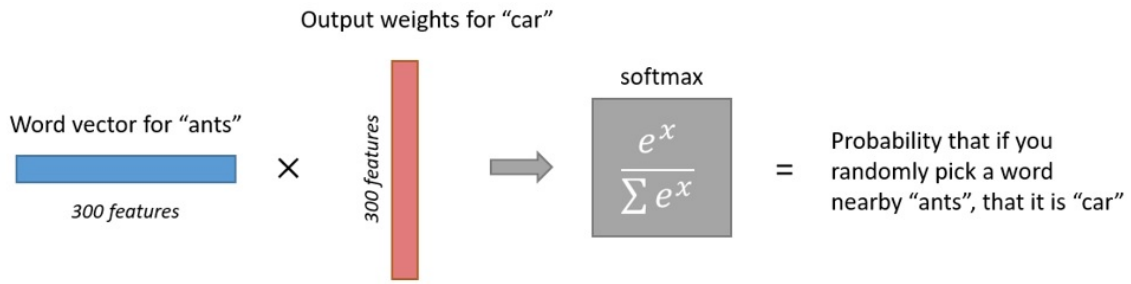


Figura 16 – Saída do neurônio.
Fonte: (MCCORMICK, 2016)

Glove

Global Word Vectores (GLOVE) (PENNINGTON; SOCHER; MANNING, 2014) tem uma forma de aprender *embeddings* diferente do Word2vec, nesta técnica não são utilizadas as redes neurais de forma preditiva e sim a matriz de co-ocorrência de palavras e contextos, onde esta é fatorada em uma matriz dimensional inferior, a fatoração é uma característica da representação de cada palavra extraída no vocabulário.

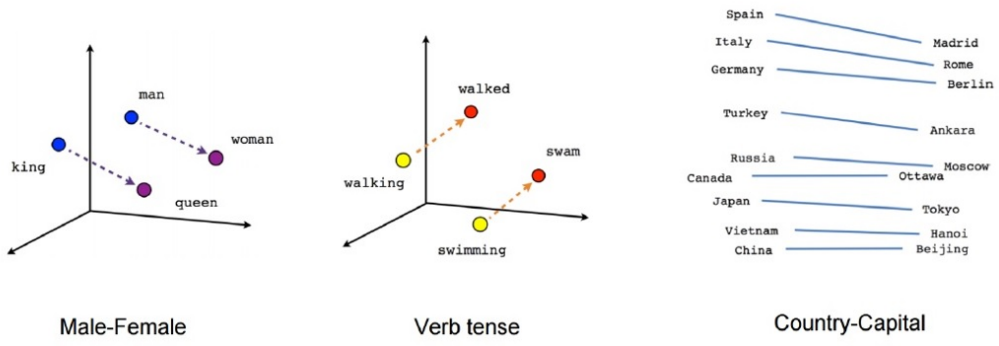


Figura 17 – Exemplos de relações lineares entre palavras que podem ser encontradas no word2vec.
Fonte:(TENSORFLOW. . . , 2016)

A Figura 17, apresenta as relações semânticas das palavras por meio das direções no espaço vetorial, por exemplo, a relação entre sexo masculino, feminino, verbo e até mesmo relações país-capital entre palavras.

A vetorização de palavras é caracterizada pela frequência de palavras em uma base de dados, onde estas frequências são estatísticas de ocorrência que fornecem informações de métodos não supervisionados de representação.

As palavras de modelos semânticos de espaços vetoriais são representadas por um vetor de valor real, estes vetores podem ser utilizados para recuperar informações (MANNING et al., 2008), classificação do documento (SEBASTIANI, 2002), reconhecimento de entidade nomeada (TURIAN; RATINOV; BENGIO, 2010), respondendo perguntas (TELLEX et al., 2003) e análise (SOCHER et al., 2013).

Os métodos para se criar vetores de palavras em sua maioria são dados pela distância dos pares de vetores de palavra ou seu ângulo sendo este o método de avaliação de qualidade para representação de palavras. Um novo método de avaliação foi proposto recentemente por (MIKOLOV; YIH; ZWEIG, 2013), onde a representação das palavras é realizada por meio de um esquema baseado nas analogias das palavras, observando não mais a distâncias dos pares de palavras e sim suas diferentes dimensões.

Desta forma é possível observar que para se aprender os vetores de palavras os dois principais métodos são:

1. Método global de fatoração de matriz, por exemplo a análise semântica latente (LSA) (DEERWESTER et al., 1990);

2. Método das janelas de contexto local usadas no modelo skip-gram (MIKOLOV; YIH; ZWEIG, 2013).

A fatoração de matriz gera uma baixa dimensão na representação de palavras. O método utiliza uma baixa classificação das palavras e decompõe as grandes matrizes por meio da captura de informações estatísticas observada na base de dados. Estas informações variam de acordo a aplicação estabelecida. No método de LSA as matrizes são do tipo "documento-termo", onde as linhas correspondem as palavras e as colunas correspondem aos diferentes documentos da base de dados.

No método baseado em janelas rasas as quais realizam a representação das palavras baseado nas previsões do contexto local, por exemplo (BENGIO et al., 2003), apresentou um modelo de aprendizagem vetorial a partir da representação de palavras como parte de uma arquitetura de rede neural simples. A rede neural possui uma completa estrutura para aprender representações de palavras úteis.

Os modelos criados por Mikolov, sendo eles o skip-gram e o CBOW, propõem uma arquitetura simples de sua única camada baseada internamente nos produtos entre dois vetores de palavras e aprendem padrões linguísticos como relação linear entre os vetores de palavras.

A desvantagem do modelo baseado em janelas rasas é que este não utiliza as estatísticas de co-ocorrência da base de dados, e sim utilizam as janelas em todo a base de dados, mas esta técnica não se mostra representativa pelo fato do mal aproveitamento na grande quantidade de repetição dos dados.

Os dois métodos possuem desvantagens, o LSA utiliza as informações estatísticas, porém não se mostram eficaz na relação de semelhança entre as palavras, onde indicam um espaço vetorial sub-ótimo. O skip-gram apresentam-se de forma contrária do LSA (PENNINGTON; SOCHER; MANNING, 2014).

2.5 Redução de Dimensionalidade

A Redução de Dimensionalidade ou de características é o principal problema da classificação de texto. A maioria das bases de dados apresentam-se com dezenas, milhares de características, onde, a maioria destas são irrelevantes ou redundantes para o texto. Desta forma é necessário utilizar meios que reduzam estes problemas para se obter uma melhor classificação das palavras(SEBASTIANI, 2002).

O processo de redução de dimensionalidade melhora o desempenho da classificação de texto e o esforço computacional para execução dos experimentos. Base de dados com elevados números de características quando vetorizadas pelo método *bag-of-words* considera a maioria dos termos do documento, sendo estes redundantes ou não.

A Redução da Dimensionalidade das características é subdividida em seleção de características e transformação ou extração das características(SEBASTIANI, 2002). Ambos os método consiste em eliminar dos documentos palavras com pouco poder significativo, ou seja, palavras redundantes e ou irrelevantes que não acrescentam em nada a compreensão do texto.

O método de seleção de características trabalha selecionando um subconjunto formado apenas pelas características mais relevantes do conjunto de documento original, deste modo, este método não altera as características originais. O método então ao analisar apenas as características relevantes ele consequentemente desprezas todas as outras palavras dos documentos (PINHEIRO; CAVALCANTI; REN, 2015).

Para o método de transformação de características, a redução da dimensionalidade ocorre de forma que, são criados novos conjuntos de características, sendo estes menores que o conjunto de característica original. Nestes novos conjuntos novos termos gerados a partir da combinação ou transformação dos termos originais (PINHEIRO; CAVALCANTI; REN, 2015).

3 Trabalhos Relacionados

O presente Capítulo detalha os principais trabalhos que tratam de problemas similares ao desta dissertação, o objetivo é entender e levantar os diversos problemas distribuídos na arquitetura e analisar as diversas propostas e aplicações que façam uso deste mecanismo para reduzir a dimensionalidade das características. Este capítulo satisfaz ao primeiro objetivo específico 1.1.2 visto nesta dissertação. Aqui são apresentados trabalhos que fizeram uso dos algoritmos de classificação de texto e são detalhado os métodos utilizados por cada um dos trabalhos para reduzir a dimensionalidade das características, as bases de dados utilizadas são as mesmas tanto para os trabalhos relacionados quanto para a presente proposta. Por fim, algumas conclusões e comparações são apresentadas em forma de uma tabela que resumi as abordagens empregadas em cada um dos trabalhos.

Roberto e seus colaboradores (PINHEIRO; CAVALCANTI; REN, 2015) propuseram dois métodos de filtragem para seleção de recursos em classificação de textos. O método máximo f características por documento (MFD, do inglês *Maximum f Features per Document*) que analisa todos os documentos para garantir que cada documento na formação do conjunto é representado por um vetor de característica final e o (MFDR, do inglês *Maximum f Features per Document - Reduced*) que apresenta a redução f por documento. No MFDR são analisadas apenas os documentos com alta dimensionalidade (FEF, do inglês Feature Evaluation Function) que é conhecido como métodos de filtragem de características por meio de ranqueamentos utilizando algoritmos determinísticos e métricas estatísticas conhecidas como funções de avaliação de características. Ambos os algoritmos determinam o número de recursos selecionados f de modo orientado a dados, usando um*ranking* global do recurso de função de avaliação (FEF). Este trabalho utilizou o classificador Naïve Bayes (NB) para a realização dos seus testes, por este classificador assumir atributos que são condicionalmente independentes. Ambos os métodos foram comparados com o método (ALOFT, do inglês, *At Least One Feature*) que em seu treinamento pelo menos um recurso é observado, ou seja garante a contribuição dos termos de cada documento para o subconjunto final.

De acordo com Labani, Moradi, Ahmadizar e Jalili (LABANI et al., 2018) um método de filtragem de seleção de recursos foi proposto, conhecido como método de seleção de recurso multivariada, chamado Critério de discriminação relativa multivariada (MRDC, do inglês *Multivariate Relative Discrimination Criterion*). Este método é utilizado na classificação de textos e visa considerar a relevância e reduzir os conceitos de redundância em seu processo de avaliação das características. A redundância é analisada através do conceito de redundância de mínima e máxima relevância. O método proposto não só seleciona as características com máxima relevância, mas também a redundância entre

eles levando em consideração a métrica de correlação. Os algoritmos de classificação de texto utilizados neste trabalho são: o *Multilayer Perceptron* (MLP) que é um feed-forward artificial de rede neural este classificador é formado por uma camada de entrada, possui pelo menos uma camada escondida e uma camada de saída. O classificador probabilístico Multinomial Bayes (MNB, do inglês *Multinomial Naïve Bayes*) os recursos de entrada deste classificador são independentes de outro dado a classe de destino, e por fim o classificador conhecido como árvore de decisão (DT, do inglês *Decision Tree*), este classificador cria padrões que prevê o valor de uma variável de destino por meios de dados de treinamento e aprende as regras simples de decisão do dados.

(BHARTI; SINGH, 2015), utilizou em seu trabalho uma abordagem híbrida, que engloba diferentes aspectos completos de recurso de relevância para seleção de subconjunto de recursos de características, recebendo uma atenção considerável ao problema de alta dimensionalidade. Tradicionalmente, união ou interseção é usada para mesclar sublistas de recurso selecionadas com diferentes métodos. O método proposto é uma abordagem modificada do método da união. Esta abordagem se aplica a União, por meio da seleção das características que estão no topo do *ranking* de características e aplica-se a interseção nas sublistas de características restantes. Deste modo a seleção das características que estão no topo do*ranking* é garantida, como também as características comuns identificadas pelo método, sem que estas aumentem a o tamanho da dimensão no espaço de características. Neste trabalho são utilizados para o cálculo da pontuação de relevância os métodos de seleção de recursos termo variância (TV, do inglês *Term Variance*), TV atribui uma pontuação de relevância a recursos baseado no valor de seu desvio médio e frequência de documento (DF, do inglês *Document Frequency*), DF atribui a pontuação do recurso com base no número de documentos abrangidos, ou seja, determinado termo possui uma pontuação de alta relevância e isto contribui para abrangência de um número maior de documentos. Estes dois métodos de seleção de recursos atribuem pontuações de relevância a cada recurso observado. Em seguida, uma análise de componentes principais de método de extração característica (PCA, do inglês *Principal Component Analysis*) é realizada, onde é aplicada para reduzir ainda mais as dimensões no espaço recurso sem perder muita informação.

(UYSAL, 2016) em seu trabalho intitulado "*An improved global feature selection scheme for text classification*" utilizou um esquema de seleção de características global melhorado (IGFSS, do inglês *Improved Global Feature Selection Scheme*), este esquema altera o último passo da seleção de recursos, afim de obter um conjunto de recursos mais representativo. O método IGFSS melhora o desempenho de classificação dos métodos de seleção de recursos globais criando um conjunto de recursos que representa todas as classes igualmente. Os algoritmos de classificação utilizados neste trabalho foram o *Support Vector Machine* (SVM) e Naïve Bayes (NB).

Os colaboradores de "*Chi-square Statistics Feature Selection Based on Term Frequency and Distribution for Text Categorization*" (JIN et al., 2015) utilizaram um método de seleção de recursos modificado, conhecido como estatística qui-quadrado (CHI), esta também pode ser chamada de distribuição de frequência do termo abordagem CHI. O método proposto mostrou-se eficiente quando comparado ao recurso clássico dos métodos de seleção em termos de Macro-F1 e Micro-F1, o algoritmo de classificação utilizado nesta abordagem foi o (K-NN, do inglês *K-Nearest Neighboors*), por ser um classificador de simples aplicação como também um classificador interessante para avaliar o desempenho dos métodos de características, pela influencia dos termos selecionados.

Em (WANG; WANG; YI, 2010) uma abordagem diferente das citadas anteriormente foi realizada para classificar documentos de texto, a entropia máxima (ME, do inglês *Maximum Entropy*), o ME consiste em determinar a redução da dimensionalidade das características utilizando probabilidades, o melhor modelo para determinados dados é aquele que maximiza a entropia dos conjuntos de distribuições de probabilidade. Esta abordagem apresentou vantagens quando comparada com os algoritmos de classificação de texto, tais como (NB) e (SVM), que são dois algoritmos populares e precisos no domínio da classificação de texto.

(ROUSSEAU; KIAGIAS; VAZIRGIANNIS, 2015) trataram a classificação de texto como um problema de classificação de gráfico. Onde os documentos textuais são representados como um gráfico de palavras. Esta abordagem utilizou o algoritmo de classificação SVM em seus experimentos, e os resultados obtidos a partir do gráfico de palavras mostrou-se estatisticamente significativo e com maior precisão em Macro-F1-Pontuação quando comparado a média da base.

Rogério (FRAGOSO, 2016) em seu trabalho utiliza a técnica de filtragem com o intuito de melhorar o desempenho de classificação em comparação com os métodos atuais e de tornar possível a automatização da escolha do tamanho do conjunto final de características, o método utilizado na seleção de características é o (cMFDR, do inglês *Category-dependent Maximum f Features per Document*), este método define um limiar para cada categoria para determinar quais documentos serão considerados no processo de seleção de características. O cMFDR aprimora o desempenho visto no método (MFDR, do inglês *Maximum f Features per Document-Reduced*) por meio da seleção de características. O método MFDR, seleciona f características por cada documento e satisfaz a condição baseada em sua relevância (DR, do inglês *Dimensionality Reduction*). Outro método considerado neste trabalho é o (AFSA, do inglês *Automatic Features Subsets Analyzer*) este método introduz um procedimento para determinar de maneira guiada por dados, o melhor subconjunto de características dentre um número de subconjuntos gerados. O algoritmo de classificação utilizado para análise dos dados foi o classificador Naïve Bayes Multinomial (MNB, do inglês *Multinomial Bayes*). Em Naïve Bayes Multinomial o modelo

utiliza as ocorrências dos termos selecionados do vocabulário nos documentos.

A Tabela 2, mostra algumas comparações entre os trabalhos relacionados e a presente dissertação, algumas diferenças podem ser percebidas, como a utilização do algoritmo de classificação *Random Forest* e SVM, alguns trabalhos utilizam o SVM, no entanto esta dissertação se difere dos demais por usar como classificadores o *Random Forest* e o SVM, a escolha desses dois classificadores foi dada pelo fato destes serem os mais recomendados na literatura para a classificação de texto (FERNÁNDEZ-DELGADO et al., 2014). Uma outra diferença em relação aos trabalhos relacionado é que neste trabalho a redução das características foi dada por meio da transformação, onde a partir dos grupos de características originais, grupos menores de características foram criados sem que ocorresse perda de informação. Uma outra característica que diferencia os trabalhos é que alguns deles utilizaram como representação vetorial de suas palavras o método de *bag-of-words*, já nesta dissertação a abordagem se deu em relação a representação vetorial pelo grupos semânticos e isto torna-se o grande diferencial deste trabalho em relação aos trabalhos já vistos na literatura.

Tabela 2 – Comparação dos trabalhos relacionados

Referências	**Classificadores utilizados**	**Redução de características**	**Representação das características**
PINHEIRO et.al, 2015	NB	Seleção	*bag of words*
LABANI et.al, 2018	MLP, MNB e DT	Seleção	*bag of words*
BHARTI et.al, 2015	SVM e NB	Seleção	*bag of words*
UYSAL, 2016	SVM e NB	Seleção	*bag of words*
JIN et. al, 2015	KNN	Seleção	*bag of words*
WANG et.al, 2010	Entropia Máxima	Seleção	*bag of words*
ROUSSEAU et.al, 2015	SVM	Seleção	*graph of words*
FRAGOSO, 2016	MNB	Seleção	*bag of words*
PROPOSTA	SVM e RF	Transformação	Grupos Semânticos

4 Método

Conforme a proposta desta dissertação, um método de criação de grupos semânticos é proposto para reduzir a dimensionalidade das características, a partir da utilização de algoritmos de agrupamentos e *Word Embeddings*, onde este grupo semântico é formado por grupos menores de características sem perder informações de seus dados, ou seja, os grupos de características são reduzidos a tamanho menores que o original, sem que estes percam suas informações. Necessitou-se realizar, para a execução deste experimento duas etapas, sendo elas a etapa de treinamento e a etapa de classificação. As próximas seções irão detalhar os processos de criação de grupos semânticos e classificação.

4.1 Criação de grupos semânticos

Na etapa inicial, a de criação de grupos semânticos, neste trabalho denominada etapa de treinamento, realizou-se a representação dos documentos utilizando as técnicas de pré-processamento, *tokenização* e remoção de *stopwords*. Após este procedimento foram criados vetores de palavras, ou seja, as palavras foram transformadas em vetores que representam os documentos e por fim criou-se os grupos de características, através da utilização dos algoritmos de agrupamentos.

A Figura 18 apresenta uma arquitetura geral da etapa de treinamento realizada no presente trabalho.

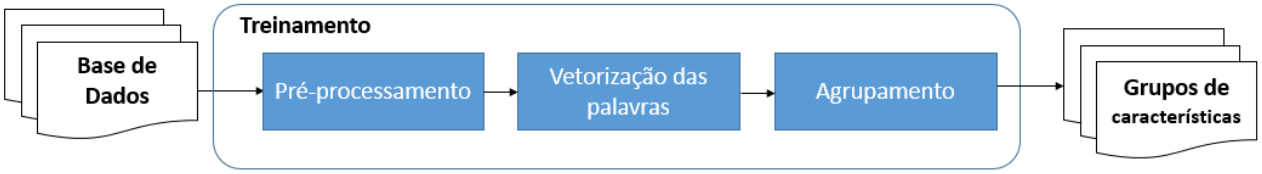


Figura 18 – Arquitetura geral do processo de treinamento.

4.1.1 Representação do documento

Para obter uma classificação satisfatória, ou seja, uma classificação onde não existam palavras com pouco poder representativo no documento, deve-se realizar um bom tratamento nas bases de dados. Os textos que constituem essas bases são obtidos de forma bruta, com erros ortográficos, caracteres irrelevantes, formatações indesejáveis entre outros.

Logo, algumas técnicas devem ser aplicadas para resolver estes problemas, iniciando-se pela fase de pré-processamento, onde os textos recebem tratamento e são transformados em vetores de características, posteriormente o agrupamento das palavras é realizado pelos

diferentes algoritmos empregados neste trabalho e finalmente é iniciada a classificação das palavras.

Para os procedimentos de tratamento da base utilizou-se o Python versão 3.6.4, uma linguagem de programação de fácil interação para manipular arquivos de textos. Na fase do pré-processamento utilizou-se juntamente com o Python uma biblioteca de código aberto a *Natural Language Toolkit* (NLTK) (NLTK, 2016), que possui vários recursos como: classificação, *tokenização*, *stemming*, *tagging*, *parsing* e raciocínio semântico, tais recursos são utilizados para análise de textos.

Na Figura 19, é apresentado um exemplo de um código que forma uma árvore semântica, com as classes de cada palavra, ou seja, se a palavra é uma preposição, artigo, verbo ou um nome próprio. A partir da identificação destas categorias combinada com outras técnicas é possível identificar a semântica das frases.

```
>>> from nltk.corpus import treebank
>>> t = treebank.parsed_sents('wsj_0001.mrg')[0]
>>> t.draw()
```

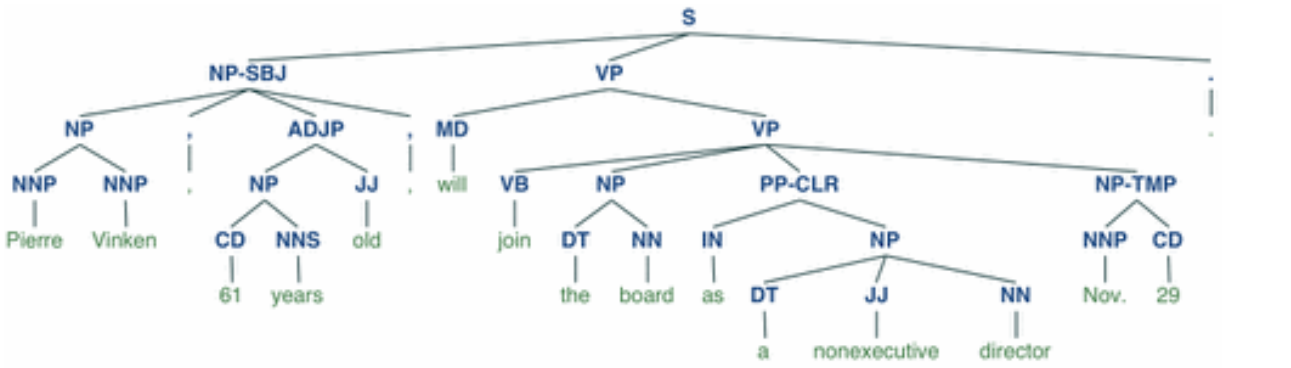


Figura 19 – Exemplo de uma árvore semântica com as classes de cada palavra.
Fonte:(Edward Loper; Ewan Klein, 2009)

4.1.2 Pré-processamento

O pré-processamento tem como objetivo organizar, tratar as informações de um texto e então prepara-las para análise. É possível identificar e resolver nesta etapa, problemas presentes nas bases de dados como é o caso do grande número de palavras com pouco poder representativo, a inserção de caracteres especiais e termos muito frequentes nos documentos. Diante estes fatores algumas técnicas são utilizadas para resolver estes problemas antes de sua indexação.

- ***Tokenização:*** Consiste em separar um texto em *tokens* ou sentenças, ou seja, decompõe o documento pelos termos que o constitui. Na tokenização são removidos

também os caracteres especiais, essa remoção provoca perda de informação em algumas palavras, como também o seu significado um exemplo desta perda são as palavras que utilizam o hífen.

Utilizando um dos documentos que compõem as bases de dados em estudo, serão apresentados exemplos de *tokenização* e eliminação de *stopwords*. As sentenças em destaque por um retângulo em azul, como apresentado na Figura 20, serão utilizadas como exemplos.

Figura 20 – Exemplo de um texto da base de dados.

Exemplo 1: Separando o texto em *tokens* e em sentenças.

"Subscriptions to 300 mln guilders were met in full, amounts above 300 mln at 50 pct."

Separação por tokens: ['Subscriptions' 'to' 'mln' 'guilders' 'were' 'met' 'in' 'full' 'amounts' 'above' 'mln' 'at' 'pct']

Separação por sentenças: ['Subscriptions to mln guilders were met in full amounts above mln at pct']

"Dealers expect this week's money market shortage to be around 12 billion guilders."

Separação por tokens: ['Dealers', 'expect', 'this', 'week', 's', 'money', 'market', 'shortage', 'to', 'be', 'around', 'billion', 'guilders']

Separação por sentenças: ['Dealers expect this week s money market shortage to be around billion guilders']

- **Remoção de *stopwords*:** Retira dos textos as palavras irrelevantes como preposições, artigos por exemplo (*o, a, um, uma*), advérbios, conjunções e outras classes de palavras, elimina também palavras muito frequentes nos documentos onde estas não acrescentam valor significativo a classificação.

  Exemplo 2: Removendo termos de pouca relevância.

  "Subscriptions to 300 mln guilders were met in full, amounts above 300 mln at 50 pct."

  ['amounts', 'pct', 'mln', 'guilders', 'subscriptions', 'met', 'full' '.']

  "Dealers expect this week's money market shortage to be around 12 billion guilders."

  ['Around', 'shortage', 'expect', 'billion', 'guilders', 'dealers', 'money', 'week', 'market', '.']

4.1.3 Vetorização das palavras

Após a etapa de pré-processamento onde realizou-se o processo de remoção de *stopwords* e *tokenização* dos documentos, necessitou-se utilizar técnicas, para transformar palavras em vetores. Esta vetorização é dada a partir da similaridade entre as palavras presentes nos documentos. Após a criação dos vetores de palavras, treinou-se a base de dados.

A vetorização tem como objetivo gerar vetores de palavras. Nesta dissertação, esta vetorização foi obtida a partir da utilização do Word2Vec e Glove. Estes são explicados na Seção 2.4 em *Word Embeddings*.

Na vetorização de palavras utilizando o Word2Vec, o algoritmo associa cada palavra do texto a um vetor correspondente, sendo este vetor denominado Wordvec. Aqui palavras de contextos similares são representadas por vetores próximos, ou seja, palavras semelhantes possuem valores semelhantes. A partir da criação dos vetores, algumas operações podem ser efetuadas, por exemplo: obter o valor de um vetor de palavra e comparar com outras palavras. Um outro exemplo clássico desta técnica, é a capacidade de previsão da próxima palavra do contexto. Mais um exemplo de aplicação do Word2Vec é que as palavras podem ser separadas a partir de seu gênero, como é o caso da palavra rei e da palavra rainha, pode-se também verificar a relação entre palavras como é o caso dos verbos no passado e dos verbos em ação, como também é capaz de prever a capital de um país.

O Word2Vec possui uma vetorização de n-dimensões, o que torna a fácil aplicação desta técnica em diversas situações, além disto o Word2Vec está associado ao processamento de linguagem natural (NLP) e tem a vantagem de gerar sentenças, a partir da criação de rotas entre seus vetores.

No Glove a vetorização das palavras é apresentada por meio das frequências de cada uma das palavras da base de dados, a frequência de sua ocorrência fornece informações de sua vetorização. Diferente do Word2Vec o Glove é um modelo baseado em contagem onde se aprende seus vetores, através da redução de dimensionalidade na matriz de co-ocorrência. Logo, tem-se como resultado representações menores da dimensão, nas quais é possível fornecer explicação da maior parte da variação nos dados de alta dimensão.

Neste trabalho o Word2Vec e o Glove foram utilizados em Python para realizar a vetorização das palavras do texto. O procedimento utilizado para aplicação do Word2vec é inserir como entrada as bases de dados compostas por textos e ter como saída um conjunto de vetores, aqui o propósito é agrupar vetores de palavras que sejam semelhantes entre si. O treinamento realizado pelo Glove é baseado em estatísticas que são associadas as co-ocorrências globais das palavras e das palavras em um banco de dados, a saída apresentada pelo Glove são vetores de palavras resultantes das subestruturas lineares dos termos.

Aqui temos um exemplo onde são mostrados os vetores de palavras gerados pelo Word2Vec como mostra a Tabela 3 e Glove na Tabela 4. Após a *tokenização* e eliminação de *stopwords* da frase, os termos restantes foram transformados em vetores de palavras, é possível notar que cada método gera vetores diferentes para cada uma das palavras.

Exemplo 3: “Subscriptions to 300 mln guilders were met in full, amounts above 300 mln at 50 pct."

Tabela 3 – Vetores de palavras realizada pelo Word2Vec.

amounts	1	0,065	-0,020	0,155	0,140	0,092	0,152
full	0,065	1	-0,009	-0,008	0,149	0,141	0,135
met	-0,020	-0,009	1	0,029	-0,023	-0,025	-0,056
guilders	0,155	-0,008	0,029	1	0,060	0,166	0,148
subscriptions	0,140	0,149	-0,023	0,060	1	0,203	0,191
pct	0,092	0,141	-0,025	0,166	0,203	1	0,642
mln	0,152	0,135	-0,056	0,148	0,191	0,642	1

Tabela 4 – Vetores de palavras realizada pelo Glove.

amounts	1	0,330	0,104	0,163	0,239	0,140	0,078
full	0,330	1	0,206	0,008	0,214	-0,100	0,031
met	0,104	0,206	1	-0,055	-0,010	0,045	-0,067
mln	0,163	0,008	-0,055	1	0,206	0,359	0,558
subscriptions	0,239	0,214	-0,010	0,206	1	0,114	0,150
guilders	0,140	-0,100	0,045	0,359	0,114	1	0,071
pct	0,078	0,031	-0,067	0,558	0,150	0,071	1

4.1.4 Agrupamento ou *Clustering*

Depois de realizar a vetorização das palavras, testou-se alguns algoritmos de agrupamentos, analisou-se e verificou-se o desempenho dos mesmos em relação aos diferentes grupos formados por eles. Os algoritmos de agrupamento utilizados neste trabalho são: K-means, DBSCAN, BIRCH e Agglomerative clustering.

Para realizar o agrupamento dos dados utilizou-se a biblioteca de aprendizagem de máquina o *scikit-learn* que possui um módulo chamado *sklearn.cluster*, que reúne os mais populares algoritmos de agrupamento para dados não-supervisionados, que possuem diferentes tipos de matrizes como entrada.

Ao aplicar os algoritmos de agrupamento nos documentos da base de dados, grupos foram formados de acordo com os pressupostos de cada um dos algoritmos utilizados. Para o agrupamento inicialmente definiu-se um valor de **k**, onde **k** representa a redução máxima do texto original. No agrupamento, os grupos de características criados servem como filtro na etapa de transformação das características.

4.2 Classificação

Na segunda etapa, a de classificação, são empregadas nos grupos formados a transformação das características. Esta transformação consiste em reduzir os grupos de características criados na primeira etapa, a de treinamento, por meio da utilização de *Word Embeddings*. Após esta redução, são aplicados os algoritmos de classificação de texto, *Support Vector Machine* (SVM) e *Random Forest* (RF).

Após a formação dos novos grupos, agora já transformados em grupos menores de características, ocorreu a aplicação dos algoritmos de classificação de texto. A etapa de classificação é dividida em duas fases, a de transformação das características e a fase de aplicação dos algoritmos de classificação.

Na Figura 21, tem-se como entrada os documentos não categorizados, onde foram agrupados de acordo com as suas características e por meio da transformação das características utilizando *Word Embeddins* e os algoritmos de agrupamento, estes grupos foram transformados em grupos semânticos menores e assim foram aplicados os algoritmos de classificação de texto.

O agrupamento foi realizado com base no word2vec pré-treinado [1]. Nesta etapa o agrupamento realizado não considera apenas as palavras do treinamento e teste mais sim todas do Word2vec.

[1] Disponível em: https://code.google.com/archive/p/word2vec/

A Figura 21 apresenta uma arquitetura geral da etapa de aplicação realizada no presente trabalho.

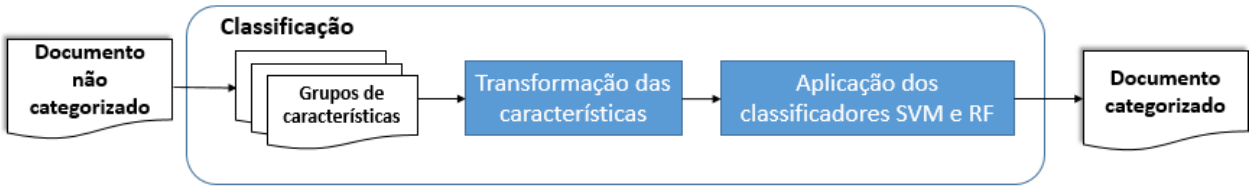


Figura 21 – Arquitetura geral do processo de aplicação.

4.2.1 Transformação das características

Como visto na Seção 2, a transformação das características consiste em diminuir a dimensionalidade das características computadas em relação a dimensionalidade geral, a transformação também procura reduzir os custos computacionais em termos de execução. Na etapa de classificação, a primeira fase a ser executada é a de transformação das características. Aqui foram processados os documentos com os grupos, ou seja, a entrada para este treinamento foram os documentos completos e obteve-se como saída apenas uma palavra que representa o grupo de características, sendo estes grupos as características finais do documento.

Como exemplo de transformação de características, temos a realização do seguinte procedimento:

Exemplo 4:

Sejam **Grupo A:** A_1, A_2, A_3, A_4 e **Grupo B:** B_1, B_2, B_3, B_4 dois grupos criados durante a etapa de treinamento, onde A_n e B_n são palavras, e o documento de exemplo D_1 contem 6 palavras $D_1 = A_1, A_2, A_3, A_4, B_2, B_3$, após o processamento o documento final, ou seja, o documento de saída que será utilizado para classificação de texto contem apenas 2 características: A, A, A, A, B, B.

Aqui todas as palavras do grupo **A** foram transformadas em um único grupo de características. O grupo **A** representa todos os termos similares (A_1, A_2, A_3, A_4).

4.2.2 Aplicação dos algoritmos de classificação nos documentos

Ainda na etapa de classificação, foram aplicados os algoritmos de classificação de textos nos documentos, estes estão detalhados na Seção 2. Foram selecionados dois algoritmos usualmente utilizados na literatura, para resolver problemas de TC, sendo eles o SVM e RF, para melhor entender a fase de classificação, observe a Figura 21.

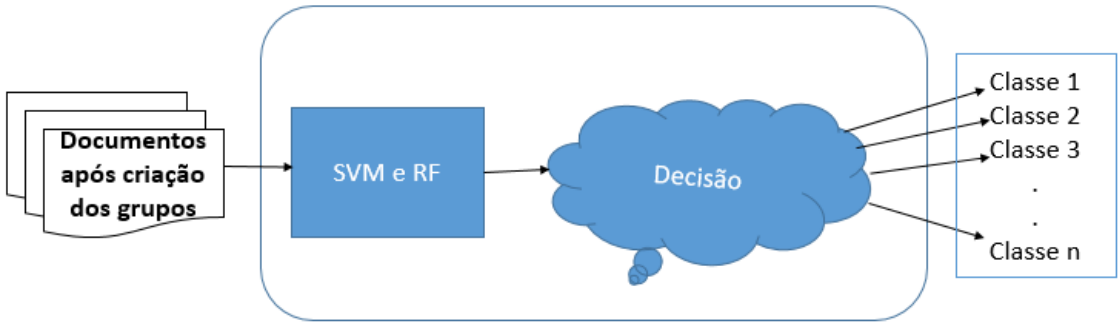


Figura 22 – Procedimento de classificação do documento.

Na etapa de classificação, tem-se como entrada um conjunto de documentos que já passaram pelo processo de criação de grupos semânticos, ou seja, já foram transformados em grupos menores de características. Estes ainda não se encontram categorizados. Logo, neste conjunto são aplicados os algoritmos de classificação de texto o SVM Seção (2.2.1.1) e RF Seção (2.2.1.2), o método de classificação nesta fase reduz o custo computacional, antes visto na fase de treinamento na Seção 4.1. Agora esse custo é reduzido pelo fato da classificação simplificar os grupos de características, como visto no **Exemplo 4** anteriormente citado, onde um documento que possui 6 termos, foi reduzido para duas características representadas por **(A e B)**.

Após esta redução, os classificadores SVM e RF destinam cada um desses documentos a uma ou mais classes como pode ser visto na Figura 22, ou seja, rotulam cada um dos documentos para um determinado tipo de categoria. Para que este procedimento seja realizado, é necessário que o documento inicialmente seja representado vetorialmente (descrição vista na Seção 4.1.3) e após a vetorização das palavras, são aplicados os classificadores que determinam a categoria que os documentos foram destinados com base nos valores dos atributos.

Como prova de conceito, foi desenvolvido um software [2] com o intuito de analisar o comportamento de cada algoritmo citado. Esse software foi desenvolvido em Python e esta disponível em: <https://github.com/ELAINEMARQUES2018/$programas_dissertacao>

[2] https://github.com/ELAINEMARQUES2018/*programas_dissertacao*

5 Experimentos

A presente seção descreve os processos realizados para execução dos experimentos. Os algoritmos de agrupamentos empregados foram o K-means, DBSCAN, BIRCH e Agglomerative Clustering. A realização dos agrupamentos foi dada por sua execução em Python. Os termos das bases de dados foram agrupados de acordo com o valor de **k**, ou seja, especificou-se a quantidade de grupos criados para cada um dos algoritmos citados de acordo com suas características, conforme apresentado no Capítulo 2, na Seção 2.3.

5.1 Configurações dos Experimentos

Os algoritmos de classificação adotados, SVM e o RF, são aplicados nos grupos semânticos criados utilizando o Word2vec e o Glove. Estes foram escolhidos por serem classificadores bastantes aplicados para resolver problemas de classificação de texto, o classificador SVM apresenta bons resultados na maioria de suas aplicações, logo, este fator foi testado para o presente experimento, outra característica importante são ajustes que pode ser realizado no Kernel, já o classificador RF foi escolhido por apresentar o melhor desempenho em um estudo onde foram analisados a eficácia de 20 classificadores de texto (FERNÁNDEZ-DELGADO et al., 2014). As vetorizações aqui mencionadas encontra-se definidas no Capítulo 2, na Seção 2.4. E assim são empregadas nos experimentos.

As análises dos classificadores foram realizadas no *software* WEKA, foram extraídas as medidas de validação dos algoritmos de classificação, Precisão, *Recall*, *F-Measure*, todas estas informações foram obtidas por meio da matriz de confusão para as diferentes bases de dados utilizadas.

No Weka, utilizou-se o método de *Percentage split* (porcentagem da divisão) na aba *Classify*, quando este método é selecionado, deve-se informar o valor relativo a fração da base de dados. Neste trabalho o valor selecionado refere-se a 70% das informações das bases de dados que foram destinadas ao treinamento e nos 30% restantes foram realizados testes do algoritmo. Estes procedimentos servem como validação do modelo proposto (BOUCKAERT et al., 2008). Os algoritmos utilizados foram o SVM e RF, cujas configurações de execução dos classificadores são apresentadas pelo seguinte esquema utilizado pelo *software* Weka:

- weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007-calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"

- weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Para validação dos resultados, utilizou-se a técnica estatística conhecida como Anova de medidas repetidas. Esta técnica analisa a situação em que, para cada uma das bases de dados apresentadas estas são medidas por condições experimentais ou momentos diferentes. Na Anova analisou-se a Esfericidade (ε), pelo teste de Mauchly's, onde valores de $p \leq 0,05$ indicam a violação da hipótese, quando este pressuposto é violado alguns testes devem ser aplicados para realizar a correção da hipótese de Esfericidade, sendo eles o de Greenhouse-Geiser e a de Huynh-Feldt, a escolha do teste depende do *p-valor*.

- Quando p-valor $p \leq 0,05$ é dito que o pressuposto de esfericidade é violado. Logo, o teste de Greenhouse-Geiser deverá ser utilizado se o valor de Epsilon (ε) for $\varepsilon \leq 0,750$.
- Quando o pressuposto de esfericidade é violado e o valor de Epsilon (ε) for $\varepsilon > 0,750$ deve-se aplicar o teste de Huynh-Feldt.

Realizou-se os experimentos em uma máquina dedicada exclusivamente para obtenção dos resultados sugeridos com processador Intel Quadre Core 2 de 2.5 Giga Hertz, memória RAM de 4 Gigabytes e sistema operacional Windows 10. As seções que seguem detalham as bases de dados utilizadas.

5.1.1 Bases de Dados (*Datasets*)

As bases de dados são compostas por uma coleção de documentos previamente classificados, que servem como instrumento de avaliação para a classificação de textos. Para o presente trabalho utilizou-se 2 (duas) bases de dados bastante mencionados na literatura de TC (FORMAN, 2003; FENG et al., 2015; PINHEIRO; CAVALCANTI; REN, 2015; TANG; KAY; HE, 2016; UĞUZ, 2011; YANG et al., 2012), sendo as bases *Reuters* 21578 e alguns subconjuntos da base WebKB, cada uma destas bases possuem tamanho específico, tanto no número de termos dos documentos quanto em seu vocabulário, elas também se diferenciam em relação a proporção de documentos por categoria e em conteúdo.

Quantidades de categorias, documentos e termos foram adotadas de acordo com a metodologia utilizada nos trabalhos relacionados, alguns trabalhos utilizaram seus dados realizando a análise a partir do método *Cross Validation* e outros empregaram em seus dados a utilização do método *Percentage split*. Logo, foi possível realizar a comparação dos resultados presentes na literatura com a presente dissertação. A Tabela 5 apresenta um resumo das informações de cada uma das bases utilizadas e em seguida é apresentada uma descrição de suas características.

Tabela 5 – Descrição das bases de dados.

Base	Nº de categorias	N º de documentos	Nº de termos
WebKB	4	4.199	7.770
Reuters 10	10	9.980	10.987

- **WebKB**

  As informações contidas nesta base fazem referência a documentos de 4 (quatro) faculdades norte americanas obtidas no ano de 1997. Esta é composta por 8.282 documentos e possuem 7 (sete) categorias. No presente trabalho utilizou-se um subconjunto destas informações, utilizando-se apenas 4.199 documentos com um vocabulário de 7.770 termos e apenas 4 categorias sendo elas: (*course, faculty, project, student*), a maior categoria apresenta aproximadamente 39% dos documentos em relação ao tamanho total da base, e a menor categoria possui 12% dos documentos.

- **Reuters 10**

  A partir da coleção *Reuters*-21578 extraiu-se um subconjunto de dados *Reuters* 10 sendo esta uma das bases mais utilizadas para trabalhos na área de classificação de texto (TC) (DEBOLE; SEBASTIANI, 2005). A base é caracterizada por documentos que foram coletados na coleção *Reuters newswire* de 1987 e possui 135 categorias. Diante a grandeza do número de categorias da base de dados, optou-se por utilizar no presente trabalho um subconjunto das informações, apenas as 10 maiores categorias existentes na base, por isso a denominação *Reuters* 10, que possui 9.980 documentos e 10.987 termos em seu vocabulário. Abordagens semelhantes a esta é observada em (CHANG; CHEN; LIAU, 2008) (CHEN et al., 2009). A base de dados *Reuters* 10 apresenta um desbalanceamento em relação a distribuição de seus documentos, algumas classes são representadas desde $2,3\%$ até 39% dos documentos em relação ao tamanho total da base.

Todos os documentos das duas bases de dados passaram pelo processo de pré-processamento, onde foram removidos: pontuação, caracteres irrelevantes, números, remoção de termos com no máximo duas letras, remoção de *stopwords* (SALTON; MCGILL, 1971) e *tokenização*.

5.1.2 Critérios de Avaliação

As medidas são calculadas a partir da equação:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5.1}$$

P é a medida de precisão (do inglês *precision*), R é a cobertura (do inglês *recall*). A precisão é calculada pela fórmula 5.2 e a cobertura por 5.3:

$$P_{(c_j)} = \frac{TP_j}{TP_j + FP_j} \tag{5.2}$$

$$R_{(c_j)} = \frac{TP_j}{TP_j + FN_j} \tag{5.3}$$

TP_j é a quantidade de instâncias classificadas corretamente, ou seja, são as instâncias que foram categorizadas como pertencentes à categoria c_j. Falso Positivo (FP_j), são as instâncias classificadas incorretamente, ou seja, são as instâncias classificadas incorretamente como pertencentes à categoria c_j e Falso Negativo (FN_j), representa a quantidade de instância classificadas incorretamente como não pertencentes a classe c_j.

As medidas de avaliação do classificador, Macro - F1 e Micro - F1, também utilizadas neste trabalho, possuem diferenças na forma em que são calculadas. Aqui a diferença ocorre no cálculo das médias da precisão e da cobertura.

A Micro - F1, apresenta em seu cálculo, um peso correspondente a categoria com base em seu tamanho, este peso contribui no desempenho de algumas categorias, favorecendo aquelas que possuem mais documentos em seu resultado final. O cálculo da Macro - F1 é dado pela média aritmética, ou seja, todas as classes recebem o mesmo peso, favorecendo assim o desempenho das classes que possuem poucos documentos.

Para calcular as medidas Macro - F1 e Micro - F1, novos cálculos das medidas de precisão e cobertura são realizados e apresentados a seguir.

$$P_\mu = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q}(TP_j + FP_j)} \tag{5.4}$$

$$R_\mu = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q}(TP_j + FN_j)} \tag{5.5}$$

Os cálculos da precisão e cobertura para a Micro - F1, pode ser visto nas Equações 5.4 e 5.5 e para a Macro - F1 os cálculos da precisão é realizado por meio da Equação 5.6 e a cobertura pela Equação 5.7.

$$P_M = \frac{\sum_{j=1}^{Q} P_{(c_j)}}{Q} \tag{5.6}$$

$$R_M = \frac{\sum_{j=1}^{Q} R_{(c_j)}}{Q} \tag{5.7}$$

5.2 Resultados dos Experimentos

Nesta seção serão expostos os resultados dos experimentos realizados a partir dos métodos propostos neste trabalho. Os resultados fazem referência a cada um dos objetivos específicos descritos na presente dissertação. Os gráficos apresentados facilitam a interpretação dos resultados obtidos por meio das medidas de análise de *precisão*, *recall*, *f-measure*, para cada umas das bases de dados do estudo, além da parte gráfica, testes estatísticos foram realizados afim de comprovar a eficácia das medidas de avaliação dos grupos semânticos e dos classificadores de texto.

Os resultados apresentarão comparações entre os classificadores de texto, *Word Embeddings* e os algoritmos de agrupamento. As análises estatísticas apresentarão o melhor grupo de características para cada um dos *embeddings*, como também o melhor algoritmo de classificação de texto que apresenta os melhores grupos de características. A análise das bases de dados deu-se através da comparação dos critérios de avaliação das medidas Micro-F1 e Macro-F1, onde estas verificam o nível de adequação das medidas de decisão.

5.2.1 Resultados obtidos da base de dados Reuters

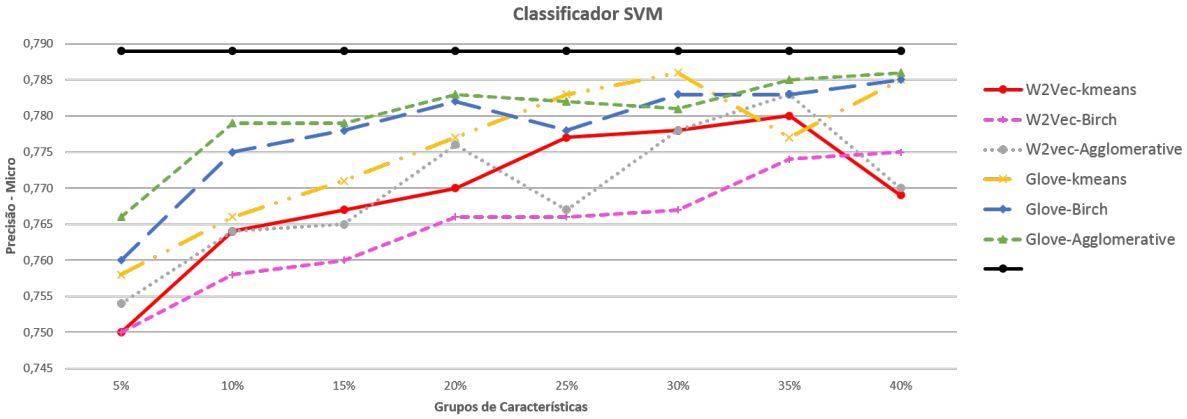
Os primeiros resultados são referentes a base de dados Reuters, onde compara-se a precisão por meio das medidas de avaliação do classificador Micro - F1 e Macro - F1, respectivamente, afim de verificar o desempenho dos grupos semânticos juntamente com a forma em que ocorreu a vetorização das palavras, quando utiliza-se os classificadores SVM e RF.

Os grupos de características de 5% a 40%, variando os grupos a cada 5%, determinam o número de características que devem ser selecionadas por documento. Os melhores resultados dos grupos de características encontram-se quando os grupos assumem valores de 30%. Verificou-se que para os grupos de características maiores que 30% não são observadas melhorias nos grupos. Apenas valores de grupos de características de 5%, 10%, 15%, 20%, 25%, 30%, 35% e 40% foram utilizados nos experimentos, para ambos os métodos.

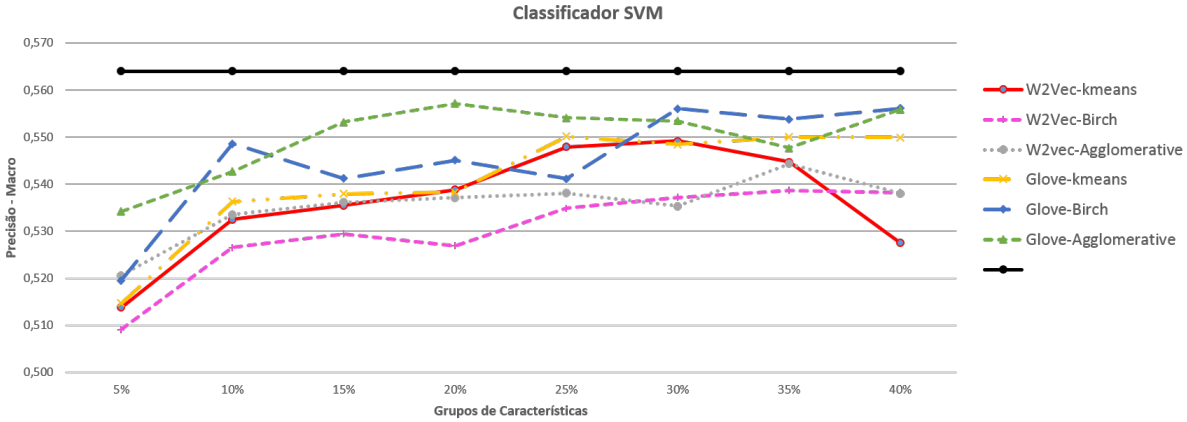
A Figura 23, exibe os resultados da medida de precisão utilizando o algoritmo de classificação SVM. No gráfico **(a)**, os melhores resultados da precisão encontra-se nos grupos de características entre 30% e 40%. Considerando o grupo de 30% o Glove mostrou-se com melhor resultado de vetorização de palavras juntamente com o algoritmo de agrupamento K-means, que apresenta-se em destaque nos grupos semânticos de características criados.

Para o gráfico **(b)**, a melhor precisão encontra-se entre o mesmo intervalo de grupos de características do anterior, o Glove apresenta-se como o *embedding* em destaque, e o algoritmo de agrupamento com maior valor de precisão é o Birch, também para o grupo de características de 30%.

A linha de cor preta representa o desempenho geral da precisão, ou seja, a média da precisão para as medidas de Macro - F1 e Micro - F1, de 100% das informações dos documentos. Em classificação de texto, a medida de precisão próxima a 1, significa que as palavras capturadas são relevantes para o documento. Logo, a medida de precisão mais relevante para o estudo encontra-se no gráfico **(a)** de avaliação Micro - F1 e com valor de precisão = 0,789.



(a) Precisão com avaliação Micro-F1



(b) Precisão com avaliação Macro-F1

Figura 23 – Exemplos da medida de avaliação precisão com o algoritmo SVM

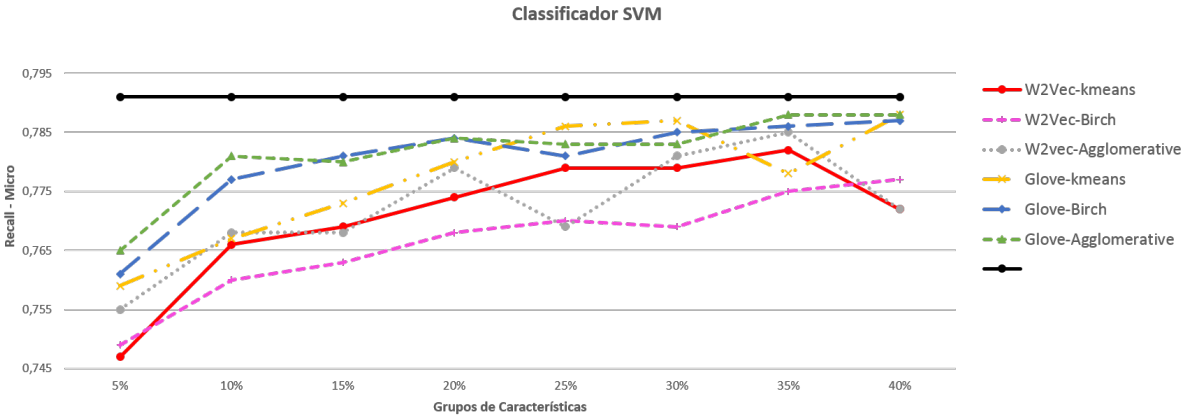
A Figura 24, apresenta graficamente as informações para a medida de avaliação da medida *recall*, para o gráfico **(a)** e **(b)**. Ambos os gráficos apresentam como um resultado bem interessante para a análise a vetorização de palavras realizada pelo Glove. Este *embedding* destaca-se para os maiores grupos de características.

Os algoritmos de agrupamento K-means e *Agglomerative Clustrering* apresenta-se como melhores grupos de características com (30%) e (35%), respectivamente, para a medida de avaliação do classificador Micro - F1, como mostra o gráfico **(a)**.

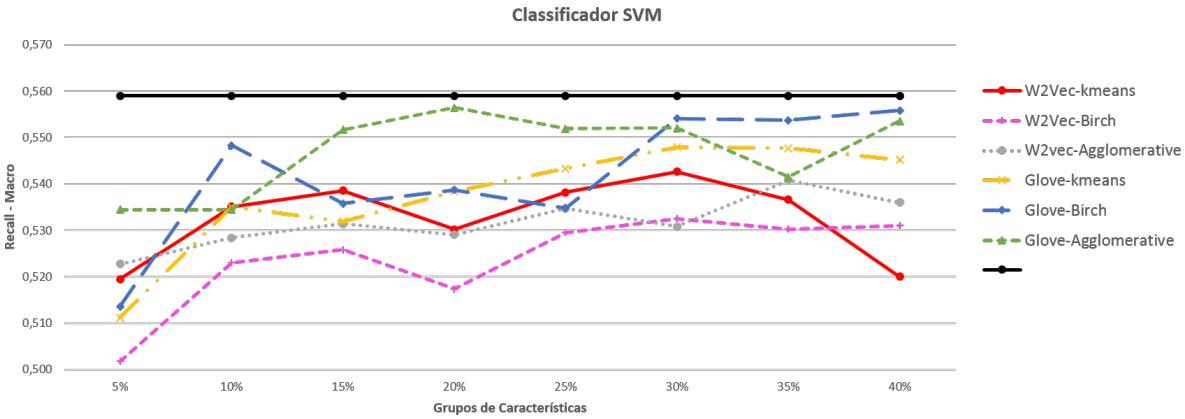
Quando analisa-se o gráfico **(b)**, é possível observar o desempenho da medida de *recall* para a medida de avaliação Macro - F1. Logo, temos como características dos

grupos semânticos os algoritmos Birch e o *Agglomerative Clustrering* que apresenta as características dos documentos com grupos de (30%) e (35%), respectivamente.

A linha de cor preta representa o desempenho geral da cobertura, ou seja, a média da cobertura de todos os documentos, no gráfico **(a)** a acurácia indica a relevância das palavras na recuperação, considerando o conjunto total de palavras relevantes com um valor 0,791.



(a) Recall com avaliação Micro-F1



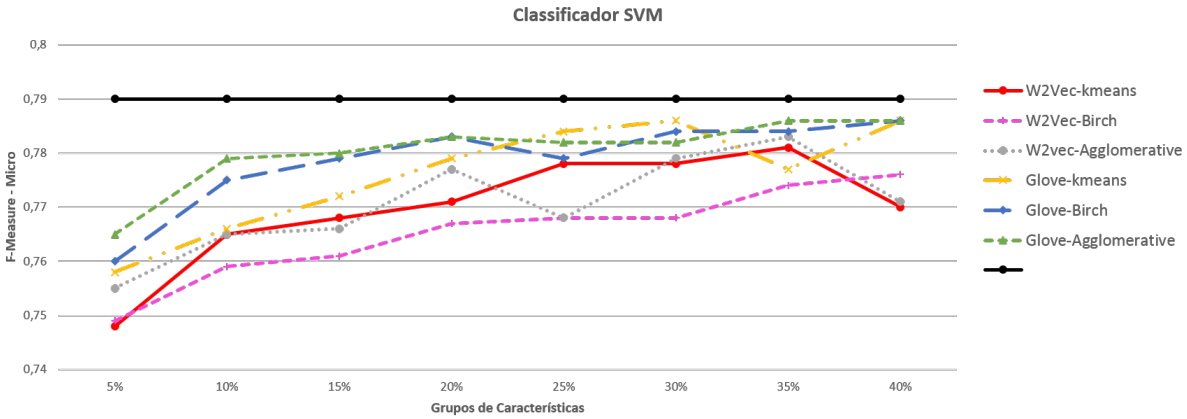
(b) Recall com avaliação Macro-F1

Figura 24 – Exemplos da medida de avaliação recall com o algoritmo SVM

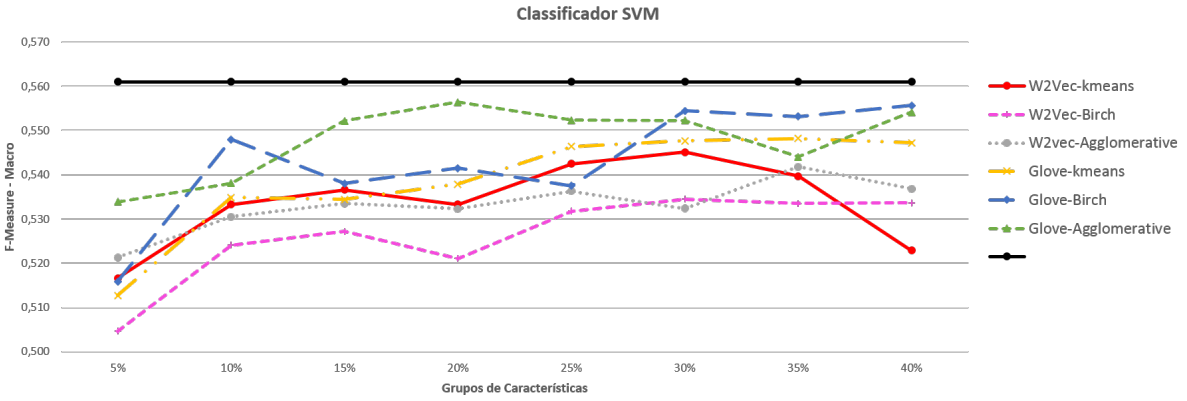
Para a medida *F-measure*, medida harmônica entre a precisão e o *recall* a Figura 25, também apresenta o Glove como a melhor vetorização de palavras dos documentos juntamente com a medida de avaliação do classificador Micro - F1. Os grupos formados pelos algoritmos K-means e *Agglomerative Clustering* apresenta-se em destaque para os grupos de (30%) e (35%), respectivamente, das características em relação ao conjunto de características original, isto pode ser observado no gráfico **(a)**.

No gráfico **(b)**, para a medida de avaliação Macro - F1, os algoritmos de agrupamentos em destaque foram o Birch e o *Agglomerative Clustering* com o grupo de característica de 30% cada.

A linha de cor preta representa mais uma vez o desempenho geral da medida em estudo. Neste caso, a média geral de *f-measure*, apresenta um valor médio geral de 0,790.



(a) F-Measure com avaliação Micro-F1



(b) F-Measure com avaliação Macro-F1

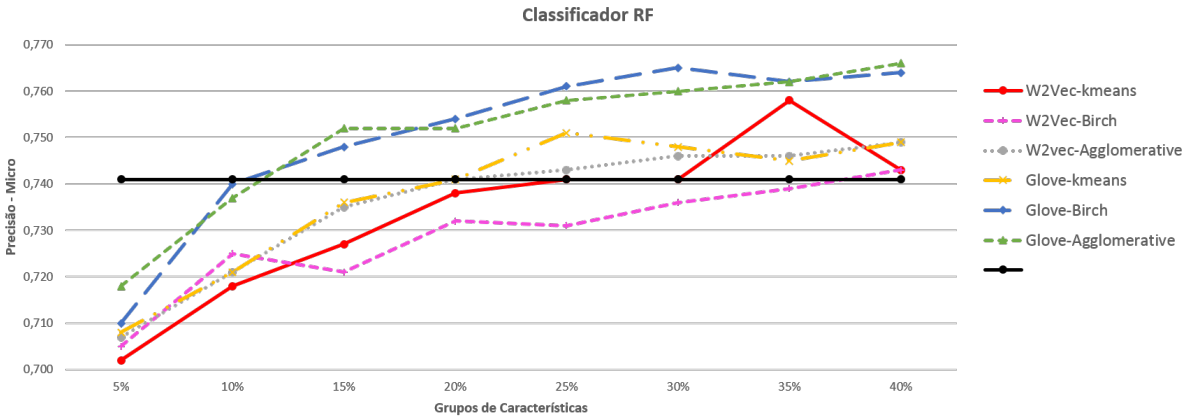
Figura 25 – Exemplos da medida de avaliação f-measure com o algoritmo SVM

Logo, conclui-se que, o *Word Embedding* que apresentou resultados superiores para todas as medidas de avaliação foi o Glove, já os algoritmos de agrupamentos apresentam resultados mais aconselháveis quando formados por grupos de características maior ou igual a 30%. Os grupos de características que aparecem na maioria das análises são os grupos que assumem valores de 30% e 35%.

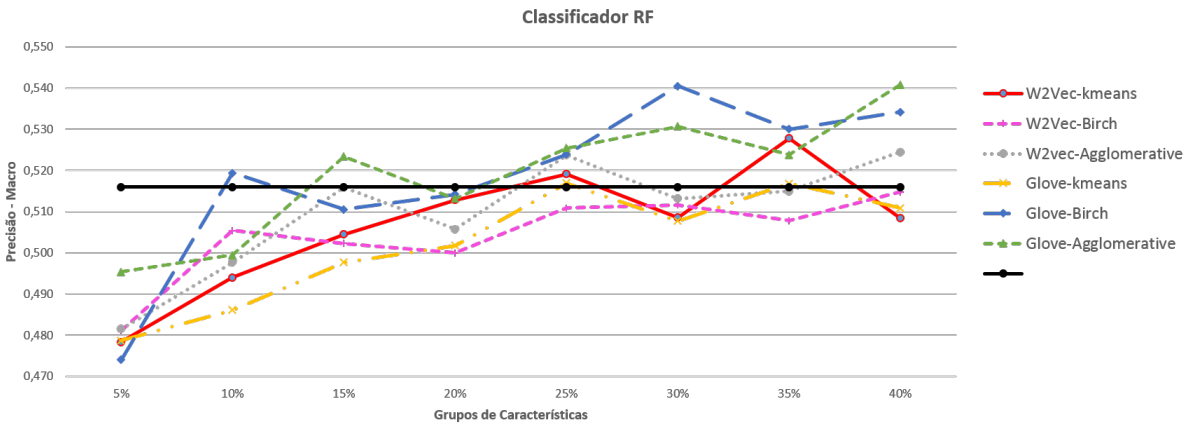
Nos gráficos a seguir os resultados fazem referência ao algoritmo de classificação *Random Forest*, os resultados apresentam informações da mesma base de dados, a Reuters, e as medidas analisadas são as mesmas medidas de avaliação mencionas na análise anterior.

A Figura 26, apresenta o resultado da medida de precisão, desta vez, utilizando o classificador *Random Forest* para as medidas de avaliação dos classificadores, sendo elas as medidas Micro - F1 **(a)** e Macro - F1 **(b)**. No gráfico que analisa a precisão, o Glove destaca-se como melhor *embedding*, pois apresenta como melhor grupo de característica o grupo que contém 30% das informações. Para este algoritmo de classificação a média de Micro - F1 da precisão e Macro - F1 encontram-se abaixo das médias dos grupos de características, representado pela linha de cor preta. As médias obtidas por meio dos grupos semânticos, só comprovam a eficácia do método proposto.

Onde a criação destes grupos semânticos, mesmo com uma quantidade menor de características em relação aos documentos originais, apresentam resultados precisos para a análise, ou seja, é possível explicar determinados assuntos com apenas 30% das informações obtidas nas bases.



(a) Precisão com avaliação Micro-F1



(b) Precisão com avaliação Macro-F1

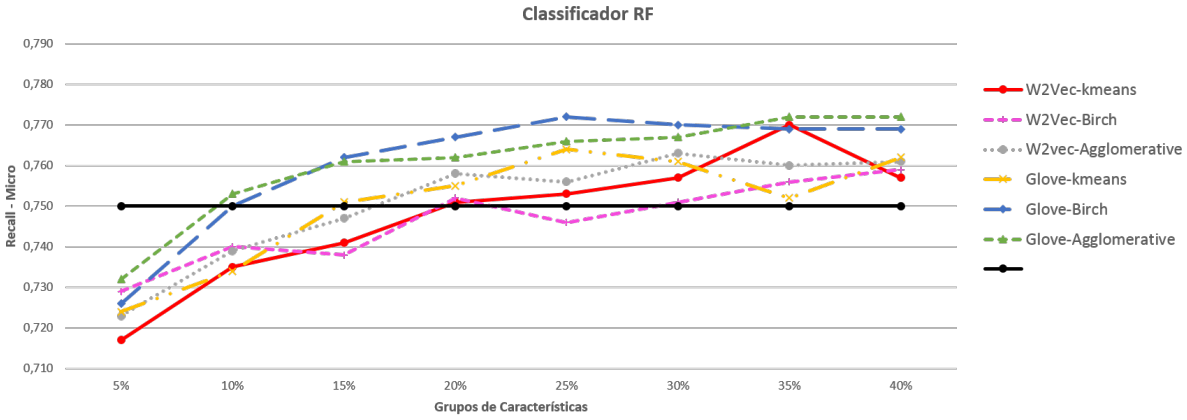
Figura 26 – Exemplos da medida de avaliação precisão com o algoritmo RF

A Figura 27, faz analise da medida *recall*, ou seja, a cobertura fornece informações da relevância das palavras recuperadas com êxito, considerando seu conjunto total.

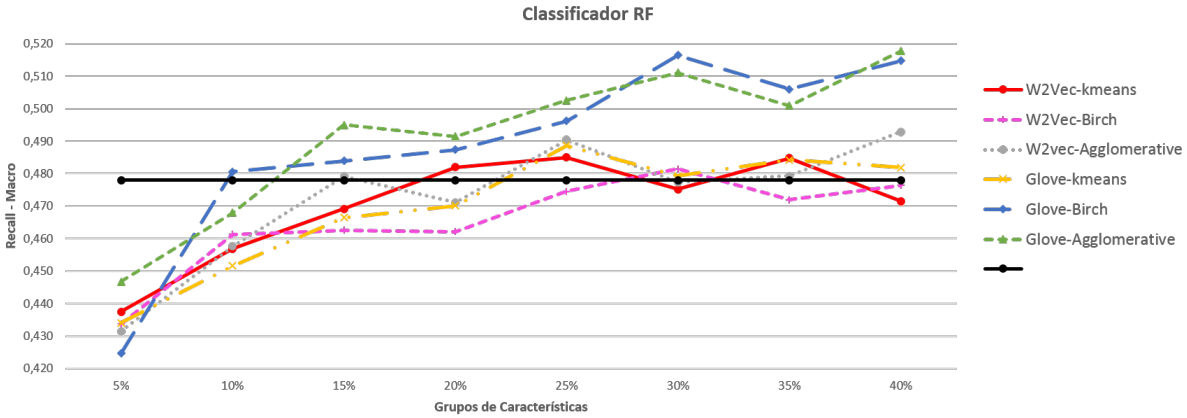
O gráfico **(a)** apresenta o Glove como a melhor forma de vetorizar as palavras por meio do agrupamento realizado pelo algoritmo Birch. Pode-se perceber também que as medidas de avaliação do classificador, a Micro - F1, mostra resultados superiores em relação a medida Macro - F1. Este fator pode ser visto a partir da linha de cor preta, que representa a média geral da medida de cobertura da base de dados.

A média geral observada no gráfico **(a)** é **0,750** e a média do gráfico **(b)** que representa a Macro - F1 é de **0,478**.

Pode se observar também que os grupos de características formados pelo Glove-Agglomerative apresenta resultados próximos ou superiores ao agrupamento Glove-Birch em alguns dos pontos analisados.



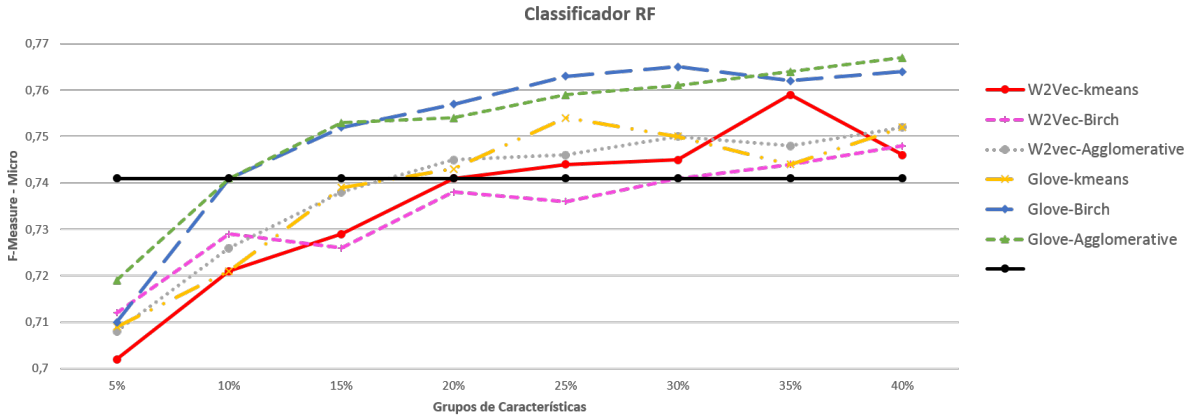
(a) Recall com avaliação Micro-F1



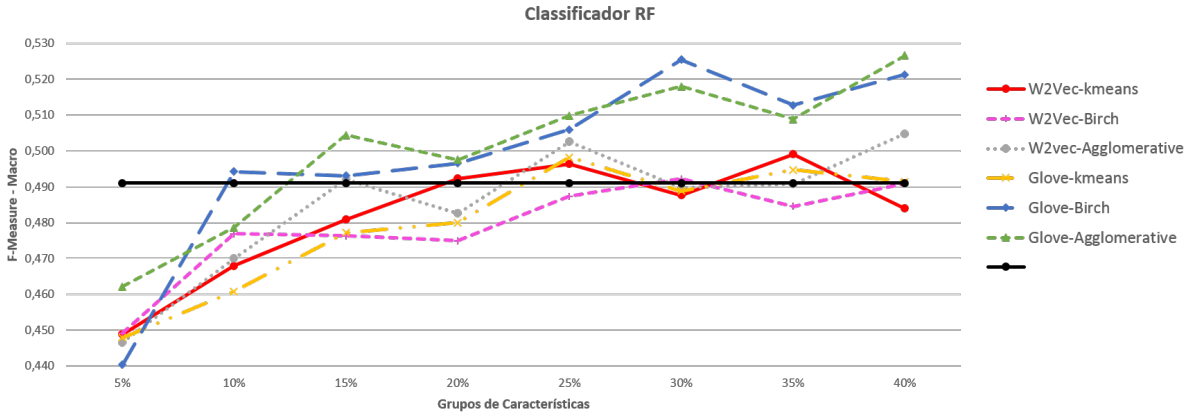
(b) Recall com avaliação Macro-F1

Figura 27 – Exemplos da medida de avaliação recall com o algoritmo RF

A Figura 28, apresenta os resultados obtidos pela medida *f-measure*, o grupo em destaque continua sendo o grupo com 30% das características e os algoritmos de agrupamento que aparecem com este percentual são o Birch e *Agglomerative Clustering*, as médias dessa medida de avaliação, mostram-se superiores à média geral dos documentos, representado pela linha de cor preta.



(a) F-Measure com avaliação Micro-F1



(b) F-Measure com avaliação Macro-F1

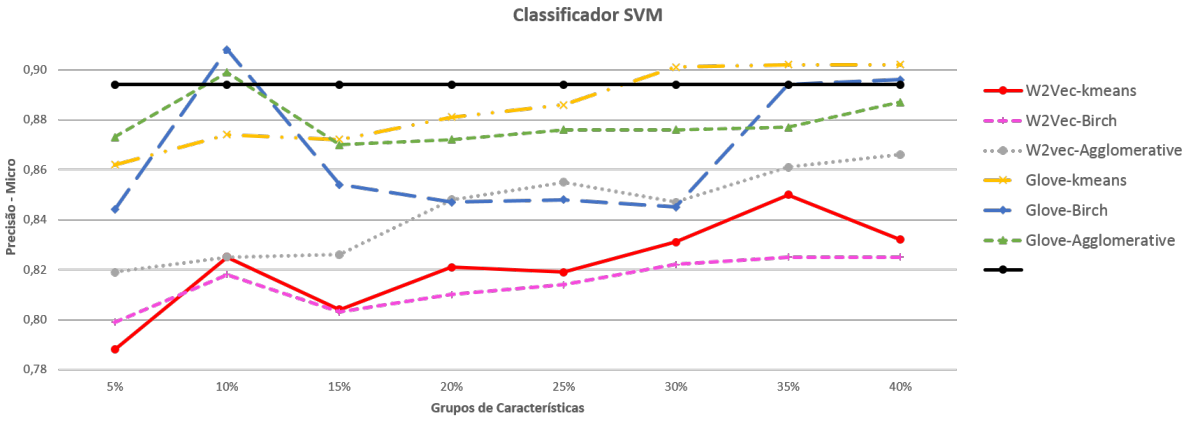
Figura 28 – Exemplos da medida de avaliação F-Measure com o algoritmo RF

5.2.2 Resultados obtidos da base de dados WebKB

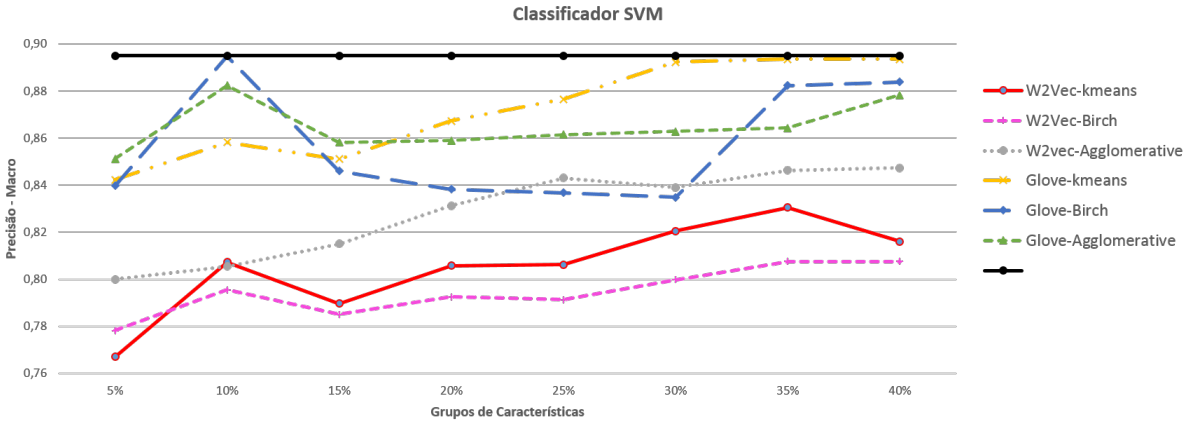
Para os resultados adquiridos na base WebKB utilizando as medidas de desempenho de Precisão, *Recall* e *F-Measure* dos classificadores de texto SVM e RF por meio das métricas de avaliação Micro - F1 e Macro - F1, foi possível observar que, entre os grupos de características de maiores destaques estão os grupos entre 30% e 40%. Contudo, é importante perceber que os melhores resultados foram alcançados com 10% das características. Neste banco, os vetores de palavras são melhores representados vetorialmente pela técnica Glove, ou seja, a representação vetorial de Word Embeddings do Glove apresenta resultados superiores ao Word2Vec.

Vale destacar que para o algoritmo de classificação de texto *Random Forest* o Glove - Birch e Glove - Kmeans, apresentam os melhores resultados, sendo o primeiro, na maioria dos experimentos a pontuar os maiores valores para o grupo de características de 35%.

Diferente dos resultados no banco *Reuters*, aqui todos os valores das diferentes medidas de avaliação convergiram para os mesmos intervalos, por isso foi apresentado a análise em conjunto dos resultados e não gráfico a gráfico, como na seção anterior.

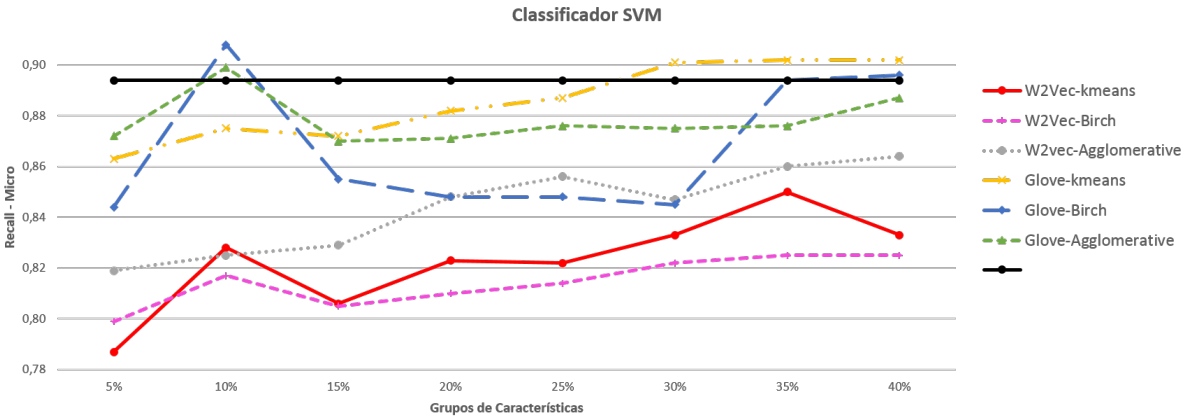


(a) Precisão com avaliação Micro-F1

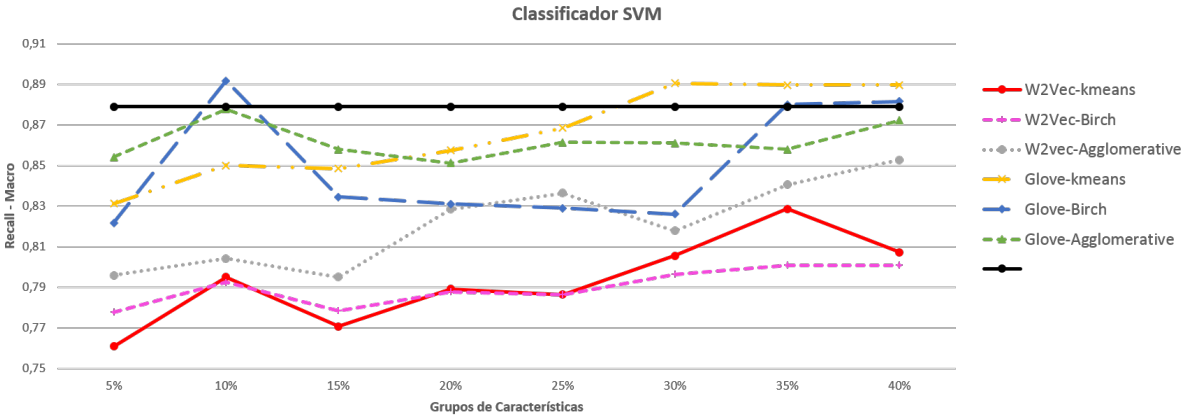


(b) Precisão com avaliação Macro-F1

Figura 29 – Exemplos da medida de avaliação precisão com o algoritmo SVM

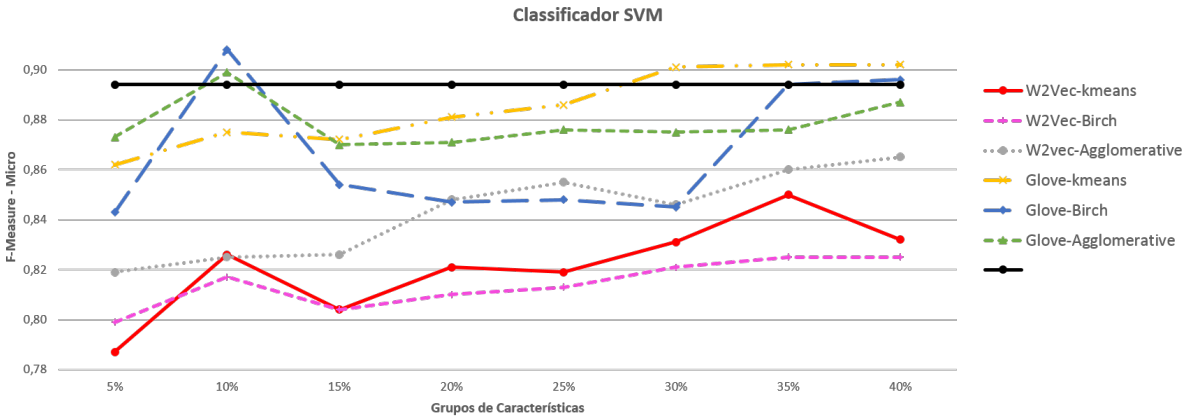


(a) Recall com avaliação Micro-F1

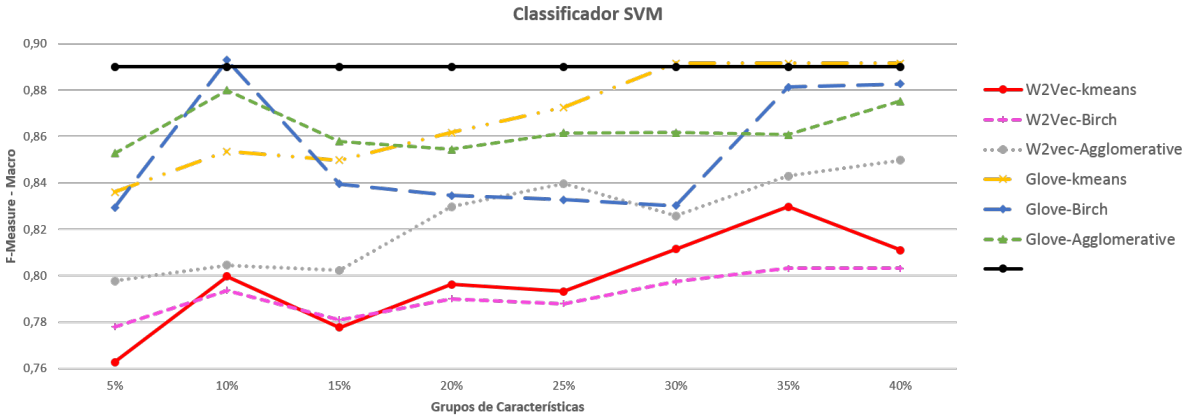


(b) Recall com avaliação Macro-F1

Figura 30 – Exemplos da medida de avaliação recall com o algoritmo SVM

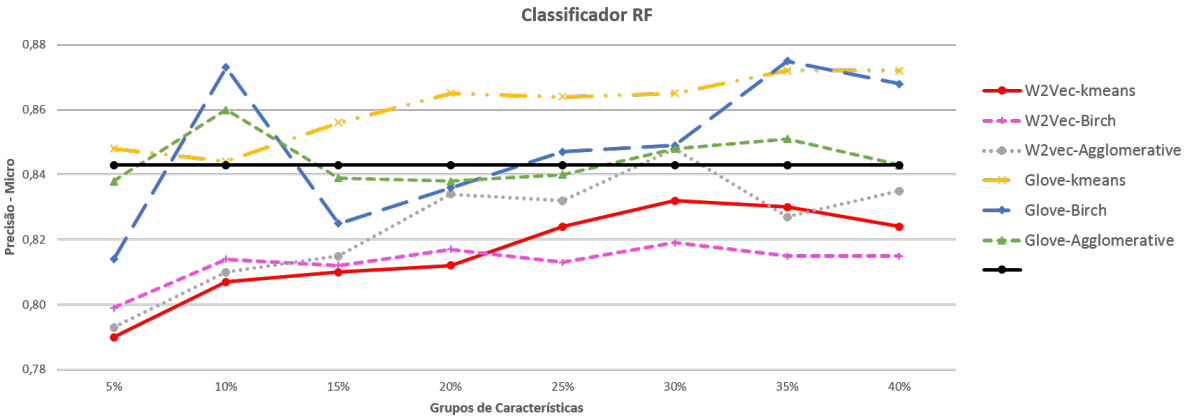


(a) F-Measure com avaliação Micro-F1

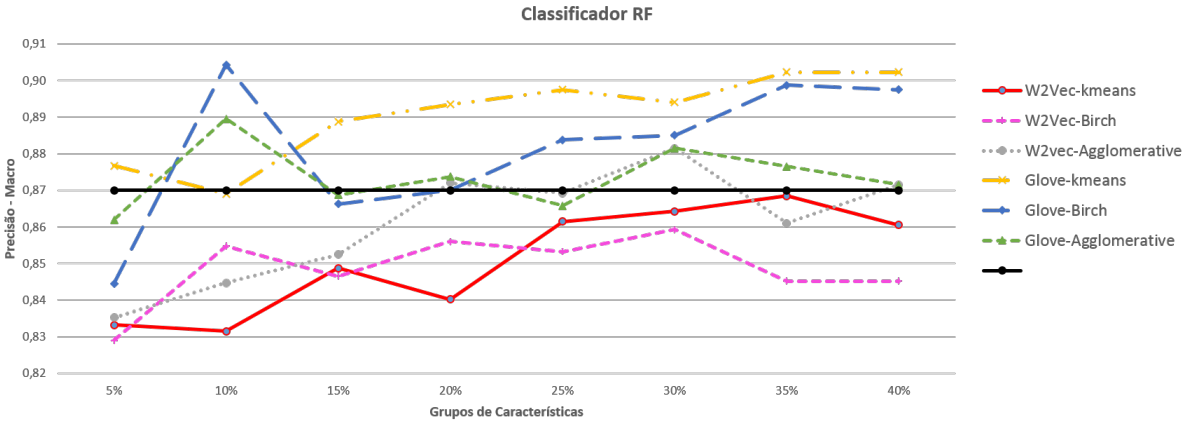


(b) F-Measure com avaliação Macro-F1

Figura 31 – Exemplos da medida de avaliação F-Measure com o algoritmo SVM

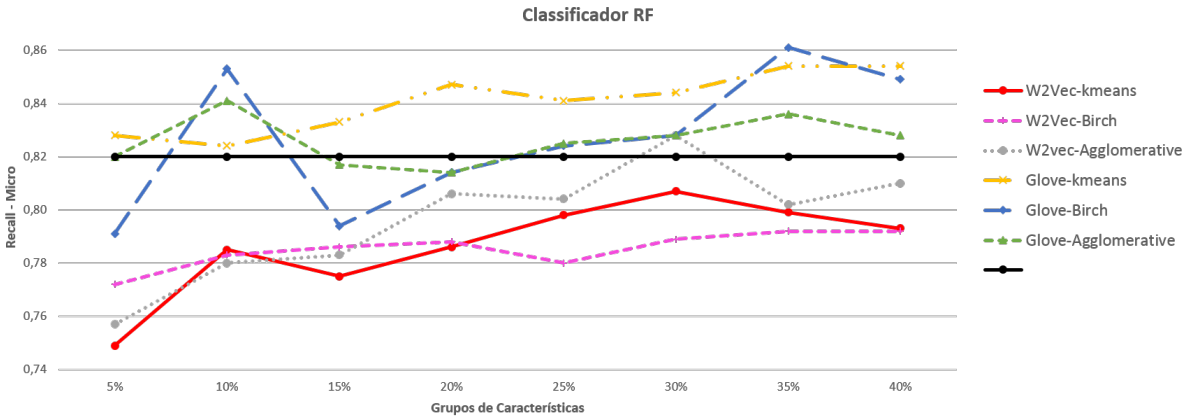


(a) Precisão com avaliação Micro-F1

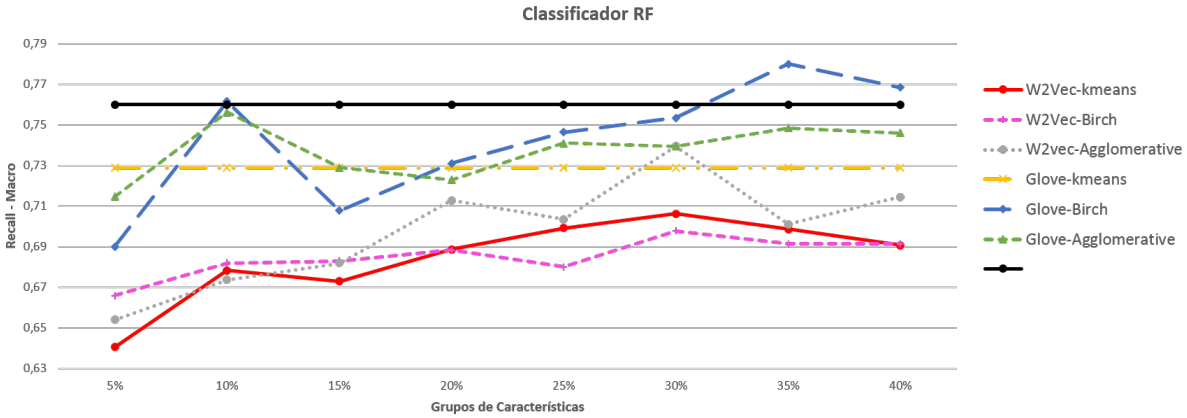


(b) Precisão com avaliação Macro-F1

Figura 32 – Exemplos da medida de avaliação Precisão com o algoritmo RF

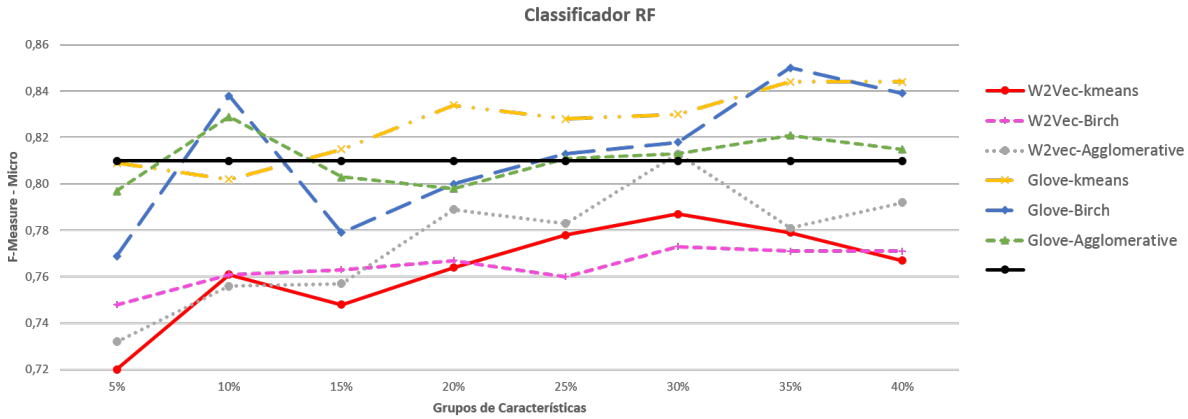


(a) Recall com avaliação Micro-F1

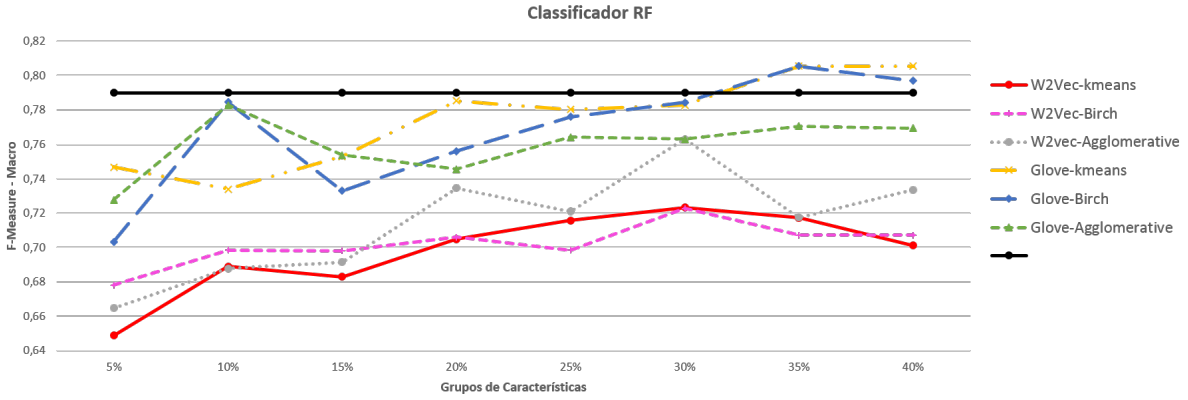


(b) Recall com avaliação Macro-F1

Figura 33 – Exemplos da medida de avaliação Recall com o algoritmo RF



(a) F-Measure com avaliação Micro-F1



(b) F-Measure com avaliação Macro-F1

Figura 34 – Exemplos da medida de avaliação F-Measure com o algoritmo RF

5.2.3 Anova de Medidas Repetidas

Neste tipo de análise estatística, existe a vantagem de reduzir a variabilidade dos dados. Logo, alguns pressupostos devem ser satisfeitos para que se tenha resultados significativos diante a análise estatística. Aqui a distribuição de normalidade dos dados é uma suposição a ser considerado, principalmente dos resíduos, mas o pressuposto de maior importância e mais relevante para a Anova de Medidas Repetidas com dois fatores é o de Esfericidade (ϵ).

O pressuposto de Esfericidade esta relacionado a igualdade das variâncias das diferenças entre os pares dos diferentes níveis de condições experimentais. Para que ocorra esta igualdade é necessário que exista no mínimo três condições experimentais. As condições experimentais deste modelo são, a relação dos grupos semânticos com os grupos de características, os grupos semânticos em relação aos *Word Embeddings* e os grupos semânticos em relação aos algoritmos de classificação.

Para que os testes de comparações do pares de grupos sejam realizadas é necessário primeiramente analisar se os grupos semânticos apresentam igualdade nas variâncias das diferenças entre os pares em estudo. Se o teste de Mauchhly's apresenta p-valor maior que 0,05 isto significa, que a hipótese de Esfericidade é aceita. O teste de Mauchly's, é o teste estatístico que determina se a hipóteses estatística definida é violada ou não.

As hipóteses foram as seguintes:

H_0 = Há homogeneidade na diferença das variâncias entre todos os pares de medidas.

x

H_1 = Não há homogeneidade na diferença das variâncias entre todos os pares de medidas.

De acordo com a Tabela 6, e observando o p-valor em **Sig.**, é possível identificar que o teste de Mauchly's não viola a hipótese nula definida, ou seja, não rejeita H_0, o que traz informações sobre a homogeneidade na diferença das variâncias entre todos os pares de medidas. Esta homogeneidade ocorre pelo fato da base de dados Reuters possuir um número maior de categorias em relação a WebKB.

Na Tabela 7, apenas a medida de validação do modelo *Recall* Micro viola a hipótese de Esfericidade, ou seja, rejeita H_0. Quando o pressuposto de Esfericidade é violado o teste de Greenhouse - Geiser deverá ser utilizado se o valor de Epsilon (ϵ) for $\leq$ 0,750, para que ocorra a correção da hipótese de Esfericidade. Estes conceitos estão definidos na Seção 5.1.

Tabela 6 – Teste de Esfericidade - Reuters

Within Subj. Effect	**Mauchly's W**	**Approx. Chi-Square**	**df**	**Sig.**	**Esfericidade**		
					Greenhouse Geisser	Huynh Feldt	Lim. Inferior
Precisão Micro	0,186	8,571	14	**0,878**	0,688	1,000	0,200
Precisão Macro	0,314	5,915	14	**0,974**	0,666	1,000	0,200
Recall Micro	0,060	15,603	14	**0,391**	0,501	1,000	0,200
Recall Macro	0,228	7,546	14	**0,925**	0,603	1,000	0,200
F-measure Micro	0,108	11,343	14	**0,700**	0,597	1,000	0,200
F-measure Macro	0,228	7,546	14	**0,925**	0,603	1,000	0,200

Tabela 7 – Teste de Esfericidade - WebKb

Within Subj. Effect	**Mauchly's W**	**Approx. Chi-Square**	**df**	**Sig.**	**Esfericidade**		
					Greenhouse Geisser	Huynh Feldt	Lim. Inferior
Precisão Micro	0,014	21,734	14	**0,114**	0,547	1,000	0,200
Precisão Macro	0,036	16,900	14	**0,312**	0,506	1,000	0,200
Recall Micro	0,000	54,560	14	0,000	**0,227**	0,550	0,200
Recall Macro	0,136	10,192	14	**0,781**	0,547	1,000	0,200
F-measure Micro	0,004	25,578	14	0,026	**0,499**	1,000	0,200
F-measure Macro	0,228	7,546	14	**0,925**	0,603	1,000	0,200

Após a verificação do pressuposto de Esfericidade estão propícios os demais testes de comparações entre os pares de grupos do experimento em investigação.

5.2.4 Testes Estatísticos para comparação dos algoritmos de classificação

Esta seção descreve as configurações para execução dos testes estatísticos em relação ao experimento de comparação entre os classificadores de texto SVM e RF mencionados na Seção 2.2.1.1 e 2.2.1.2, respectivamente.

A análise de variância de medidas repetidas com dois fatores (do inglês, Anova - *two-way*) visto na Seção (5.1), foi realizada afim de verificar as condições de avaliação dos resultados dos experimentos, onde estes, são avaliados na mesma unidade experimental ou no mesmo indivíduo. Os resultados obtidos são originados por meio de uma classificação que acontece dentro das subpopulações ou tratamentos. Por esta razão essas foram as análises estatísticas empregadas nos experimentos.

As estimativas dos classificadores apresentadas nas Tabelas 8 e 9, respectivamente, fornecem informações de suas médias e seus respectivos intervalos de confiança de 95%, para as duas condições avaliadas, ou seja, os classificadores SVM e RF. Estas informações

permitem analisar e identificar em quais momentos ocorrem as diferenças significativas apontadas no teste multivariado.

Tabela 8 – Estimativas - F-Measure Micro - F1 - Reuters

Classificador	**Média**	**Std. Error**	**Intervalo de Confiança (95%)**	
			Limite Inf.	Limite Sup.
SVM	0,773	0,002	0,769	0,778
RF	0,742	0,002	0,738	0,746

Tabela 9 – Estimativas - F-Measure Micro - F1 - WebKB

Classificador	**Média**	**Std. Error**	**Intervalo de Confiança (95%)**	
			Limite Inf.	Limite Sup.
SVM	0,851	0,002	0,847	0,856
RF	0,792	0,002	0,788	0,796

Inicialmente, por meio dos resultados, surgiu a necessidade de saber se existe diferença entre os algoritmos de classificação SVM e RF, logo, construiu-se uma hipótese estatística afim de facilitar a decisão a ser tomada, duas hipóteses foram definidas:

$$H_0 = \text{Os algoritmos de classificação são iguais}$$
$$\text{x}$$
$$H_1 = \text{Os algoritmos de classificação são diferentes}$$

Pelo intervalo de confiança (IC) é possível observar que existe uma diferença significativa entre os classificadores SVM e *Random Forest*, ou seja, com 95% de confiança, os resultados indicam que os classificadores não são iguais, como pode ser observado na Tabela 8, para a base de dados Reuters e Tabela 9, para a base de dados WebKB. Aqui os classificadores SVM e *Random Forest* apresentam diferenças em todas as medidas de validação do modelo, ou seja, para as medidas de precisão, *recall* e *f-measure*, como para as medidas de avaliação dos classificadores a Micro - F1 e Macro - F1. Diante os resultados é possível observar que os intervalos relevantes encontram-se para o classificador SVM, quando este é avaliado pela Micro - F1.

Nas Tabelas 8 e 9, são apresentados apenas os intervalos da medida *F-measure*, visto que esta medida é uma medida harmônica entre a precisão e a cobertura, definiu-se que apenas ela seria necessária para explicar a diferença entre os estimadores para as outras medidas de validação, já que o intervalo entre a precisão e cobertura estão próximas a *f-measure*.

Em resumo, o classificador SVM apresenta-se em destaque para os experimentos realizados neste trabalho. Os valores das medidas de Micro - F1 são superiores as medidas de Macro - F1, pelo fato destas receber um valor/peso, de acordo com o tamanho das categorias. Para os experimentos realizados, a base de dados WebKB tem resultados superiores a base Reuters, pelo fato desta possuir um número menor de características, o que influencia num valor maior da média das distribuições das palavras para cada classe.

A interpretação e a posição das diferenças entre os classificadores SVM e RF, podem ser observadas por meio das Tabelas 10 e 11, onde é realizada a comparação por pares dos algoritmos de classificação, aqui é realizado o teste de Post-Hoc de Bonferroni, que realiza as comparações entre pares independentes. Nesta tabela em **Sig.** é possível observar os ajustes de Bonferroni para múltiplas comparações, onde os resultados destas comparações apresentam diferenças significativas entre os classificadores SVM e *Random Forest*.

O teste de Bonferroni fornece informações das comparacões par-a-par por meio das médias marginais estimadas para os níveis de fatores e interações entre fatores. Esta comparação entre pares produz comparação para todas as combinações de nível dos fatores especificados. As médias marginais estimadas não estão disponíveis para modelos multinomiais.

Tabela 10 – Comparação por pares F-measure Micro-F1, Reuters

CLASSIF. (I)	**CLASSIF. (J)**	**Diferença Média**	**Std.Error**	**Sig.**	**Intervalo de Confiança (95%)**	
					Limite Inf.	Limite Sup.
SVM	RF	0,031	0,003	0,000	0,025	0,037
RF	SVM	-0,031	0,003	0,000	-0,037	-0,025

Tabela 11 – Comparação por pares F-measure Micro-F1, WebKB

CLASSIF. (I)	**CLASSIF. (J)**	**Diferença Média**	**Std.Error**	**Sig.**	**Intervalo de Confiança (95%)**	
					Limite Inf.	Limite Sup.
SVM	RF	0,059	0,002	0,000	0,054	0,065
RF	SVM	-0,059	0,002	0,000	-0,065	-0,054

5.2.5 Testes Estatísticos comparando os grupos de características

A análise dos grupos formados pelas características, foi realizada para se obter informações de qual é o melhor grupo de características para os experimentos realizados. Após a transformação dos grupos, onde os mesmos foram transformados em grupos de 5% a 40%, passou a verificar que o grupo de características com maior predominância é o grupo de 30%, ou seja os melhores resultados dos experimentos apresentam-se com

este percentual. Grupos de características com valores maiores que 30% não apresentam melhorias na performance dos experimentos. Na análise estatística foi observada que a transformação dos grupos de características em grupos menores, dispõe de informações suficientes para classificação utilizando menos de 50% das informações dos documentos originais.

A Tabela 12, apresenta os subgrupos de características formados após a transformação das características, por meio do teste de Tukey (HSD), que compara as médias dos grupos duas a duas, verificou-se que a medida de *f-measure* apresenta a maior parte de seus resultados concentradas no grupo que possui 30% das características. Nota-se que o subgrupo mais relevante é o grupo de 30%, logo, é possível verificar que para grupos com valores maiores que o mencionado anteriormente os mesmos não apresentam diferenças significativas, ou seja, não acrescentam informação alguma ao experimento.

Tabela 12 – Subrupos de Características , F-measure Micro, Reuters

Grupos	**N**	**1**	**2**
5%	2	0,732	
10%	2	0,749	0,749
15%	2		0,755
20%	2		0,761
25%	2		0,763
30%	**2**		**0,765**
35%	2		0,767
40%	2		0,767
Sig.		0,155	**0,096**

De acordo com a Tabela 13, da base de dados Reuters, para a medida de validação *F-measure* que é uma medida harmônica entre a precisão e o *recall* é possível observar que a maior média dos grupos de características está presente para o grupo de 30% para valores maiores as médias se apresentam com mínimas diferenças significativas em relação ao grupo que aparece na maioria dos experimentos.

Tabela 13 – Estimação dos grupos de características - Reuters

Grupos de Características	**Média**	**Std. Error**	**Intervalo de Confiança**	
			Limite Inf.	Limite Sup.
5%	0,733	0,004	0,724	0,741
10%	0,749	0,004	0,740	0,758
15%	0,755	0,004	0,747	0,764
20%	0,762	0,004	0,753	0,770
25%	0,763	0,004	0,755	0,772
30%	**0,766**	**0,004**	**0,757**	**0,774**
35%	0,767	0,004	0,759	0,776
40%	0,767	0,004	0,758	0,776

As mesmas análises anteriormente comentadas, são feitas para a base de dados WebKB. A Tabela 14, apresenta os subgrupos de características formados após a transformação das características, por meio do teste de Tukey (HSD), que compara as médias dos grupos duas a duas, verificou-se que a medida de *f-measure* apresenta alguns de seus resultados concentradas no grupo que possui 35% das características. Logo, é possível verificar que para grupos com valores maiores que o mencionado anteriormente os mesmos não apresentam diferenças significativas, ou seja, não acrescentam informação alguma ao experimento.

Tabela 14 – Subgrupos de Características , F-measure Micro, WebKB

Grupos	**N**	**1**	**2**	**3**
5%	2	0,796		
15%	2	0,807		
20%	2		0,819	
25%	2		0,822	
10%	2		0,824	
30%	2			0,829
40%	2			0,836
35%	**2**			**0,837**
Sig.		0,404	0,117	**0,078**

De acordo com a Tabela 15, da base de dados WebKB, para a medida de validação *F-measure* que é uma medida harmônica entre a precisão e o *recall* é possível observar que a maior média dos grupos de características esta presente para o grupo de 35%, para valores maiores as médias e o intervalo de confiança encontram-se dentro do grupo de 35% das características, logo, diferenças significativas não são observadas.

Tabela 15 – Estimação dos grupos de características - Webkb

Grupos de Características	**Média**	**Std. Error**	**Intervalo de Confiança**	
			Lim. Inf.	Lim. Sup.
5%	0,797	0,004	0,788	0,805
10%	**0,825**	**0,004**	**0,816**	**0,833**
15%	0,808	0,004	0,800	0,816
20%	0,819	0,004	0,811	0,827
25%	0,822	0,004	0,814	0,831
30%	0,829	0,004	0,821	0,838
35%	**0,838**	**0,004**	**0,829**	**0,846**
40%	0,836	0,004	0,828	0,845

5.2.6 Testes Estatísticos comparando Word Embeddings

Nesta seção, realizou-se a análise comparativa dos *Word Embeddings*, ou seja, verificou-se os dois métodos adotados afim de obter informações em relação as palavras que foram transformadas em vetores em relação as outras medidas investigadas. Aqui, as comparações dos efeitos dos testes dentro dos grupos foram realizadas, onde, verificou-se a existência de diferenças entre os *embeddings*, Word2Vec e Glove, diferença dos *embeddings* entre os classificadores e dos *embeddings* entre os algoritmos de agrupamento. Estas diferenças foram observadas por meio dos valores da Esfericidade assumida, para as métricas de avaliação *precision*, *recall* e *f-measure*.

Verificou-se que os efeitos dos testes dentro dos grupos, comparando os dados entre os classificadores SVM e RF, para a base de dados Reuters é significativa. Na Tabela 16, é possível observar que pela significância tanto para a medida Micro - F1 quanto para a Macro - F1, as técnicas de *Word Embeddings* apresentam diferenças entre os métodos de Word2Vec e Glove, isto é comprovado pelas três medidas de avaliação do estudo.

Tabela 16 – Efeitos dos testes dentro dos grupos de *Word Embeddings*- Reuters

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,004	0,004	5	5	0,001	0,001	0,000	0,000
Recall	0,002	0,007	5	5	0,000	0,001	0,005	0,000
F-measure	0,004	0,006	5	5	0,001	0,001	0,000	0,000

As mesmas análises foram realizadas para a base de dados WebKb, onde mais uma vez foram observadas se existe diferenças entre o Word2Vec e o Glove. Pela Tabela 17 verificou-se que existe diferença significativa entre os métodos de vetorização de palavras tanto para as medidas de Micro - F1 como para Macro - F1.

Tabela 17 – Efeitos dos testes dentro dos grupos de *Word Embeddings*- WebKb

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,048	0,048	5	5	0,010	0,010	0,000	0,000
Recall	0,002	0,007	5	5	0,000	0,001	0,005	0,000
F-measure	0,065	0,006	5	5	0,013	0,001	0,000	0,000

Na análise das Tabela 18 e Tabela 19, o efeito dos testes dentro dos grupos de *Word Embeddings* x Classificadores SVM e RF, pelos valores da significância é possível observar

que existe diferença significativa entre o Word2Vec e Glove e os algoritmos de classificação de texto para a medida Macro - F1, de ambas as bases, Reuters e WebKb. Observou-se que apenas a medida *recall* não é significativa na Micro - F1 para abse Reuters e a *recall* e *f-measure* não é significativa para a base Webkb.

Tabela 18 – Efeitos dos testes dentro dos grupos de *Word Embeddings* x Classificador - Reuters

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,000	0,000	5	5	6,539E-5	9,314E-5	0,001	0,004
Recall	0,001	0,001	5	5	0,000	0,000	0,178	0,009
F-measure	0,000	0,001	5	5	5,469E-5	0,000	0,001	0,007

Tabela 19 – Efeitos dos testes dentro dos grupos de *Word Embeddings* x Classificador - WebKb

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,003	0,008	5	35	0,001	0,000	0,000	0,000
Recall	0,003	0,003	35	35	9,269E-5	8,346E-5	0,511	0,002
F-measure	0,001	0,002	5	35	0,000	6,746E-5	0,010	0,003

Nas tabelas a seguir 20 e 21, foram analisados se existe diferença significativa entre os *Word Embeddings* e os grupos de características criados. Pelo nível de significância a precisão, o *recall* e o *F-measure* apresentaram diferença quando utilizou-se a medida Macro - F1 para ambas as bases de dados em estudo. Logo, é possível informar pela tabela que as diferenças entre *Word Embeddings* e os grupos de caraterísticas são observadas quando utiliza-se os dados da medida Macro - F1. Para os dados da medida Micro - F1, as medidas de avaliação não significativas, ou seja, não apresenta diferença entre os *embeddings* e os grupos de características.

Tabela 20 – Efeitos dos testes dentro dos grupos de *Word Embeddings* x Grupos de Características - Reuters

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,001	0,002	35	35	2,029E-5	5,413E-5	0,045	0,005
Recall	0,003	0,003	35	35	9,269E-5	8,346E-5	0,511	0,002
F-measure	0,001	0,002	35	35	2,302E-5	6,746E-5	0,100	0,003

Tabela 21 – Efeitos dos testes dentro dos grupos de *Word Embeddings* x Grupos de Características - Webkb

	Soma de Qudr. Tipo III		**df**		**Desvio Padrão**		**Sig.**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**	**Micro F1**	**Macro F1**
Precision	0,008	0,006	35	5	0,000	0,001	0,000	0,000
Recall	0,001	0,001	5	5	0,000	0,000	0,178	0,009
F-measure	0,010	0,001	35	5	0,000	0,000	0,000	0,007

5.3 Comparação dos presentes resultados com trabalhos relacionados

Nesta Seção iremos abordar as comparações dos resultados obtidos na presente dissertação, após a realização do experimento com diferentes combinações em relação aos trabalhos relacionados mencionados na Seção (3).

Para a comparação com os trabalhos relacionados, selecionou-se apenas a melhor configuração dos resultados dos experimentos para cada um dos bancos de dados. Os resultados mais prováveis surgiram para a medida de *f-measure*, sendo para a base de dados Reuters a melhor combinação entre Glove - Kmeans com 30% dos grupos de características para o classificador SVM e para base WebKB a melhor combinação se deu para o Glove - Birch com 10% dos grupos de características também para o classificador SVM. Para o presente cálculo de comparação, utilizou-se a validação cruzada dos dados (*cross validation*), mesmo método utilizado nos trabalhos apresentados na Tabela 22.

É possível observar por meio da tabela que a melhor configuração aparece no trabalho de Roberto (PINHEIRO; CAVALCANTI; REN, 2015), para a base de dados Reuters, quando observa-se os valores de Micro-F1 **(0.93)** e Macro-F1 **(0.80)**, no entanto

a proposta do presente trabalho aparece com melhores resultados de Micro-F1 **(0.93)** e Macro-F1 **(0.90)** para a base de dados WebKB.

A proposta sugerida apresenta melhores resultados pelo fato desta, utilizar a base de dados WebKB, onde esta é composta por páginas da Web (por exemplo: Twitter, fóruns de discussão na Web), ou seja, os textos presentes nesta base são em sua maioria de natureza desestruturada, diferente dos documentos que compõem a base de dados Reuters (que contém notícias), textos menos desestruturados. Logo, os resultados encontrados nos levam a acreditar que o método proposto se aplica melhor em base de dados mais desestruturadas, o que nos leva a aprofundar mais os estudos neste tipo de característica, ou seja, testar este método em diversas outras bases de dados menos estruturada, para assim comprovar tal hipótese.

Os resultados de Micro-F1 e Macro-F1 obtidos pela proposta deste trabalho, é melhor quando se trabalha com documentos desestruturados pelo fato do método utilizar *Word Embeddings*, esta técnica realiza a captura de palavras muito utilizadas nos documentos, ou seja, obtém várias palavras de contextos diferentes e quando estas passam pelo processo de vetorização eles excluem só as palavras que não estão consideradas no *embedding*. Uma outra vantagem obtida a partir dos resultados é que nesta proposta criou-se grupos de palavras similares, contrário dos métodos apresentados nos demais trabalhos que simplesmente eliminam palavras dos documentos.

Tabela 22 – Comparação das configurações dos resultados

Trabalhos Relacionados	**Reuters**		**WebKB**	
	Micro F1	**Macro F1**	**Micro F1**	**Macro F1**
1. (PINHEIRO; CAVALCANTI; REN, 2015)	**0.93**	-	0.88	-
2. (LABANI et al., 2018)	-	**0.80**	-	0.65
3. (BHARTI; SINGH, 2015)	-	0.48	-	0.39
4. (UYSAL, 2016)	0.86	0.68	0.84	0.84
5. (JIN et al., 2015)	0.83	0.62	0.82	0.71
6. (FRAGOSO, 2016)	0.82	0.67	0.86	0.85
7. Proposta	0.79	0.61	**0.93**	**0.90**

6 Conclusão

No presente trabalho foi proposto um método de criação de grupos semânticos que visa reduzir a dimensionalidade das características. A redução da dimensionalidade ocorreu da seguinte forma, grupos semânticos foram criados, onde inicialmente as palavras das bases de dados passaram por uma vetorização utilizando dois métodos o Word2Vec e o Glove. Após a vetorização das palavras, foram aplicados os algoritmos de agrupamento que teve como função criar grupos de características reduzidos, ou seja, ocorreu a transformação das características para grupos menores de características em relação aos grupos originais.

O resultados dos experimentos mostraram que em 99% dos casos, o Glove mostrou-se superior ao Word2Vec na vetorização das palavras, juntamente com os algoritmos de agrupamento K-means, Birch e Agglomerative Clustering. Estes resultados só compravam que reduzir a dimensionalidade das características por meio de grupos semânticos é uma maneira eficaz.

Por meio dos testes estatísticos foi comprovado que o classificador SVM difere-se do classificador RF. Nos gráficos que realizam a comparação dos grupos semânticos criados em relação aos algoritmos de classificação, o classificador SVM mostrou-se melhor que o classificador RF, pelo fato das medidas de avaliação estarem sempre próximas a 1, ou seja, na classificação quanto mais o valores se aproximam do valor 1, maiores são as chances dos documentos relevantes terem sido classificados corretamente dentro de cada classe.

Os principais grupos semânticos para a base de dados Reuters e WebKB surgem quando utiliza-se o classificador de texto SVM, os grupos que são vetorizados pelo método do Glove e que utilizam os algoritmos de agrupamento K-means são os grupos de características de maior destaque na análise dos experimentos. A mesma vetorização aparece quando utiliza-se o classificador Random Forest, para as mesmas bases de dados o que diferencia os experimentos com este classificador dos outros é o tipo de algoritmo de agrupamento que apresenta resultado superior na a nálise, sendo ele o agrupamento realizado pelo algoritmo Birch.

Os resultados mais relevantes dos grupos semânticos encontram-se quando os grupos assumem valores de 30% para a base de dados Reuters e 35% para a base de dados WebKB. Isto fornece informações de que o método proposto consegui reduzir os grupos de características com menos de 50% das informações.

O método proposto de criação de grupos semânticos se diferencia dos trabalhos relacionados por este utilizar um algoritmo de classificação diferente dos outros trabalhos, ou seja, o algoritmo Random Forest, onde este foi aplicado para as mesmas bases de dados presentes nos trabalhos relacionados. Um outro diferencial deste trabalho é criar estes

grupos semânticos por meio da vetorização de palavras utilizando *Word Embeddings* com algoritmos de agrupamento. Estes algoritmos de agrupamento foram importantes para a redução da dimensionalidade das características na classificação de texto. Para analise dos resultados dos experimentos um método estatístico foi utilizado, para realizar comparações entre grupos de experimentos a ser investigados, o método estatístico é conhecido como Anova de medidas repetidas. A Anova consiste em reduzir a variabilidade não sistemática dos experimentos, resultando em um poder maior de detecção de efeitos exatos.

6.1 Contribuições

A principal contribuição do trabalho foi a proposta de um novo método de redução de dimensionalidade. O método proposto consegue fornecer resultados precisos por meio da criação dos grupos semânticos com menos de 50% das informações contidas nos grupos de características. Estes resultados podem ser obtidos por meio da utilização dos algoritmos de agrupamento e *Word Embeddings*, que apresentaram como resultado a redução da dimensionalidade das características na classificação de texto.

Para isto, comparações dos desempenhos dos métodos de *Word Embeddings* para seleção de características utilizando os algoritmos SVM e RF foram realizadas, o classificador SVM apresenta resultados superiores quando comparado com o classificador RF. Este resultado pode ser visto por meio das medidas de avaliação do classificador, a medida Micro - F1 apresenta uma média próximo de 1 (um) para as medidas de precisão, cobertura e *f-measure* do classificador de texto SVM, esta diferença pode ser vistas em todos os experimentos realizados.

Uma outra contribuição deste trabalho, pode ser percebida, na forma de vetorizar as palavras, aqui foram abordados os métodos de *Word embeddings*, após esta vetorização grupos de palavras foram criados a partir de algoritmos de agrupamento que reduziram o tamanho dos grupos de características originais. E a junção do *Words Embeddings* com os algoritmos de agrupamento, permitiu realizar comparações para se apresentar o melhor grupo semântico formado por eles. Em 99% dos resultados do experimento o Glove apresentou melhor vetorização de palavras em relação ao Word2Vec juntamente com os algoritmos de agrupamento K-means e Birch.

6.2 Limitações

As limitações vistas neste trabalho são em relação a transformação das características. Aqui o método de transformação tornou-se custoso computacionalmente em relação ao tempo de execução dos experimentos. Esta atividade apresentou alguns problemas de memória na máquina utilizada, pelo fato da transformação ser custosa, e além disto,

utilizar algoritmos de agrupamento para reduzir a dimensionalidade das características.

Uma outra dificuldade encontrada foi em relação ao algoritmo de agrupamento DBSCAN, onde o parâmetro **K** é definido de forma automática. Isso levou a criação de grupos com pouca informação e em alguns casos grupos vazios.

6.3 Trabalhos Futuros

Para os trabalhos futuros, sugere-se os seguintes melhoramentos:

- Realizar mais experimentos utilizando outras bases de dados além da Reuters e WebKb, para avaliar a eficácia do método proposto. O presente trabalho realizou a análise de seus métodos em apenas duas bases de dados, a análise de outras bases deverão ser incluídas e comparadas com os resultados já obtidos nesta dissertação;
- Avaliar a metodologia aplicada utilizando outros algoritmos de agrupamento, como por exemplo o KNN, o Mean-Shift, e o DBSCAN procurando aprofundar-se em analisar melhor a definição de seu parâmetro **K** e outros algoritmos de classificação, como por exemplo, o classificador Naïve Bayes que assume atributos condicionalmente independentes, como também testar a Entropia Máxima que reduz a dimensionalidade das características utilizando probabilidades, outros classificadores de textos vistos nos trabalhos relacionados também serão testados;
- Utilizar o método proposto em uma aplicação real, como por exemplo classificação de notícias em páginas Web, que visa apresentar ao leitor conteúdos de notícias mais próximo de seu gosto baseado na análise de dados históricos, como também aplicar em postagens feitas em Twitter para identificar padrões de comportamento de determinado assunto, como por exemplo, o índice de aceitação de um determinado produto.

Referências

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca**. [S.l.]: Bookman Editora, 2013. Citado 4 vezes nas páginas 25, 26, 28 e 29.

BARION, E. C. N.; LAGO, D. Mineração de textos. **Revista de Ciências Exatas e Tecnologia**, v. 3, n. 3, p. 123–140, 2015. Citado na página 25.

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003. Citado 2 vezes nas páginas 41 e 48.

BHARTI, K. K.; SINGH, P. K. Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. **Expert Systems with Applications**, Elsevier, v. 42, n. 6, p. 3105–3114, 2015. Citado 2 vezes nas páginas 52 e 90.

BOUCKAERT, R. R.; FRANK, E.; HALL, M.; KIRKBY, R.; REUTEMANN, P.; SEEWALD, A.; SCUSE, D. Weka manual for version 3-6-0. **University of Waikato, Hamilton, New Zealand**, 2008. Citado na página 63.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 31.

BÜTTCHER, S.; CLARKE, C. L.; CORMACK, G. V. **Information retrieval: Implementing and evaluating search engines**. [S.l.]: Mit Press, 2016. Citado na página 21.

CHANG, Y.-C.; CHEN, S.-M.; LIAU, C.-J. Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. **Expert Systems with Applications**, Elsevier, v. 34, n. 3, p. 1948–1953, 2008. Citado na página 65.

CHEN, J.; HUANG, H.; TIAN, S.; QU, Y. Feature selection for text classification with naïve bayes. **Expert Systems with Applications**, Elsevier, v. 36, n. 3, p. 5432–5435, 2009. Citado na página 65.

COLYER, A. The amazing power of word vectors. **The Morning Paper. Retrieved from https://blog. acolyer. org/2016/04/21/the-amazing-power-of-word-vectors/ Google Scholar**, 2016. Citado 3 vezes nas páginas 32, 41 e 42.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 28.

DEBOLE, F.; SEBASTIANI, F. An analysis of the relative hardness of reuters-21578 subsets. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 56, n. 6, p. 584–596, 2005. Citado na página 65.

DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. **Journal of the American society for information science**, American Documentation Institute, v. 41, n. 6, p. 391, 1990. Citado na página 47.

Edward Loper, S.; Ewan Klein. **Processamento da linguagem natural com Python**. [S.l.]: O'Reilly Media Inc., 2009. Citado na página 56.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Kdd**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citado na página 38.

FAHAD, A.; ALSHATRI, N.; TARI, Z.; ALAMRI, A.; KHALIL, I.; ZOMAYA, A. Y.; FOUFOU, S.; BOURAS, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. **IEEE transactions on emerging topics in computing**, IEEE, v. 2, n. 3, p. 267–279, 2014. Citado na página 34.

FENG, G.; GUO, J.; JING, B.-Y.; SUN, T. Feature subset selection using naive bayes for text classification. **Pattern Recognition Letters**, Elsevier, v. 65, p. 109–115, 2015. Citado na página 64.

FERNÁNDEZ-DELGADO, M.; CERNADAS, E.; BARRO, S.; AMORIM, D. Do we need hundreds of classifiers to solve real world classification problems. **J. Mach. Learn. Res**, v. 15, n. 1, p. 3133–3181, 2014. Citado 3 vezes nas páginas 23, 54 e 63.

FERREIRA, R.; CABRAL, L. de S.; LINS, R. D.; SILVA, G. P. e; FREITAS, F.; CAVALCANTI, G. D.; LIMA, R.; SIMSKE, S. J.; FAVARO, L. Assessing sentence scoring techniques for extractive text summarization. **Expert systems with applications**, Elsevier, v. 40, n. 14, p. 5755–5764, 2013. Citado na página 21.

FORMAN, G. An extensive empirical study of feature selection metrics for text classification. **Journal of machine learning research**, v. 3, n. Mar, p. 1289–1305, 2003. Citado na página 64.

FRAGOSO, R. C. P. Algoritmos de seleção de características personalizados por classe para categorização de texto. Universidade Federal de Pernambuco, 2016. Citado 5 vezes nas páginas 22, 23, 27, 53 e 90.

JIN, C.; MA, T.; HOU, R.; TANG, M.; TIAN, Y.; AL-DHELAAN, A.; AL-RODHAAN, M. Chi-square statistics feature selection based on term frequency and distribution for text categorization. **IETE journal of research**, Taylor & Francis, v. 61, n. 4, p. 351–362, 2015. Citado 2 vezes nas páginas 53 e 90.

JOACHIMS, T. **A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.** [S.l.], 1996. Citado 2 vezes nas páginas 22 e 27.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: SPRINGER. **European conference on machine learning**. [S.l.], 1998. p. 137–142. Citado 3 vezes nas páginas 22, 28 e 30.

LABANI, M.; MORADI, P.; AHMADIZAR, F.; JALILI, M. A novel multivariate filter method for feature selection in text classification problems. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 70, p. 25–37, 2018. Citado 2 vezes nas páginas 51 e 90.

LANDAUER, T. K.; FOLTZ, P. W.; LAHAM, D. An introduction to latent semantic analysis. **Discourse processes**, Taylor & Francis, v. 25, n. 2-3, p. 259–284, 1998. Citado na página 25.

LEOPOLD, E.; KINDERMANN, J. Text categorization with support vector machines. how to represent texts in input space? **Machine Learning**, Springer, v. 46, n. 1-3, p. 423–444, 2002. Citado na página 28.

LI, C. H.; PARK, S. C. Neural network for text classification based on singular value decomposition. In: IEEE. **Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on**. [S.l.], 2007. p. 47–52. Citado 2 vezes nas páginas 22 e 23.

MACHADO, A. P.; FERREIRA, R.; BITTENCOURT, I. I.; ELIAS, E.; BRITO, P.; COSTA, E. Mineração de texto em redes sociais aplicada à educação a distância. **Colabor@-A Revista Digital da CVA-RICESU**, v. 6, n. 23, 2010. Citado 2 vezes nas páginas 21 e 25.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability**. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 36.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. et al. **Introduction to information retrieval**. [S.l.]: Cambridge university press Cambridge, 2008. v. 1. Citado na página 47.

MANNING, C. D.; SCHÜTZE, H. **Foundations of statistical natural language processing**. [S.l.]: MIT press, 1999. Citado na página 40.

MCCORMICK, C. **Word2vec tutorial-the skip-gram model**. [S.l.]: Retrieved, 2016. Citado 3 vezes nas páginas 44, 45 e 46.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citado 4 vezes nas páginas 23, 41, 42 e 43.

MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2013. p. 746–751. Citado na página 47.

NLTK. 2016. <https://www.nltk.org/>. [Online; accessed 09-March-2018]. Citado na página 56.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543. Citado 3 vezes nas páginas 23, 46 e 48.

PINHEIRO, R. H.; CAVALCANTI, G. D.; REN, T. I. Data-driven global-ranking local feature selection methods for text categorization. **Expert Systems with Applications**, Elsevier, v. 42, n. 4, p. 1941–1949, 2015. Citado 6 vezes nas páginas 22, 49, 51, 64, 89 e 90.

RAKOTOMAMONJY, A. Variable selection using svm-based criteria. **Journal of machine learning research**, v. 3, n. Mar, p. 1357–1370, 2003. Citado na página 28.

ROUSSEAU, F.; KIAGIAS, E.; VAZIRGIANNIS, M. Text categorization as a graph classification problem. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. [S.l.: s.n.], 2015. v. 1, p. 1702–1712. Citado na página 53.

SALTON, G.; MCGILL, M. **The SMART retrieval system—experiments in automatic document retrieval**. [S.l.]: Prentice Hall Inc., Englewood Cliffs, NJ, 1971. Citado na página 65.

SANTANA, R. A importância do papel do profissional da ciência da informação nos processos de recuperação de conteúdos digitais estruturados. **Ensino e pesquisa em biblioteconomia no Brasil: a emergência de um novo olhar. Marília: Cultura acadêmica**, p. 145–154, 2008. Citado na página 21.

SCIKIT-LEARN. 2018. <scikit-learn.org/stable/index.html>. [Online; accessed 09-March-2018]. Citado 4 vezes nas páginas 34, 35, 36 e 39.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM computing surveys (CSUR)**, ACM, v. 34, n. 1, p. 1–47, 2002. Citado 5 vezes nas páginas 21, 22, 26, 47 e 49.

SERAPIÃO, P. R. B.; SUZUKI, K. M. F.; MARQUES, P. M. de A. Uso de mineração de texto como ferramenta de avaliação da qualidade informacional em laudos eletrônicos de mamografia. **Radiologia Brasileira**, Radiologia Brasileira, v. 43, n. 2, p. 103–107, 2010. Citado 2 vezes nas páginas 21 e 25.

SOCHER, R.; BAUER, J.; MANNING, C. D. et al. Parsing with compositional vector grammars. In: **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2013. v. 1, p. 455–465. Citado na página 47.

TAN, P.-N. et al. **Introduction to data mining**. [S.l.]: Pearson Education India, 2006. Citado na página 32.

TANG, B.; KAY, S.; HE, H. Toward optimal feature selection in naive bayes for text categorization. **IEEE transactions on knowledge and data engineering**, IEEE, v. 28, n. 9, p. 2508–2521, 2016. Citado na página 64.

TELLEX, S.; KATZ, B.; LIN, J.; FERNANDES, A.; MARTON, G. Quantitative evaluation of passage retrieval algorithms for question answering. In: ACM. **Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval**. [S.l.], 2003. p. 41–47. Citado na página 47.

TENSORFLOW Word2Vec tutorial. 2016. <https://www.tensorflow.org/tutorials/word2vec>. [Online; accessed 20-March-2018]. Citado na página 47.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 48th annual meeting of the association for computational linguistics**. [S.l.], 2010. p. 384–394. Citado na página 47.

UĞUZ, H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. **Knowledge-Based Systems**, Elsevier, v. 24, n. 7, p. 1024–1032, 2011. Citado na página 64.

UYSAL, A. K. An improved global feature selection scheme for text classification. **Expert systems with Applications**, Elsevier, v. 43, p. 82–92, 2016. Citado 3 vezes nas páginas 21, 52 e 90.

WANG, H.; WANG, L.; YI, L. Maximum entropy framework used in text classification. In: IEEE. **Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on**. [S.l.], 2010. v. 2, p. 828–833. Citado na página 53.

WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; PHILIP, S. Y. et al. Top 10 algorithms in data mining. **Knowledge and information systems**, Springer, v. 14, n. 1, p. 1–37, 2008. Citado 2 vezes nas páginas 28 e 29.

XU, R.; WUNSCH, D. **Clustering**. [S.l.]: John Wiley & Sons, 2008. v. 10. Citado 2 vezes nas páginas 33 e 34.

YANG, J.; LIU, Y.; ZHU, X.; LIU, Z.; ZHANG, X. A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. **Information Processing & Management**, Elsevier, v. 48, n. 4, p. 741–754, 2012. Citado na página 64.

YU, L.; LIU, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: **Proceedings of the 20th international conference on machine learning (ICML-03)**. [S.l.: s.n.], 2003. p. 856–863. Citado na página 27.

ZHANG, W.; YOSHIDA, T.; TANG, X. Text classification based on multi-word with support vector machine. **Knowledge-Based Systems**, Elsevier, v. 21, n. 8, p. 879–886, 2008. Citado 2 vezes nas páginas 28 e 29.