UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ITALO FRANCYLES SANTOS DA SILVA

CLASSIFICAÇÃO DE SUCESSO MUSICAL BASEADA EM LETRAS

São Luís
2017

ITALO FRANCYLES SANTOS DA SILVA

CLASSIFICAÇÃO DE SUCESSO MUSICAL BASEADA EM LETRAS

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº Dr. João Dallyson Sousa de Almeida

São Luís

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Silva, Italo Francyles Santos da.
Classificação de sucesso musical baseada em letras / Italo Francyles Santos da Silva. - 2017.
55 f.

Orientador(a): João Dallyson Sousa de Almeida.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2017.

1. Características textuais. 2. Classificação. 3. Letras. 4. Sucesso musical. 5. SVM. I. Almeida, João Dallyson Sousa de. II. Título.

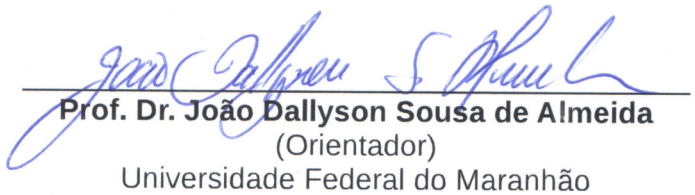
ITALO FRANCYLES SANTOS DA SILVA

CLASSIFICAÇÃO DE SUCESSO MUSICAL BASEADA EM LETRAS

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

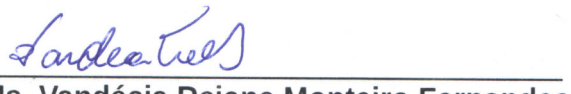
Aprovada em 27 de janeiro de 2017

BANCA EXAMINADORA



Prof. Dr. João Dallyson Sousa de Almeida
(Orientador)
Universidade Federal do Maranhão

Prof. Dr. Geraldo Braz Junior
Examinador 1
Universidade Federal do Maranhão



Profa. Ms. Vandécia Rejane Monteiro Fernandes
Examinador 2
Universidade Federal do Maranhão

São Luís
2017

À minha avó, Maria Santana, e à minha mãe, Elza Helena.

Agradecimentos

Em primeiro lugar, a Deus. Até aqui me ajudou o Senhor.

À minha família, por todo amor, carinho, e dedicação. Em especial, à minha avó, Maria Santana e à minha mãe, Elza Helena, pelo incentivo a cursar esta graduação e também ao meu pai, Paulo Henrique, pelo apoio diuturno.

Aos amigos do grupo Los Computeros, pela amizade e todo apoio dado ao longo desta caminhada.

Aos amigos de longa data do grupo Terceirão. Apesar da distância, a amizade permanece forte.

Aos amigos da UNASUS/UFMA, principalmente Dilson, Aldrea e Alisson pela oportunidade de trabalhar com eles, o que contribuiu bastante para meu crescimento profissional.

Ao meu orientador, João Dallyson, por todo apoio e pela paciência.

A todos que me acompanharam nesta jornada, e que contribuíram direta ou indiretamente na minha formação acadêmica.

“Não saber é o primeiro estágio para o conhecimento” - René Descartes

RESUMO

A classificação de texto é uma tarefa dentro da mineração de texto na qual documentos textuais são categorizados em classes predefinidas. Este processo pode ser aplicado em vários cenários, inclusive, o musical. A letra de uma música possui consigo grande parte dos significados semânticos de uma canção. E com base nisso, este trabalho apresenta uma proposta para a classificação de sucesso musical baseada em letras, na qual as músicas serão categorizadas em caso de sucesso ou insucesso a partir da análise das características de suas letras, utilizando o classificador máquina de vetores de suporte (SVM). Essa abordagem de classificação pode auxiliar artistas no momento de composição de suas canções, pois, lançar uma música de sucesso pode lhes prover ganhos não só financeiros, como também nas relações profissionais. O trabalho também explana sobre os modelos Bag-of-words, Doc2vec e a concatenação destes para representação de características textuais, fazendo uma análise comparativa entre estas quanto à sua aplicação na proposta desenvolvida.

Palavras-chaves: Classificação. Sucesso musical. Letras. SVM. Características textuais. Análise Comparativa.

ABSTRACT

Text classification is a text mining task wherein textual documents are categorized in predefined classes. This process can be applied in many scenarios including musics. Lyrics has a great part of a song's semantic meanings. Based on it, this work presents a proposal to lyrics-based success musical classification in witch the songs will be categorized in case of success or unsuccess using SVM classifier. This approach of classification can help artists when composing their songs because the publishment of a success song provide financial gains and professional relationship improvements. This work also explores the models Bag-of-words, Doc2vec and the concatenation of these models to represent textual features, making a comparative analysis between them as their application in the developed proposal.

Keywords: Text Classification. Success musical. Lyrics. SVM. Textual Features. Comparative Analysis.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
API	Application Programming Interface
BOW	Bag-of-words
CBOW	Continuous Bag-of-words
DBOW	Distributed Bag-of-words
DMPV	Distributed Memory Paragraph Vector
IDF	Inverse Document Frequency
LOO	Leave-one-out
LSA	Latent Semantic Analysis
MFCC	Mel Frequency Cepstral Coefficient
MLP	Multi-layer Perceptron
NMO	Número Mínimo de Ocorrências
PLN	Processamento de Linguagem Natural
RBF	Radial Basis Function
RIM	Recuperação de Informação Musical
TF	Term Frequency
SG	Skip-gram
SGD	Stochastic Gradient Descent
SMS	Short Message Service
SVM	Support Vector Machine

Lista de símbolos

γ	Parâmetro gama utlilizado na função do Kernel RBF

Sumário

1 Introdução

A música é considerada parte da vida das pessoas. É comum encontrar indivíduos ouvindo música em seus dispositivos portáteis em vários lugares, como em casa, no ambiente de trabalho, durante atividades físicas, etc. A popularização da tecnologia da informação e da internet intensificou cada vez mais a relação entre as pessoas e a música, fazendo com que a forma padrão de consumir música se alterasse. Assim, com a criação de novas formas de se ouvir, adquirir e compartilhar músicas, como serviços de streaming, sites e repositórios digitais, as pessoas têm acesso ao conteúdo musical de seus artistas preferidos a todo momento e em qualquer lugar do mundo.

Uma das áreas que lida com a grande quantidade de informação musical disponível na web é a Recuperação de Informação Musical (RIM). Os processos de RIM podem, por exemplo, auxiliar usuários a encontrar músicas relevantes às suas necessidades. Sistemas dessa área também realizam tarefas em análise automática de música, como, por exemplo, classificação de gêneros, similaridade de ritmos e também análise de letras.

A letra carrega grande parte dos significados semânticos de uma música. Isto torna a sua análise mais desafiadora. Quando documentadas em texto, as letras podem ser vistas também como objeto de estudo em processamento de linguagem natural, podendo, assim, ser submetidas, também, à tarefa de classificação.

Para classificar textos em categorias ou classes pré-fixadas, é preciso analisar suas peculiaridades de forma a individualizá-los mesmo quando pertencem a uma mesma categoria. Existem métodos capazes de fazer esta seleção de características. Por exemplo, o modelo Bag-of-words (HARRIS, 1954), que é amplamente utilizado em tarefas de classificação de texto, extrai características por meio de métricas de contagem de palavras relevantes ao contexto ao qual o documento faz parte e as representa em um modelo de espaço vetorial. Outra abordagem que surge como alternativa ao Bag-of-words é a representação vetorial distribuída de parágrafos, também conhecida como Doc2vec (LE; MIKOLOV, 2014). Nesse modelo, os documentos também são representados por vetores, mas as características são extraídas por meio de treinamento não supervisionado.

A classificação de textos é um processo de aprendizado supervisionado. Logo, utiliza-se uma base de dados textuais (*corpus*) previamente rotulada, a qual servirá como fonte de conhecimento para que o classificador consiga categorizar os documentos da melhor forma possível. Para isso, além de meios de extração e representação de características, é preciso escolher um classificador. Como exemplos de classificadores,

pode-se citar o K-Nearest Neighboors (KNN), Naive Bayes e as Máquinas de vetores de suporte (SVM)(CORTES; VAPNIK, 1995).

Neste trabalho, o processo de classificação de texto é aplicado à música propondo uma abordagem que classifique-a como caso de sucesso ou insucesso baseando-se na sua letra. Esta proposta pode servir como um auxílio à gravadora e ao artista que, ao escrever uma letra de música, teria como verificar se esta tem chances de ser um sucesso ou não baseado na fonte de conhecimento construída para este processo. Adolfo (1997) retrata que compositores devem ter um estilo próprio de composição e que isso é uma das coisas mais importantes na criação músicas. Acredita-se, então, que isso pode ser distintivo o suficiente para identificar a qual classe a música pertence.

O trabalho aborda as fases do processo de classificação, desde a criação da base de dados de letras, passando pela etapa de pré processamento, em seguida pela extração e representação de características e, por fim, discutirá os resultados alcançados com o classificador SVM.

Este trabalho também dá um enfoque às abordagens de extração e representação de características textuais, explanando seu funcionamento. Por fim, far-se-á, por meio dos resultados da classificação, uma análise comparativa entre os modelos utilizados, a saber, Bag-of-words, representação vetorial distribuida de parágrafos (Doc2vec) e uma representação obtida pela concatenação destas duas.

1.1 Justificativa

A música e o complexo industrial e empresarial que a cerca são atualmente pertencentes a um conjunto único, influindo na criação, produção, divulgação, distribuição e no consumo de produtos musicais (TROTTA, 2009). Com base nisso, é possível inferir que os gastos com a produção, gravação e divulgação de um artista e de suas músicas são elevados, pois há uma estrutura complexa que depende dos rendimentos deste artista para se manter erguida.

Em casos de artistas independentes, isto é, sem o suporte de uma gravadora, estes podem optar pelo uso de ferramentas de produção menos custosas financeiramente. Com o avanço das tecnologias digitais, é possível que produções caseiras ou independentes tenham padrões profissionais. Por outro lado, a distribuição é dificultada pela enorme complexidade e altos custos envolvidos no processo (CASTRO, 2008).

A música é um conjunto entre instrumental e letra. A letra traz consigo grande parte dos significados semânticos da canção, pois, é através dela que é transmitida a mensagem principal da música ao seu ouvinte. Sendo assim, seria importante que

houvesse uma ferramenta que pudesse auxiliar o compositor em seu processo de criação informando-lhe se a letra composta pode fazer sucesso ou não. Isto poderia evitar situações em que se investe muito dinheiro na produção e divulgação de uma canção e, no final, ela não agrada o público, tornando-se um insucesso.

Lançar uma música de sucesso pode trazer bons rendimentos a um artista e à estrutura empresarial que o cerca, não somente em questões financeiras, mas também na garantia de maior prestígio por parte do público o que acarreta em melhoras nas relações profissionais.

1.2 Objetivos

O objetivo deste trabalho é propor um método computacional para classificar músicas em casos de sucesso e insucesso baseada no conteúdo de suas letras utilizando o classificador SVM. Os objetivos específicos são:

- Corroborar o processo de classificação de texto quanto a sua aplicação em letras de música.
- Verificar se as abordagens de extração e representação de características, amplamente utilizadas em outras tarefas de categorização, conseguirão capturar as peculiaridades das letras de músicas acarretando em resultados de classificação satisfatórios.
- Avaliar os resultados da classificação de letras com o SVM para cada abordagem de extração e representação de características e fazer uma análise comparativa entre estas.

1.3 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: O capítulo 2 apresenta alguns trabalhos relacionados a este, relatando brevemente como eles contribuíram para o desenvolvimento da proposta deste. O capítulo 3 apresenta a fundamentação teórica utilizada no desenvolvimento do presente trabalho. São abordados o conceito de mineração de texto e o processo de classificação de texto em uma forma geral e como suas fases podem ser aplicadas na classificação de letras de música. Neste capítulo, relata-se as etapas de pré processamento e também sobre as abordagens de extração e representação de características utilizadas na proposta deste trabalho. O capítulo 4 discorre sobre os recursos computacionais e ferramentas utilizadas no desenvolvimento do método proposto, e apresenta este método. Este capítulo também explica o processo de criação da base de dados (*corpus*) e a metodologia empregada nos experimentos

com as abordagens de representação de texto. O capítulo 5 apresenta e discute os resultados obtidos com os experimentos, fazendo também uma análise comparativa do desempenho do classificador SVM para cada modelo de representação de texto. Por fim, o capítulo 6 apresenta as conclusões sobre a proposta e também propõe formas de melhorá-la em trabalhos futuros.

2 Trabalhos Relacionados

Este capítulo apresenta o referencial teórico utilizado para o desenvolvimento deste estudo. Os trabalhos aqui mencionados referem-se ao processo de classificação de textos englobando abordagens de extração e representação de características e também sua aplicação em diferentes cenários, principalmente em classificação de músicas baseada em letras.

Gasperin e Lima (2000) apresentam um levantamento dos modelos estatísticos existentes para o tratamento de diversos aspectos ligados ao processamento de linguagem natural (PLN). Exploram-se fundamentos matemáticos, como teoria de probabilidades e teoria da informação consolidando sua utilidade em muitas tarefas de PLN, como classificação de textos. Também são abordados os problemas encontrados quando se trabalha com bases de dados textuais devido sua natural inconsistência e como a aplicação de técnicas de pré processamento, como eliminação de símbolos e tokenização, é essencial para uma boa análise. É introduzido também o conceito de *corpus* oportunista, que são aqueles cujo conteúdo é aquilo que se deseja.

Medeiros (2004) mostra uma visão geral sobre possíveis aplicações do processo de classificação de texto, desde que os documentos do *corpus* estejam organizados cuidadosamente em categorias preestabelecidas de acordo com o conteúdo que os compõe. Também é feito um estudo experimental sobre a utilização do algoritmo SVM na tarefa de classificação de documentos quanto a pertencerem ou não a determinada categoria, aplicando o modelo Bag-of-words (BOW) para extração e representação de características textuais.

Santos et al. (2012) elaboraram experimentos com análise semântica latente (LSA) para a avaliação automática de respostas discursivas e mostra como a classificação de texto pode auxiliar especialistas de um domínio em suas tarefas. Apesar de utilizar uma abordagem diferente de classificação em relação aos demais trabalhos, os autores, por sua vez, fizeram uso das principais técnicas de pré processamento de texto, como remoção de stopwords.

Uma aplicação possível para a classificação de textos é em análise de sentimento. Dosciatti e Ferreira (2013) estudaram a aplicação do SVM em uma solução multiclasse referente às emoções a serem captadas em textos. Os vetores de características textuais foram construídos de acordo com a metodologia TF-IDF em conjunto com técnicas matemáticas para a redução da dimensionalidade. O *corpus* foi composto por documentos em português, construído pelos autores. Eles enfatizaram a carência de bases de dados textuais em língua portuguesa.

Outro experimento em análise de sentimento que utiliza um *corpus* em português foi feito por Franca e Oliveira (2014). O trabalho observa o nível de aprovação dos brasileiros em relação aos protestos que aconteceram entre junho e agosto de 2013 por meio de tweets coletados neste período. Os autores reforçam a necessidade de se conhecer a base de dados textuais para que se utilizem as técnicas de pré processamento de forma correta.

Dıaz-Galiano e Montejo-Ráez (2015) apresentam a análise de polaridade (positivo/negativo) de tweets utilizando também o classificador SVM. O corpus é formado por textos em espanhol. O trabalho, entretanto, faz uso da abordagem de vetores pré treinados de palavras e parágrafos (Word2vec e Doc2Vec, respectivamente) para representar as características textuais em contraste ao método bag-of-words amplamente utilizado. Os vetores de palavras foram obtidos com o treinamento do *corpus* da Wikipedia[1]. Já os vetores de parágrafos foram treinados a partir da coleção de tweets. Os resultados alcançados da classificação com Word2vec e Doc2vec isolados não foram altos (51% e 42% de acurácia), entretanto, quando combinados, apresentam significativa melhora (63% de acurácia). Apesar de não serem os resultados esperados, os autores os consideram bastante promissores.

Aguiar e Prati (2015) retratam o processo de classificação automática de spam em mensagens de texto (SMS). Esta tarefa mostra-se desafiadora, pois os SMS costumam ser curtos devido a utilização de gírias, abreviações e emoticons. Sendo assim, os autores aplicam técnicas de pré processamento como dicionários de gírias e desambiguação de texto para contornar este problema. O trabalho investiga a utilização de vetores de característica construídos pela concatenação entre vetores BOW e vetores Doc2vec no processo de classificação em comparação à técnica baseada em dicionários apresentada por Silva et al. (2014), e atesta que estes métodos são bastante competitivos.

A combinação entre vetores de característica Bag-of-words e a representação de parágrafos também é abordada por Lilleberg, Zhu e Zhang (2015). A construção dos vetores de parágrafos, conforme proposto pelos autores, se dá pela média aritmética entre os vetores que representam as palavras de um texto construídos com o Word2vec. O experimento consiste em classificar textos em vinte, dez e duas categorias. Os autores alcançaram bons resultados de acurácia com o SVM na classificação em duas classes (entre 80% - 90%). Contudo, à medida em que se aumentou o número de categorias, a acurácia diminuiu, ficando entre 60% - 70% no experimento com dez classes e 50% -60% para vinte classes.

Os trabalhos relacionados à classificação de músicas baseada em letras caracterizam-se pela necessidade de se criar uma base de dados de letras para as tarefas

[1] Wikipedia em espanhol pode ser acessada em https://es.wikipedia.org/

propostas.

Mahedero et al. (2005) relatam experimentos com o uso de técnicas de processamento de linguagem natural para a análise de letras de músicas. Segundo os autores, as letras codificam uma parte importante da semântica de uma música, logo, sua análise complementa os metadados acústicos e culturais e é fundamental para o desenvolvimento de sistemas de recuperação de informação musical mais completos. Os experimentos foram realizados em identificação de idioma, extração de estruturas (título, refrão, versos), categorização e análise de similaridade de artistas, e alcançaram resultados promissores.

Hu, Downie e Ehmann (2009) abordam a classificação de humor em música baseada em letras. O trabalho examina a ideia de que letras de música podem desempenhar melhoras em classificação de humor quando associadas ao áudio. É descrita a construção de uma base de dados com 5.585 músicas, divididas em 18 categorias de humor. O trabalho também relata a extração de características utilizando as técnicas stemming, part-of-speech e bag-of-words, e enfatiza o tratamento diferenciado que se deve dar às letras quanto ao pré processamento, visto que elas podem conter elementos impróprios para a análise, como indicadores de seções (por exemplo, solo, refrão) e repetições. Os autores verificam que os resultados da classificação baseada somente em letras supera a baseada em áudio nos casos em que os significados semânticos da categoria estão atrelados à letra. Entretanto, na maioria das categorias, a classificação de áudio em conjunto com letras apresentou melhores resultados. Os autores também usam o classificador SVM em seus experimentos.

Fell e Sporleder (2014) realizaram experimentos com diferentes formas de extração de característica, como o modelo bag-of-ngrams (variação do bag-of-words), assim como modelos mais complexos baseados em vocabulário, estilo, semântica e estrutura da música. O trabalho objetiva concatenar estes modelos com propósito de ganho de performance para as seguintes tarefas de classificação: detecção de gêneros, distinção de melhores e piores músicas e determinar o tempo aproximado de publicação de uma música. O classificador utilizado nestes experimentos foi o SVM através de sua implementação na ferramenta WEKA (HALL et al., 2009).

Os experimentos relatados por Abburi et al. (2016) tratam sobre a detecção de sentimento em músicas Telugu[2] baseada em sua natureza, isto é, em sua constituição (texto e áudio). Estes experimentos seguiram duas vertentes: baseados em conhecimento e aprendizado de máquina. As músicas foram extraídas diretamente do YouTube[3]. As características textuais foram representadas com a abordagem Doc2vec. Quanto à classificação, as músicas deveriam se encaixar em duas classes: feliz ou

[2] Músicas Telugu são típicas da região sul da Índia

[3] https://www.youtube.com/

triste. Os resultados com o classificador SVM na classificação somente de letras de música alcançam 67,5% de acurácia. Já os resultados envolvendo características de texto e áudio alcançam 91,2%. Segundo o autor, a escolha deste classificador justifica-se porque este é conhecido por discriminar as características com mais qualidade.

Diferente dos trabalhos supracitados, o presente trabalho propõe a classificação de músicas em casos de sucesso e insucesso baseada em letras. Para isso, foi construído um *corpus* devidamente rotulado. O processo de classificação realizado aqui atua de forma semelhante aos trabalhos relacionados. Sobre o *corpus*, serão aplicadas as técnicas de pré processamento. Seguinte a isto, a extração e representação de características seguirá as abordagens Bag-of-words, Doc2vec e a concatenação destas. Por fim, os vetores resultantes destas três abordagens serão submetidos à classificação com o SVM, cujo desempenho servirá para verificar qual destas representações obteve os melhores resultados.

3 Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos utilizados no desenvolvimento deste trabalho, imprescindíveis para o entendimento das técnicas adotadas para alcançar os objetivos. São abordados os conceitos de mineração de texto e o processo de classificação, dando ênfase na explicação das suas etapas de pré processamento, extração e representação de características textuais.

Quanto à fase de pré processamento, são explanadas as seguintes técnicas: eliminação de símbolos e caracteres especiais, tokenização e os métodos de redução de dimensionalidade, a saber, o número mínimo de ocorrências de um termo e lista de stopwords. Sobre as etapas de extração e representação de características, são apresentados os modelos Bag-of-words e Vetores de parágrafos pré-treinados.

Por fim, este capítulo também apresenta o método Máquina de Vetores de Suporte (SVM) utilizado para a tarefa de classificação; e as técnicas de validação de resultados.

3.1 Mineração de textos

A mineração de textos, ou mineração de dados textuais, é um campo de estudo em processamento de linguagem natural, e, por conseguinte, em inteligência artificial, que objetiva extrair conhecimento útil a partir de fontes textuais não estruturadas, organizando informações que serão utilizadas como referência no futuro. Assemelha-se com a mineração de dados pois ambas se baseiam em um conjunto de amostras de exemplos coletados. A composição destes exemplos é diferente nos dois casos, pois, a mineração de dados lida com bases de dados categóricos e numéricos e a mineração de textos trabalha com documentos textuais em formato livre (MEDEIROS, 2004). Contudo, há similaridade nos métodos de aprendizagem utilizados nestes processos. Isso justifica o fato de que textos precisam ser representados em forma numérica similar à utilizada em mineração de dados (WEISS et al., 2010), pois esses métodos requerem que os dados estejam nesta representação.

O processo de mineração de texto, como aponta Medeiros (2004), acontece em duas etapas principais: extração de informação e mineração de dados. A primeira etapa é de bastante importância em todo o processo. Nesta fase, ocorre o pré-processamento do conjunto textual a fim de eliminar ruídos e reduzir a dimensionalidade dos dados, permitindo que se extraiam as principais características daquele conjunto textual. Nesta fase, ocorre também a representação dos textos em forma numérica. A segunda

etapa consiste na aplicação de algoritmos de aprendizado de máquina voltados para a realização da tarefa para a qual será utilizado o conjunto de textos, como, por exemplo, agrupamento ou classificação.

A classificação ou categorização de texto consiste na classificação de um texto em categorias ou classes pré-fixadas, baseando-se na análise de suas características. Para a realização desta tarefa são utilizadas técnicas de aprendizado supervisionado, em que se utiliza uma base de dados textuais (*corpus*) previamente rotulada, que servirá como fonte de conhecimento para que estas técnicas consigam predizer em qual classe um novo texto fornecido irá se encaixar. As fases que compreendem o processo de classificação de texto são:

1. Coleta de documentos textuais: nesta fase, ocorre a criação da base de dados textuais, ou *corpus*.
2. Pré processamento: nesta etapa, os documentos são submetidos a um conjunto de técnicas que visam reduzir ruídos e preservar informações relevantes, dessa forma, preparando-os para as etapas seguintes.
3. Extração e Representação de características: nesta fase, ocorre a aplicação de metodologias responsáveis pela extração de características dos documentos e sua padronização por meio de uma representação numérica.
4. Treinamento: Os documentos previamente rotulados são apresentados à técnica de aprendizado de máquina a fim de extrair conhecimento necessário para classificar novos documentos.
5. Classificação: Documentos textuais são classificados com base na fonte de conhecimento adquirido na fase anterior e organizados nas categorias predefinidas.

Este trabalho propõe aplicar o processo de classificação de textos sobre um *corpus* composto por letras de músicas brasileiras previamente rotuladas em duas classes: músicas de sucesso e músicas de insucesso; utilizando técnicas de pré-processamento de textos e extração de características e submetendo estes dados à técnica de aprendizado supervisionado Máquina de Vetores de Suporte (SVM). As seções seguintes apresentarão mais detalhes sobre as etapas de pré processamento e representação de características.

3.2 Pré processamento de dados textuais

Uma base de dados textuais, ou *corpus*, é uma coleção especial de material textual, coletado conforme certos critérios (GASPERIN; LIMA, 2000). Em geral, são

extensos, exigindo muitos recursos computacionais para manipulá-los. Um *corpus* traz consigo alguns desafios, como a falta de estruturação, repetição de palavras com o mesmo sentido, presença de sinais de pontuação e quantidade de palavras irrelevantes para o objetivo da classificação. Estes ruídos são danosos para a representação de textos, sendo, portanto, necessário que se realize a etapa de pré processamento para adequar este conteúdo textual para as etapas sequentes a esta.

Esta seção trata das técnicas de pré processamento de texto utilizadas nos experimentos deste trabalho.

3.2.1 Eliminação de símbolos e caracteres especiais

Uma das principais técnicas utilizadas no pré-processamento de texto é a eliminação de símbolos e caracteres especiais. Apesar de estes elementos serem importantes para a organização de um texto, bem como sua leitura e interpretação, não têm muito valor para o processo de classificação, pois representam ruídos no texto e são irrelevantes quanto à composição das suas principais características. Nesta etapa, os seguintes conjuntos de símbolos são removidos:

- Sinais de pontuação (“!”, “?”, “.”, “,”, “;”, “:”, “...”, “ ‘ “, “-”)
- Números cardinais e ordinais (escritos de forma numérica e acompanhados de símbolo)
- Parênteses, colchetes, chaves e demais símbolos (“>”, “<”, “+”, “=”, “\”, “/”, “#“, “%”, “&“)
- Excessos de espaços em branco, quebras de linha e tabulações

3.2.2 Tokenização

A tokenização consiste em separar os elementos constituintes do texto, identificando cada palavra que possivelmente comporá a representação estatística deste. Primeiro, estabelece-se um token delimitador para a quebra do texto em palavras, por exemplo, o espaço em branco, tabulação ou nova linha. Assim, com a identificação do delimitador escolhido, o texto é separado em tokens que são as palavras que compõem o texto. A tokenização se torna mais eficiente quando aplicada em conjunto com a eliminação de símbolos e caracteres especiais. Caso esta última técnica não seja utilizada, há o risco de que, por exemplo, um ou mais tokens sejam compostos pela junção de palavras e símbolos próximos. Também poderá haver que símbolos sozinhos, ou juntos de outros, formem tokens. Esse problema mantém a dimensão do espaço de características do texto muito alta, e bastante ruidosa. A Tabela 1 mostra os efeitos da aplicação destas técnicas de pré-processamento em um fragmento de texto.

Tabela 1 – Efeitos da aplicação das técnicas de pré processamento

Sentença original	“é pau, é pedra, é o fim do caminho...”
Tokenização com Eliminação de símbolos	“é” “pau” “é” “pedra” ”é” “o” “fim” “do” “caminho”
Tokenização sem Eliminação de símbolos	“é,” “pau,” “é,” “pedra,” ”é,” “o,” “fim,” “do,” “caminho...”

3.2.3 Redução da dimensionalidade do espaço de características

O espaço de características de um texto equivale à quantidade de palavras relevantes que ele contém. E existem alguns critérios importantes que podem ser estabelecidos a fim de definir a relevância destas palavras, servindo como um filtro. Tais filtros, quando aplicados, eliminam aquelas ditas irrelevantes para o processamento, reduzindo assim a quantidade de palavras que representam os documentos de texto. Neste trabalho, foram utilizadas as seguintes abordagens de filtragem de palavras: número mínimo de ocorrências e lista de stopwords.

3.2.3.1 Número mínimo de ocorrências de um termo em relação ao *corpus* (NMO)

A despeito do quão extensa pode ser uma coleção de documentos, há a possibilidade de que existam termos raros, isto é, aqueles que ocorrem pouquíssimas vezes quando comparados com os demais termos do CORPUS. Devido à quantidade de ocorrências ser muito baixa, estes termos são considerados irrelevantes. Logo, precisam ser removidos.

O critério de contagem de ocorrências consiste em estabelecer um limite mínimo de ocorrências de um termo em toda a base textual. Conta-se quantas vezes este termo específico aparece em todo o *corpus*. E caso este valor seja menor ou igual o limite estabelecido, o termo é removido, pois é considerado irrelevante. O valor do NMO varia de acordo com o tamanho do corpus, logo, é escolhido de forma empírica.

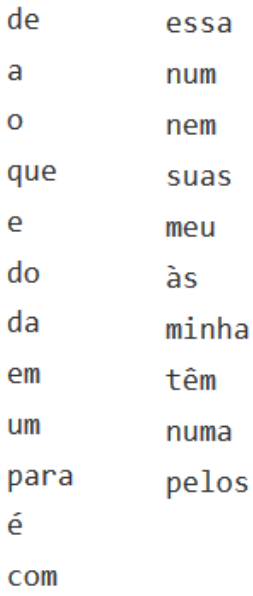
3.2.3.2 Lista de Stopwords

Stopwords são palavras que não apresentam um conteúdo semântico de significância no contexto em que se encontram no texto, ou seja, são palavras consideradas irrelevantes na análise do texto (MEDEIROS, 2004). Tratam-se de palavras auxiliares ou conectivas, não carregando informação que possa representar o conteúdo do texto a qual fazem parte.

Em uma lista de stopwords, também chamada de stoplist, são acrescentados artigos, preposições, pronomes e também palavras que apresentam alta incidência na coleção de textos. Como ocorrem em muitos documentos de forma bastante repetitiva, perdem relevância.

Para construir uma stoplist, é necessário levar em consideração o idioma dos textos do *corpus*. No caso da língua portuguesa, uma possível stoplist pode ser representada pela Figura 1.

Figura 1 – Exemplo de stopwords da língua portuguesa



FONTE: Próprio autor

É importante ressaltar que as principais características de um texto são as palavras que ele contém. Portanto, realizar o pré-processamento de textos é fundamental para que estes sejam bem representados numericamente, implicando em melhores resultados no processo de classificação.

Terminada esta fase, o próximo passo é criar uma representação numérica para os textos do corpus.

3.3 Representação de dados textuais

A etapa de pré-processamento, como visto anteriormente, objetiva preservar as características mais relevantes nos documentos por meio da eliminação de termos ruidosos e irrelevantes. Ao fim desta etapa, é preciso agrupar estas características em um conjunto chamado dicionário, ou vocabulário. Este conterá todas as palavras do *corpus* que foram preservadas no pré processamento e servirá como base para a representação dos dados textuais em estruturas compreensíveis pelas técnicas de classificação.

As representações de dados textuais utilizadas neste trabalho seguem o modelo atributo-valor. Segundo Martins (2003), este modelo, bastante usado no processo de mineração de textos, é uma estrutura adequada à maioria das tarefas de classificação

e agrupamento de documentos. Trata-se da conversão de textos em vetores numéricos n-dimensionais, em que cada dimensão representa um atributo característico de um texto em relação a toda a base de dados.

Destaca-se que uma boa de representação de documentos, em conjunto com as técnicas de pré-processamento, tem impacto fundamental no desempenho dos algoritmos de aprendizado, sejam supervisionados ou não supervisionados.

Para os experimentos deste trabalho, foram utilizadas duas abordagens de representação de texto: Bag-Of-Words e Vetores de parágrafo. A primeira é bastante popular e se caracteriza pela alta dimensionalidade por considerar todas as palavras do vocabulário do corpus como atributo. E a segunda, por sua vez, é mais complexa porque trata questões como semântica e ordem de palavras em um documento. Por fim, utiliza-se uma abordagem que une estas duas representações a fim de verificar o desempenho do algoritmo de classificação.

Explicar-se-á detalhadamente sobre as abordagens de representação textual utilizadas nas próximas subseções.

3.3.1 O modelo Bag-of-words

O modelo Bag-of-Words (BOW) (HARRIS, 1954) é bastante utilizado em trabalhos de classificação de texto. Nesse modelo, cada documento é representado por um vetor numérico de termos construído a partir do dicionário do *corpus* resultante do pré-processamento. Cada termo é associado a um valor, ou peso, que indica sua importância no documento. Os termos que não aparecem no documento em questão recebem valor zero.

Tomando como exemplo o seguinte *corpus* C = {d_1, d_2, d_3} no qual d_1 = {“o céu resplandece ao meu redor”}, d_2 = {“liberdade é correr pelo céu”}, d_3 = {“é você meu maior tesouro”}. O vocabulário, portanto, é V = {“o”, “céu”, “resplandece”, “ao”, “meu”, “redor”, “liberdade”, “é”, “correr”, “pelo”, “você”, “maior”, “tesouro”}

O vocabulário do *corpus* C é composto por 13 palavras, logo tem tamanho T_V = 13. Cria-se, então, um vetor padrão com T_V posições para o *corpus*, em que cada índice deste está associado a uma respectiva palavra do vocabulário. Esse vetor servirá como base para a representação vetorial dos documentos, de forma que cada documento seja representado por um vetor com tamanho fixo T_V e cada índice conterá um valor que represente aquele termo naquele documento. Caso o termo não ocorra no documento, seu valor será 0 (zero). Se ocorrer, outro valor é escolhido para representá-lo, podendo ser uma métrica de frequência ou simplesmente 1, no caso de uma representação binária (em que 1 indica presença e 0 ausência).

Sendo assim, os documentos e o vocabulário do corpus de exemplo C poderiam

ser representados da seguinte forma: V = [“o”, “céu”, “resplandece”, “ao”, “meu”, “redor”, “liberdade”, “é”, “correr”, “pelo”, “você”, “maior”, “tesouro”], d_1 = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0], d_2 = [0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0], d_3 = [0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1].

Neste trabalho, os pesos dos atributos são calculados de acordo com o modelo Term Frequency-Inverse Document Frequency (TF-IDF) (SALTON; BUCKLEY, 1988). Esta abordagem sugere que quanto maior for a frequência de um termo em um documento, mais representativo ele é para o conteúdo deste; e quanto mais documentos contiverem um termo, menos discriminante ele se torna. Então cada termo t em um documento d possui um peso W cujo cálculo é descrito nas equações 3.1 e 3.2.

$$W_{td} = tf_{td} * idf_t \tag{3.1}$$

$$idf_t = log\frac{N}{df} \tag{3.2}$$

Onde *tf* é a frequência de um termo t em um documento d, N é a quantidade de documentos na base de dados e *df* é o número de documentos em que o termo t ocorre.

3.3.2 Vetores de parágrafos pré-treinados

Talvez, o modelo BOW seja a mais comum forma de representação de texto em vetores de características de tamanho fixo. Isto se deve à sua simplicidade, eficiência e bons resultados de acurácia. Mas, também, possui desvantagens. Algumas delas são enumeradas por Le e Mikolov (2014)

A ordem das palavras é perdida devido a esta representação ser construída tomando como base o vetor padrão derivado das palavras do vocabulário do *corpus*. Outra desvantagem é que esta representação sofre de esparsidade de dados e alta dimensionalidade. O modelo BOW, portanto, tem muito pouco senso em relação à semântica das palavras.

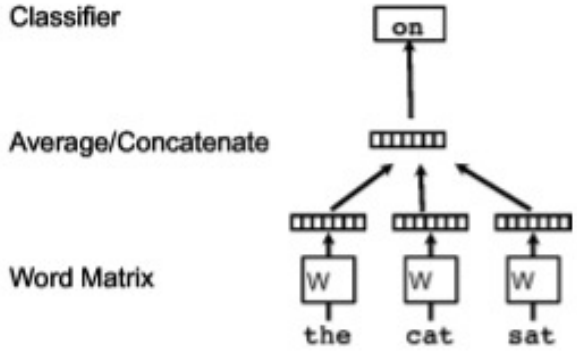
No trabalho de Le e Mikolov (2014), é proposta a abordagem Paragraph Vector (vetores de parágrafos). Este é um framework não supervisionado que aprende uma representação vetorial distribuída contínua para textos. Esta abordagem é capaz de construir representações a partir de sequências de entradas de tamanho variável, logo, é aplicável em frases, parágrafos e documentos.

A abordagem para aprender vetores de parágrafos é inspirada pelos métodos de aprendizagem de vetores de palavras apresentados por Mikolov et al. (2013).

3.3.3 Representação vetorial de palavras: Word2vec

O conceito de representação vetorial de palavras resume-se na tarefa de predizer uma palavra fornecida dadas as outras palavras em seu entorno dentro de um contexto.

Figura 2 – O contexto das palavras de entrada 'the', 'cat' e 'sat' é utilizado para predizer a palavra 'on'.



Fonte: (LE; MIKOLOV, 2014)

Cada palavra é mapeada para um único vetor representado por uma coluna em uma matriz W. A coluna é indexada pela posição da palavra no vocabulário. Cada termo na matriz W é representado por um vetor iniciado aleatoriamente. A concatenação ou soma destes vetores é usada como característica para prever uma próxima palavra na sentença (Figura 2).

Então, dada uma sequência de palavras para treinamento, o objetivo do modelo é maximizar:

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w_t|w_{t-k},...,w_{t+k}) \tag{3.3}$$

A predição é tipicamente feita por um classificador multiclasse softmax, em que:

$$p(w_t|w_{t-k},...,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{3.4}$$

Cada Yi é um log de probabilidade não normalizado para cada palavra de saída i, calculado por:

$$y = b + Uh(w_{t-k},...,w_{t+k};W) \tag{3.5}$$

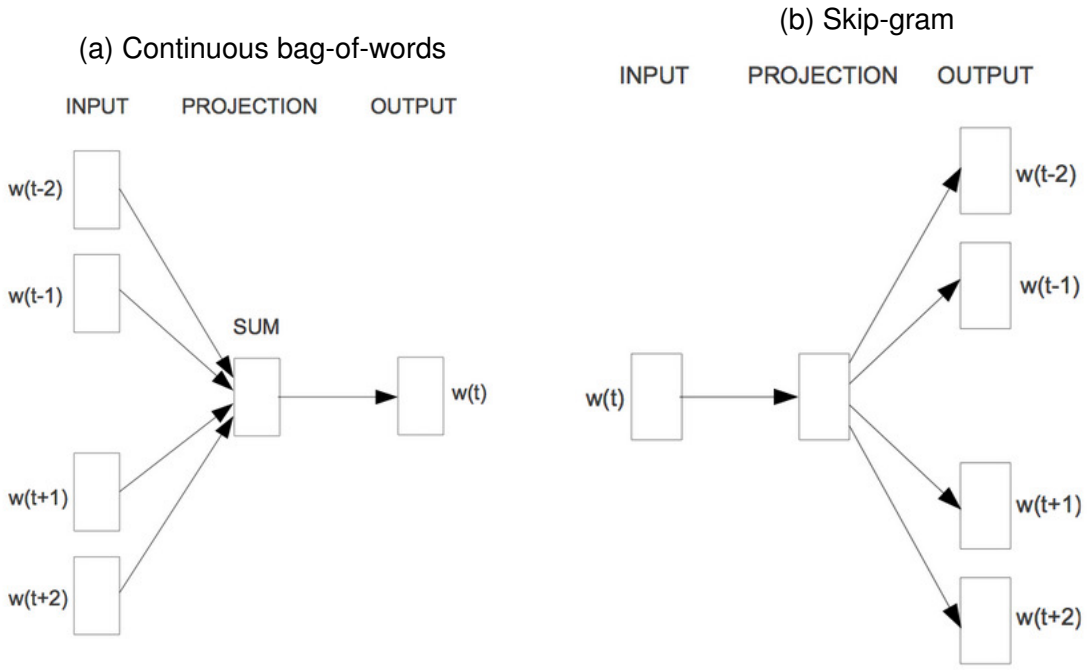
onde U e b são parâmetros do classificador softmax, h é construído a partir da concatenação ou média de vetores de palavras extraídos da matriz W.

A rede neural baseada em vetores de palavras é treinada utilizando o gradiente descendente estocástico (Stochastic gradient descent - SGD), sendo este obtido

via backpropagation. Este modelo é inspirado nos modelos neurais de linguagem apresentados por Bengio et al. (2006).

Há duas abordagens dentro do word2vec: Continuous Bag-of-Words (CBOW) e Skip-gram (SG). Em CBOW, a entrada do modelo é composta por n palavras que são combinadas por meio de soma de vetores para predizer uma outra palavra dentro daquele contexto (Figura 3a).

Figura 3 – Abordagens de representação de palavras



Fonte: (MIKOLOV et al., 2013)

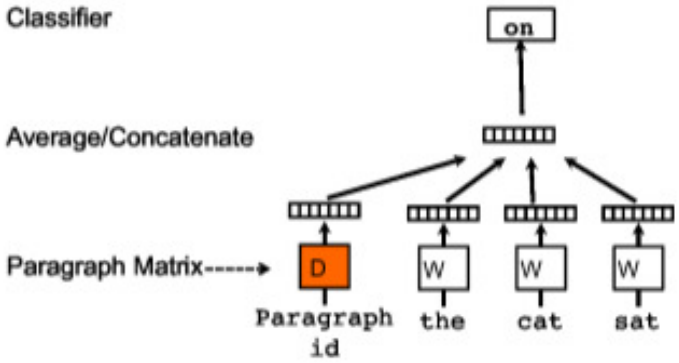
Já o Skip-gram (Figura 3b), ao invés de utilizar n palavras para predizer outra naquele contexto, segue o processo oposto. Uma palavra central é fornecida como entrada do modelo e servirá como característica para predizer n palavras em seu entorno. Neste modelo, a janela de palavras n é um hiperparâmetro.

3.3.4 Representação de parágrafos, sentenças e documentos: Doc2Vec

Doc2vec é uma extensão do Word2vec para o aprendizado de representações de documentos, sentenças e parágrafos. Possui duas abordagens para o processo de representação: Distributed Memory Paragraph Vectors (DM-PV) e Distributed Bag-of-words (DBOW) (LE; MIKOLOV, 2014).

DM-PV (Figura 4) funciona de forma similar ao CBOW. Para a entrada do modelo, além das palavras do texto, é adicionado um token que representa este documento. A principal diferença é que, em DM-PV, os vetores são concatenados e não somados como em CBOW. Logo, a predição de uma palavra dentro do contexto em que aquele

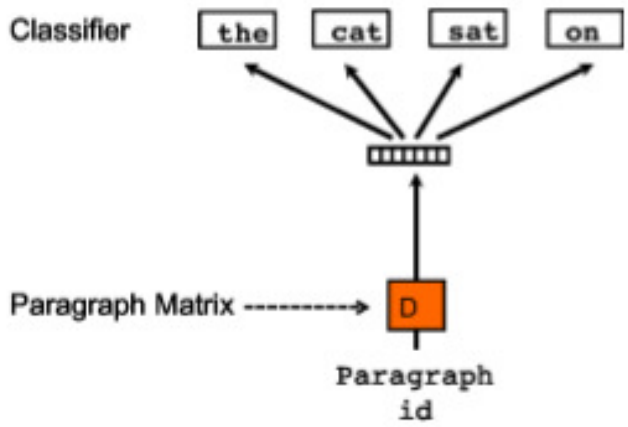
Figura 4 – Distributed Memory Paragraph Vectors



Fonte: (LE; MIKOLOV, 2014)

documento está inserido é dada pela concatenação entre o vetor de parágrafo e os vetores de palavras.

Figura 5 – Distributed Bag-of-words.



Fonte: (LE; MIKOLOV, 2014)

DBOW, por sua vez, assemelha-se à abordagem Skip-gram. Salvo que, em DBOW, a entrada se trata de um token que representa o documento. Este vetor é treinado para predizer as palavras que se encaixam no contexto em que está inserido (Figura 5).

Segundo Le e Mikolov (2014), apesar de DMPV alcançar bons resultados para a grande maioria das tarefas de classificação, DBOW é mais consistente, portanto, mais recomendada. Os experimentos realizados por Lau e Baldwin (2016) envolvendo as duas abordagens também reforçam as vantagens que DBOW possui em relação a DM-PV. Seguindo estas recomendações, para os experimentos realizados no presente trabalho, utiliza-se a abordagem DBOW.

3.4 Máquina de Vetores de Suporte (SVM)

Esta seção apresenta o método Máquina de Vetores de Suporte que é a técnica de aprendizado de máquina (AM) utilizada nos experimentos deste trabalho.

Máquina de Vetores de Suporte é um método de aprendizagem supervisionada introduzido por Cortes e Vapnik (1995) usado para estimar uma função que classifique dados em duas classes. O conceito básico por trás das SVMs compreende a construção de um hiperplano como superfície de decisão de forma que seja máxima a margem de separação entre as classes. O objetivo do treinamento por meio das SVMs é a obtenção de hiperplanos que dividam a amostra de tal maneira que sejam otimizados os limites de generalização (ALMEIDA et al., 2009).

As SVMs são sistemas de aprendizagem caracterizados pela utilização de um espaço de hipóteses de funções lineares em um espaço de características de alta dimensionalidade. São treinadas por algoritmos fortemente influenciados pela teoria da otimização e de aprendizagem estatística. SVMs vêm demonstrando superioridade frente a outros classificadores, sendo utilizadas em aplicações variadas (CRISTIANINI; SHAWE-TAYLOR, 2000).

Na metodologia das SVMs, uma variável de predição é chamada de atributo. E este atributo, quando aplicado na construção de hiperplanos, é chamado de característica. O conjunto de características selecionadas para descrever o objeto que se deseja classificar é chamado de vetor. Os vetores que se encontram mais próximos dos hiperplanos construídos para separar as classes são chamados de vetores de suporte.

As SVMs possuem diferentes *kernels* que são utilizados na resolução de problemas de espaços não-lineares. Os mais utilizados são os tipos Linear, Polinomial (que manipula uma função polinomial cujo grau pode ser definido durante os treinamentos), Sigmoidal (permite que a SVM se comporte de maneira similar à rede MLP[1]), e Gaussiano (a SVM se comporta como uma rede RBF[2]). As quatro funções básicas de *kernel* são:

- Linear: $K(x, y) = x^T y$
- Polinomial: $K(x, y) = (\gamma x^T y + r)^d, \gamma > 0$
- Sigmoidal: $K(x, y) = tanh(\gamma x^T y + r)$
- Função de base Radial (RBF): $K(x, y) = e^{-\gamma \|x-y\|^2}$

Definições diferentes do *kernel*, bem como dos demais parâmetros, causam alterações nos resultados fornecidos por uma SVM. Nos experimentos deste trabalho,

[1] Rede perceptron de múltiplas camadas (*Multi-layer Perceptron*)
[2] Funções de base radial (*Radial Basis Function*)

foi utilizado o *kernel* RBF, com parâmetros C (penalidade para classificações incorretas nos dados de treinamento) e γ positivos.

3.5 Validação de resultados

De acordo com Lorena (2006), a avaliação de um algoritmo de aprendizado de máquina supervisionado é normalmente realizada por meio da análise do desempenho do preditor gerado pelo mesmo na classificação de novos dados não apresentados previamente em seu treinamento.

No experimento de classificação de sucesso musical baseada em letras, a avaliação dos resultados fornecidos pelo classificador consiste em verificar o quão bem ele discrimina quando uma letra de uma canção corresponde ou não a um caso de sucesso. Assim, são consideradas quatro situações possíveis:

1. A música é um caso de sucesso e foi classificada corretamente - Verdadeiro Positivo
2. A música é um caso de sucesso, mas foi classificada como insucesso - Falso Negativo
3. A música é um caso de insucesso, mas foi classificada como sucesso - Falso Positivo
4. A música foi é um caso de insucesso e foi classificada como tal - Verdadeiro Negativo

Neste trabalho, são utilizadas a Especificidade (E), Sensibilidade (S) e Acurácia (A) por serem métodos estatísticos comumente utilizados para validar resultados em problemas de classificação.

A Sensibilidade é a probabilidade dos verdadeiros positivos, isto é, mede a capacidade do método de classificação de identificar corretamente as letras de músicas de sucesso dentre todo o conjunto de letras, segundo a equação:

$$S = \frac{VP}{VP + FN} \tag{3.6}$$

onde VP é o número de verdadeiros positivos, isto é, a quantidade de letras classificadas como sendo de músicas de sucesso. FN é o número de falsos negativos, ou seja, letras classificadas incorretamente como casos de insucesso.

A Especificidade consiste na probabilidade dos verdadeiros negativos, ou seja, avalia a eficácia do método de classificação em identificar corretamente as letras de

músicas que não são casos de sucesso. Pode ser melhor compreendida pela equação:

$$E = \frac{VN}{VN + FP} \tag{3.7}$$

onde VN corresponde aos verdadeiros negativos, portanto, letras de músicas de insucesso classificadas corretamente. FP é o número de falsos positivos (casos de insucesso classificados erroneamente).

A Acurácia é a probabilidade total dos acertos, dada pela equação:

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.8}$$

4 Materiais e Métodos

Este capítulo apresenta os procedimentos utilizados para a classificação de sucesso musical baseada em letras. Serão apresentadas as informações a respeito do hardware e softwares utilizados no desenvolvimento do método proposto pelo presente trabalho. Também serão explicadas cada etapa que constitui esta metodologia.

4.1 Software e Hardware utilizados

Os experimentos contaram com o suporte de ferramentas responsáveis por facilitar a aplicação dos métodos dentro da proposta deste trabalho. Estas ferramentas consistem em bibliotecas e aplicações implementadas nas linguagens de programação Java[1] e Python[2]. Estas linguagens orientadas a objetos são conhecidas por serem gratuitas e por oferecerem muitos recursos, entre eles, portabilidade, facilidade de programação, confiabilidade e documentação disponível.

Os procedimentos de criação do *corpus*, eliminação de símbolos especiais e manipulação de arquivos de texto foram implementados em linguagem Python. No experimento com a abordagem de representação de características textuais Doc2vec, foi utilizada a biblioteca Gensim (versão 0.13.4.1) (ŘEHŮŘEK; SOJKA, 2010), que consiste em um conjunto de *scripts* bastante robustos também escritos na linguagem Python, ideal para realizar modelagem semântica não supervisionada de textos.

Nos experimentos com os modelos Bag-of-words e híbrido, foi utilizada a biblioteca WEKA (versão 3.6) (HALL et al., 2009) para a geração de representações. Esta ferramenta, codificada em linguagem Java, também foi utilizada para a visualização dos resultados da classificação com o SVM. O classificador SVM foi obtido por meio da biblioteca LibSVM (versão 3.22) (CHANG CHIH-CHUNG ; LIN, 2011). Todas estas bibliotecas estão disponíveis gratuitamente na internet.

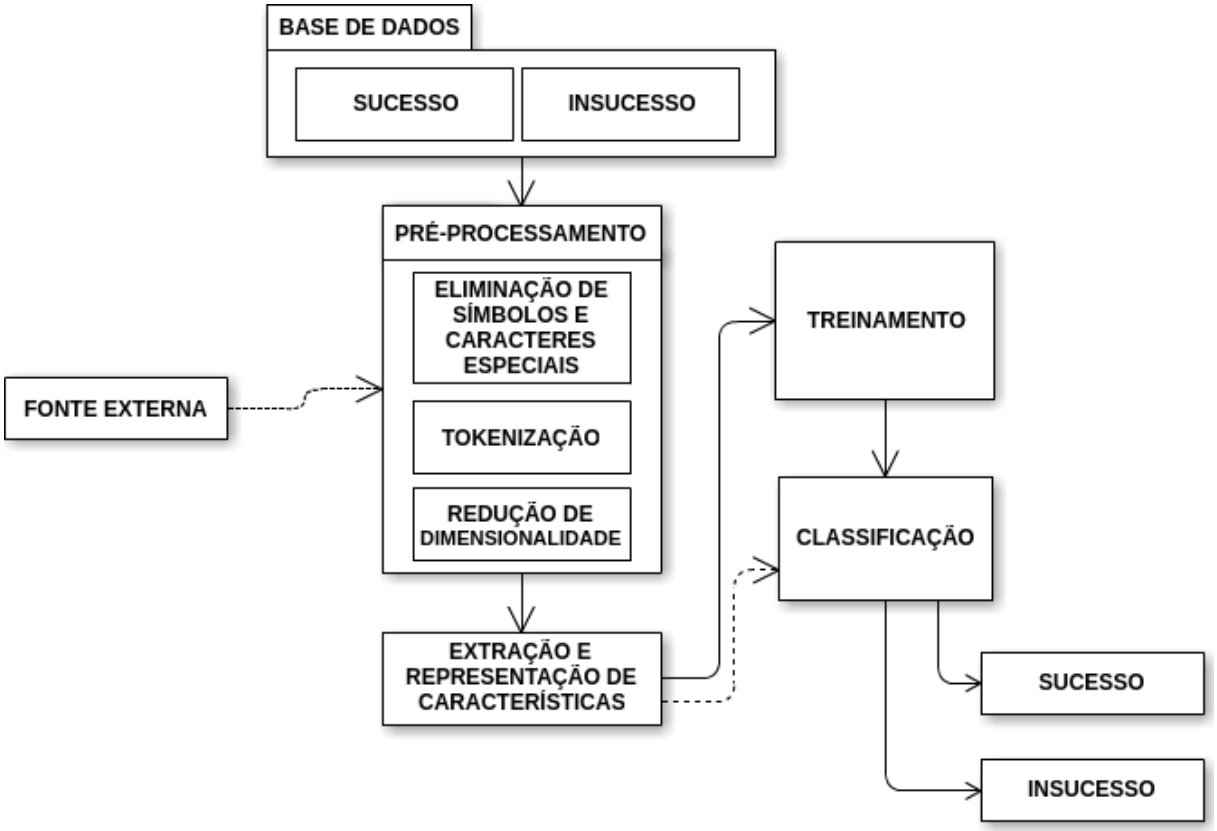
O computador utilizado para a implementação e testes da metodologia proposta conta com a seguinte configuração: processador Intel Core i5-3210M, 6GB de memória RAM e 1TB de HD.

[1] https://www.java.com
[2] https://www.python.org

4.2 Método para a classificação de sucesso musical baseada em letras

O método proposto para a classificação de sucesso musical baseada em letras pode ser visualizado através da Figura 6.

Figura 6 – O processo de classificação de textos aplicado à letras de música conforme a proposta deste trabalho



FONTE: Próprio autor

O primeiro passo consiste na coleta de letras musicais e criação da base de dados, chamada *corpus*. As letras foram previamente separadas em duas classes: sucesso e insucesso.

Em seguida, os documentos do *corpus* foram submetidos às técnicas de pré processamento explanadas na Seção 3.2 a fim de eliminar ruídos, isto é, caracteres e demais termos irrelevantes para representar as características de cada letra de música.

Subsequente ao pré processamento, ocorre a etapa de extração e representação de características dos documentos textuais, neste caso, das letras musicais. Nesta fase, as letras são representadas por vetores n-dimensionais em que cada elemento deste vetor corresponde a uma característica. O presente trabalho experimenta três

modelos para a realização desta tarefa: bag-of-words (BOW), representação vetorial de parágrafos (Doc2vec) e a representação híbrida (BOW + Doc2vec). As representações resultantes do processo anterior são submetidas ao classificador SVM para a fase de treinamento, e depois para a classificação.

Em aplicações reais, os documentos vêm de uma fonte externa. E neste caso, também passarão pelas etapas de pré processamento, extração e representação de características. Mas, após estas fases, serão submetidos diretamente à classificação.

As próximas seções discorrerão, mais precisamente, sobre as etapas supracitadas.

4.2.1 Construção do *Corpus*

Assim como nos trabalhos de Hu, Downie e Ehmann (2009), Fell e Sporleder (2014) e Mahedero et al. (2005), para o presente trabalho não foram encontradas bases de letras de música rotuladas para a tarefa proposta. No caso deste, em específico, não foi encontrada uma base constituída por letras em língua portuguesa. Então, foi necessário criar uma base de letras de música em português brasileiro através de pesquisas na web.

O site Vagalume[3] disponibiliza uma API[4] para que desenvolvedores possam utilizar o conteúdo do site em suas aplicações. Com atualização diária, este site é um dos maiores repositórios do Brasil com informações de artistas e letras de músicas. Para a construção do *corpus*, foi desenvolvido um crawler em linguagem Python[5] que consome esta API buscando pelas letras de música e extraindo-as.

As músicas rotuladas como casos de sucesso foram escolhidas tendo como base o top 100 músicas brasileiras eleitas pela revista Rolling Stone[6]. Das 100 músicas, entretanto, foram aproveitadas somente 99 devido uma delas ser apenas instrumental. Como este trabalho foca somente na classificação de letras, esta música em específico precisou ser retirada.

As músicas rotuladas como casos de insucesso foram escolhidas a partir do ranking do site Vagalume. São as músicas brasileiras menos visualizadas, elencadas por este site no período de junho de 2011 a dezembro de 2016 com saltos de 4 meses. Por exemplo, o programa consulta o ranking do mês junho de 2011. O próximo ranking a ser consultado é o de outubro de 2011, e assim por diante. As letras extraídas estavam nas 10 últimas posições de seus respectivos rankings. No total, para este conjunto, foram extraídas 180 letras de músicas. Entretanto, o conjunto insucesso é composto

[3] http://www.vagalume.com.br
[4] https://api.vagalume.com.br
[5] https://www.python.org/
[6] rollingstone.uol.com.br/listas/100-maiores-musicas-brasileiras

por 104 letras de música devido a retirada de algumas letras que ocorriam em mais de um ranking.

É importante ressaltar que tanto as letras rotuladas como sucesso quanto insucesso pertencem a vários estilos musicais. Portanto, a tarefa de classificação realizada neste trabalho visa um âmbito geral independente de estilo musical. Outro ponto é que todas as letras são de artistas brasileiros. Foram desconsideradas traduções e letras de músicas internacionais.

4.2.2 Pré processamento

Nesta fase dos experimentos, seguiram-se as técnicas explanadas na Seção 3.2. Mas, no caso específico de letras de música, outros cuidados precisam ser tomados, como apontam Hu, Downie e Ehmann (2009) e Fell e Sporleder (2014).

Algumas letras podem vir com palavras ou expressões com fins de indicação, como, por exemplo, as palavras "refrão", "verso", "interlúdio". E para indicar repetições, é comum encontrar expressões como 2X, 3X. Estas palavras e expressões foram considerados ruídos e precisaram ser removidas, pois, este trabalho enfoca no conteúdo da letra e não na estrutura da música a qual está atrelada.

Após este primeiro tratamento com os documentos do *corpus*, segue-se para os experimentos com as diferentes abordagens para representação de características.

4.2.3 Extração e representação de características

Esta seção discorre a aplicação das abordagens de extração e representação de características explanadas na Seção 3.3.

4.2.3.1 Bag-of-words (BOW)

O experimento com a abordagem Bag-of-words (HARRIS, 1954) foi desenvolvido utilizando a função StringToWordVector implementada na ferramenta WEKA (HALL et al., 2009). As letras foram convertidas do formato original, com versos e estrofes, para a forma de string em uma única linha. A função recebe estas strings e, a partir disso, realiza os procedimentos de tokenização, remoção de palavras com número mínimo de ocorrências e remoção de stopwords, estes últimos escolhidos pelo usuário. O NMO estabelecido foi igual a 5, pelo fato de a base de documentos ser pequena. As stopwords foram fornecidas ao programa em um documento[7] à parte. Por fim, a função monta o vocabulário do corpus e constrói os vetores de característica utilizando a abordagem TF-IDF (SALTON; BUCKLEY, 1988).

[7] Stoplist encontrada em https://gist.github.com/alopes/5358189

4.2.3.2 Doc2vec

O aplicação do modelo Doc2Vec foi possível com a utilização da biblioteca Gensim (ŘEHŮŘEK; SOJKA, 2010). Esta biblioteca contém as implementações dos algoritmos descritos no trabalho de Le e Mikolov (2014) por meio da função homônima Doc2vec.

Como dito anteriormente, para o presente trabalho foi escolhida a abordagem DBOW seguindo as recomendações de Le e Mikolov (2014) e Lau e Baldwin (2016). O tamanho do vetor de características foi fixado em 300. Esta escolha se deu de forma empírica baseada em testes realizados.

Os demais parâmetros para a função são o valor de NMO, que foi o mesmo utilizado no experimento com Bag-of-words (NMO = 5); e a janela de palavras a serem preditas pelo modelo, que neste caso, mantém-se o valor 5, padrão estabelecido pela ferramenta Gensim.

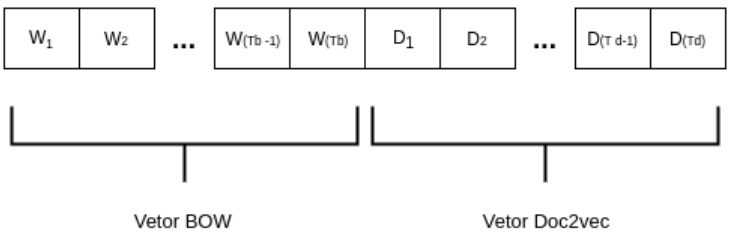
Para este experimento foi desenvolvido um programa em Python para ler os arquivos da base, convertê-los em strings e fornecê-los como entrada para a função que implementa o Doc2vec que irá gerar os vetores de parágrafos.

4.2.3.3 BOW + Doc2vec

Tanto o BOW quanto o Doc2Vec, à sua maneira, representam as características de um *corpus*. A primeira leva em consideração a relevância que as palavras possuem dentro de um contexto de busca, que pode ser visto como uma categoria ou classe. A segunda, por sua vez, captura aspectos semânticos dos textos. Sendo assim, a ideia de concatenar as duas representações surge para analisar se haverá ou não melhora nos resultados da classificação. O experimento de concatenar as duas abordagens é inspirado nos estudos de Lilleberg, Zhu e Zhang (2015) e Aguiar e Prati (2015).

Dados os vetores das representações BOW e Doc2Vec, o documento será representado pela concatenação de seus vetores de características, de forma análoga à Figura 7.

Figura 7 – Concatenação dos vetores de características BOW, com Tb dimensões, e Doc2vec, com Td dimensões



FONTE: Próprio autor

4.2.4 Treinamento e classificação com SVM

A etapa final consiste em categorizar as letras de música nas classes sucesso ou insucesso com o uso do método Máquina de Vetores de Suporte (SVM). Primeiro, as representações vetoriais dos documentos são submetidos ao classificador para o treinamento. Posteriormente, realiza-se a classificação.

Utiliza-se a biblioteca LibSVM (CHANG CHIH-CHUNG ; LIN, 2011) e WEKA (HALL et al., 2009), cuja finalidade é auxiliar os usuários a utilizar o SVM como ferramenta de uma maneira mais fácil. Nos experimentos desenvolvidos no presente trabalho, é utilizado o SVM com *kernel* RBF. Os parâmetros C e γ são requisitados quando se usa o *kernel* RBF, e precisam ser otimizados para que o SVM produza melhores resultados. Estes valores são inicialmente desconhecidos. A estimação dos parâmetros é efetuada por meio da técnica *grid search*, existente na biblioteca LibSVM, que consiste em uma busca exaustiva no espaço de parâmetros. Esta busca é realizada sobre os dados de treinamento.

São realizados testes com as representações BOW, Doc2vec e BOW + Doc2vec a fim de se fazer uma análise comparativa do desempenho do classificador para cada abordagem de representação de características. Nestes testes, é aplicada a validação cruzada, que é uma técnica que serve para avaliar a capacidade de generalização de um método (KOHAVI et al., 1995).

O tipo de validação cruzada utilizada nos experimentos é a Leave-one-out (LOO), recomendada para bases de dados pequenas devido o seu alto custo computacional. Esta técnica consiste na divisão da base de dados em k partições similar ao método *k-fold*. Enquanto neste último o valor de k é definido pelo usuário e está sujeito a variações, em LOO, entretanto, k é igual ao número de instâncias da base de dados. Logo, k-1 dados são utilizados para treinamento e o restante, para teste. Este método permite uma investigação mais ampla sobre o modelo.

5 Resultados e Discussão

Foram realizados três experimentos, cada um envolvendo uma das abordagens de representação de características, que serão analisadas de forma comparativa com base nos resultados alcançados pelo classificador SVM.

O primeiro conjunto de testes foi realizado com a base de dados dividida para cada uma das seguintes proporções de treino/teste: 80%/20%, 70%/30%, 60%/40% e 50%/50%. O segundo conjunto de testes foi realizado com a base de dados completa, seguindo o método de validação cruzada Leave-one-out (LOO). Este é recomendado nos casos em que a base de dados é pequena.

Cada teste consiste em cinco execuções do SVM. Dessa forma, é possível observar os valores obtidos de acurácia, sensibilidade e especificidade, por meio dos quais se pode fazer uma análise detalhada do desempenho dos modelos de representação de dados.

5.1 Experimento com o modelo Bag-of-words

Como explicado no Capítulo 4, o experimento com o modelo BOW foi desenvolvido com o auxílio da função StringToWordVector implementada na ferramenta WEKA. Cada termo do vocabulário é representado no vetor de características por um valor obtido pelo cálculo TF-IDF. O *corpus*, portanto, foi representado por 203 instâncias de vetores com 775 dimensões.

A Tabela 2 mostra os resultados dos testes com a base representada pelo modelo Bag-of-words. Os valores de acurácia, especificidade e sensibilidade são representados pelas letras A, E, S respectivamente. Observa-se que as médias das acurácias de cada teste mantiveram-se na faixa de 68% a 72,7%. A menor acurácia ocorreu no teste com a partição 50/50 e a maior, no teste com a base dividida em 70/30.

O melhor resultado para a abordagem BOW nos testes com a base particionada ocorreu naquele com a divisão 70/30. Além de, neste teste, o classificador ter obtido a acurácia média de 72,66%, observa-se que a diferença entre as médias de sensibilidade e especificidade é a menor em comparação com os demais testes, indicando que o classificador conseguiu categorizar corretamente grande parte das letras em suas respectivas classes. Em todos os testes com o modelo BOW, as médias de especificidade foram maiores que as de sensibilidade, logo, a classificação produziu melhores resultados em relação aos casos de insucesso.

No caso do teste com a base dividida em 80/20, por exemplo, observa-se que este produziu um dos menores resultados para as métricas de avaliação, a despeito de uma maior quantidade de dados para treino. Isso mostra que ter mais dados de treinamento não implica, necessariamente, em melhores resultados.

Tabela 2 – Resultados dos testes com a representação BOW

(a) Partição 80% treino e 20% teste

	80/20		
Execuções	A	E	S
1	68,29%	57,14%	65%
2	68,29%	66,66%	75%
3	73,17%	90,47%	55%
4	70,33%	76,19%	60%
5	60,97%	76,19%	60%
Média	68,21%	73,33%	63%

(b) Partição 70% treino e 30% teste

	70/30		
Execuções	A	E	S
1	75,8%	68,75%	76,66%
2	61,74%	78,12%	80%
3	74,19%	78,12%	70%
4	79,03%	78,12%	56,66%
5	72,58%	75%	76,66%
Média	72,66%	75,62%	71,99%

(c) Partição 60% treino e 40% teste

	60/40		
Execuções	A	E	S
1	67,07%	71,42%	67,5%
2	71,95%	85,71%	72,5%
3	67,07%	69,04%	65%
4	79,26%	69,04%	75%
5	69,51%	71,42%	62,5%
Média	70,97%	73,32%	68,5%

(d) Partição 50% treino e 50% teste

	50/50		
Execuções	A	E	S
1	61,76%	78,84%	57,99%
2	68,62%	78,84%	64%
3	69,6%	78,84%	60%
4	71,56%	73,04%	64%
5	68,62%	69,23%	54%
Média	68,03%	75,75%	59,99%

No teste com a validação cruzada LOO, os resultados para acurácia, especificidade e sensibilidade foram 72,77%, 70.87% e 74.74%, respectivamente. A peculiaridade deste teste é que a sensibilidade foi maior que a especificidade, diferindo-os dos outros testes com a base particionada. Em termos de acurácia, este teste supera os demais.

5.2 Experimento com o modelo Doc2vec

O *corpus*, neste experimento, é representado por 203 instâncias de vetores de documentos de 300 dimensões. Estes vetores foram obtidos por meio da aplicação do modelo Doc2vec sobre a base textual com o auxílio da biblioteca Gensim (ŘEHŮŘEK; SOJKA, 2010).

Os resultados obtidos com a utilização do modelo Doc2vec são vistos na Tabela 3. Esta abordagem, como explicado anteriormente, consegue captar características

semânticas e a ordem das palavras no texto, contrapondo-se às premissas do modelo BOW. Isto impacta em uma significativa melhora de acurácia (médias entre 70,5% - 75,9%), principalmente no teste em que a base foi dividida na proporção 70%/30%, alcançando 75,80%. Atenta-se, neste teste, a média das especificidades, a qual alcançou 82,49%. Já nos testes com a base particionada em 60% treino - 40% teste (Tabela 3c) e 50% treino - 50% teste (Tabela 3d), a média das sensibilidades superou a das especificidades.

Tabela 3 – Resultados dos testes com a representação Doc2vec

(a) Partição 80% treino e 20% teste

	80/20		
Execuções	A	E	S
1	80,48%	90,47%	65%
2	63,41%	57,14%	85%
3	73,17%	76,19%	70%
4	70,73%	57,14%	70%
5	73,17%	90,47%	70%
Média	72,19%	74,28%	72%

(b) Partição 70% treino e 30% teste

	70/30		
Execuções	A	E	S
1	74,19%	78,12%	73,33%
2	72,58%	75%	80%
3	77,41%	78,12%	66,66%
4	75,8%	87,5%	70%
5	79,03%	93,75%	53,33%
Média	75,80%	82,49%	68,66%

(c) Partição 60% treino e 40% teste

	60/40		
Execuções	A	E	S
1	79,26%	71,42%	87,5%
2	69,51%	76,19%	70%
3	73,17%	64,28%	82,5%
4	73,17%	66,66%	72,5%
5	79,26%	85,71%	72,5%
Média	74,87%	72,85%	77%

(d) Partição 50% treino e 50% teste

	50/50		
Execuções	A	E	S
1	79,41%	63,46%	74%
2	67,64%	63,46%	78%
3	69,6%	71,15%	68%
4	70,58%	73,07%	62%
5	65,62%	84,61%	74%
Média	70,57%	71,15%	71,2%

No teste com a validação cruzada LOO, os resultados obtidos foram 82,67% de acurácia, 78,64% de especificidade e 86,86% de sensibilidade. Este teste superou os demais.

5.3 Experimento com o modelo BOW + Doc2vec

Os testes com a abordagem híbrida (Bow + Doc2vec) apresentaram certa peculiaridade, como mostrado na Tabela 4. Neste conjunto de testes, este modelo alcançou os melhores resultados de acurácia comparado com Doc2vec e BOW sozinhos. Esta representação reúne como características a relevância das palavras do texto para classe

a qual pertence (BOW) e significados semânticos (Doc2vec). Mas, ainda possui alta dimensionalidade, embora seja menos esparsa, devido a densidade do vetor Doc2vec.

É importante ressaltar, porém, que à medida em que se equilibram os conjuntos de treino e teste, o classificador perde em performance. Analisando as médias na Tabela 4, vê-se que os resultados decaem. Mas, ainda sim, ao manterem-se na faixa de 74% a 78%, superam as outras abordagens, com destaque para o teste com a proporção 80%/20%, cuja acurácia média foi de 78,04%.

Tabela 4 – Resultados dos testes com a representação BOW + Doc2vec

(a) Partição 80% treino e 20% teste

	80/20		
Execuções	A	E	S
1	82,92%	66,66%	80%
2	75,6%	76,19%	75%
3	82,92%	90,47%	75%
4	75,6%	71,42%	80%
5	73,17%	90,47%	75%
Média	78,04%	79,04%	77%

(b) Partição 70% treino e 30% teste

	70/30		
Execuções	A	E	S
1	79,03%	78,12%	83,33%
2	69,35%	71,87%	83,33%
3	79,03%	75%	83,33%
4	77,41%	68,75%	70%
5	80,64%	81,75%	76,66%
Média	77,09%	75,09%	79,33%

(c) Partição 60% treino e 40% teste

	60/40		
Execuções	A	E	S
1	73,17%	76,19%	65%
2	74,39%	83,33%	72,5%
3	78,04%	73,8%	82,5%
4	78,04%	85,71	62,5%
5	70,73%	85,71%	60%
Média	74,87%	80,94%	68,5%

(d) Partição 50% treino e 50% teste

	50/50		
Execuções	A	E	S
1	74,5%	80,76%	60%
2	75,49%	84,61%	78%
3	71,56%	76,92%	66%
4	81,37%	78,84%	72%
5	70,58%	80,76%	68%
Média	74,7%	80,37%	68,8%

Os resultados de acurácia, especificidade e sensibilidade obtidos no teste com a validação LOO foram, respectivamente, 76,23%, 78,64%, 73,73%.

Somente no teste apresentado na Tabela 4b a média da sensibilidade superou a especificidade. Nos demais testes, ocorreu o contrário. Isto significa que, com esta abordagem, o classificador SVM desempenhou com mais eficácia a categorização das letras de insucesso.

5.4 Discussão

Com os experimentos realizados com cada abordagem de representação textual, objetiva-se fazer uma análise comparativa entre elas com base nos resultados produzidos pelo classificador SVM. Os testes consistiram em dois conjuntos. O primeiro, com a base particionada em grupamentos de treino e teste; e o segundo, feito com a base completa utilizando o método Leave-one-out (LOO). A Tabela 5 apresenta um resumo dos testes realizados, apresentando os valores de acurácia (Tabela 5a) e das medidas de especificidade e sensibilidade (Tabela 5b), destacando os melhores resultados.

Tabela 5 – Resumo dos resultados obtidos nos testes aplicados

(a) Média das acurácias

	BOW	Doc2vec	BOW+Doc2vec
LOO	72,77%	**82,67%**	76,23%
80/20	68,21%	72,19%	**78,04%**
70/30	72,66%	75,80%	**77,09%**
60/40	70,97%	74,87%	74,87%
50/50	68,03%	70,57%	74,7%

(b) Médias dos valores de Especificidade e Sensibilidade

	BOW		Doc2vec		BOW + Doc2vec	
Testes	E	S	E	S	E	S
LOO	70.87%	74.74%	78,64%	86,86%	78,64%	73,73%.
80/20	73,33%	63%	74,28%	72%	79,04%	77%
70/30	75,62%	71,99%	82,49%	68,66%	75,09%	79,33%
60/40	73,32%	68,5%	72,85%	77%	80,94%	68,5%
50/50	75,75%	59,99%	71,15%	71,2%	80,37%	68,8%

Em uma análise geral, nota-se que os resultados obtidos nos experimentos com a abordagem BOW foram os menos expressivos. Observa-se que estes testes apresentaram os menores valores de acurácia em comparação com aqueles realizados com as outras abordagens. No modelo BOW, os valores TF-IDF representam a relevância dos termos de um documento para a classe à qual ele pertence. Dessa forma, a classificação é fortemente influenciada pela informação que os termos concentram sozinhos quando presentes. No caso das letras musicais, analisar os documentos baseando-se somente na presença ou ausência de termos mostrou não ser a melhor metodologia, pois, a definição de sucesso ou insucesso transcende esta questão.

Os testes com o modelo Doc2vec apresentaram resultados mais expressivos que o modelo BOW, alcançando o melhor resultado (82,67%) no teste com a validação LOO (Tabela 5a). Mas, nos testes com a base de dados particionada, obteve médias de acurácia inferiores às dos experimentos com o modelo híbrido. A abordagem

Doc2vec capta a semântica das letras musicais. Como algumas músicas podem tratar do mesmo tema (amor, traição, diversão), logo, têm semântica parecida, isto acarreta em dificuldades para o classificador. Mas, ainda sob esta situação, os vetores de parágrafos conseguiram representar as características fundamentais de cada letra para que a classificação obtivesse ganhos de performance.

Os experimentos com a abordagem híbrida (BOW + Doc2vec) alcançaram resultados mais expressivos que os demais. Estes bons resultados corroboram a ideia de que trabalhar com metodologias de representação textual em conjunto pode acarretar em melhores resultados de classificação.

Ressalta-se que, analisando os valores médios de sensibilidade e especificidade, é possível notar que o classificador SVM conseguiu acertar em maior proporção os exemplos de insucesso nos testes com BOW + Doc2vec e os de sucesso nos experimentos com o modelo Doc2vec. Isto também pode ser visto, por exemplo, na Tabela 6 por meio das matrizes de confusão obtidas no teste com a base completa utilizando Leave-One-Out.

Tabela 6 – Matrizes de confusão

(a) BOW

	S	I
S	74	25
I	30	74

(b) Doc2vec

	S	I
S	86	13
I	22	82

(c) BOW + Doc2vec

	S	I
S	73	26
I	22	82

As classes sucesso e insucesso estão rotuladas, respectivamente, por S e I.

Com base nisso, foi realizada uma investigação a fim de descobrir quais letras foram classificadas corretamente e quais aquelas cuja classificação se deu de forma errada em ambos os experimentos. Por exemplo, as músicas "Chega de saudade" (Tom Jobim) e "Turbinada" (Zé Ricardo e Thiago) foram rotuladas, respectivamente, como sucesso e insucesso. Entretanto, tiveram sua classificação incorreta. Já a classificação das músicas "Anna Júlia" (Los Hermanos) e "De todos os loucos do mundo" (Clarisse Falcão), rotuladas como sucesso e insucesso respectivamente, se deu de forma correta.

Reitera-se que este trabalho analisa o sucesso musical apenas baseado na letra das canções e em um cenário geral da música brasileira, ou seja, não leva em consideração estilos musicais específicos, tampouco avalia a questão do sucesso em nível regional. Sendo assim, algumas das canções supracitadas podem ser muito conhecidas por determinado público adepto a aquele estilo musical. Contudo, isto não é objeto desta análise.

Vê-se por meio dos resultados alcançados que estes corroboram a aplicabilidade dos modelos que representam características semânticas de forma alternativa

à abordagem bag-of-words, que é amplamente utilizada. Doc2vec e o modelo híbrido BOW + Doc2vec obtiveram resultados superiores ao BOW isolado. BOW + Doc2vec mostra que a concatenação das representações de características pode garantir melhoramento em termos de acurácia. Porém, sua implementação tem maior complexidade em relação ao Doc2vec.

Dentro da proposta deste trabalho, pode ocorrer que algumas letras tenham semântica parecida ou palavras coincidentes, porém, no *corpus*, são rotuladas de forma diferente. Mas, ainda sob estas circunstâncias, o classificador conseguiu identificar padrões necessários para distinguir as letras e categorizá-las, em sua maioria, de forma correta. Portanto, os resultados obtidos são considerados satisfatórios.

6 Conclusão

Neste trabalho, foi proposto um método para a classificação de sucesso musical baseada em letra. O trabalho relata a aplicação de todas as fases do processo de classificação de texto para categorizar as letras de música em sucesso ou insucesso; explicando a difícil criação do *corpus* e realizando experimentos com as abordagens bag-of-words, Doc2Vec e a concatenação destas para a representação das características, submetendo-as ao classificador SVM para que fosse feita, também, uma análise comparativa entre estas representações.

Entende-se que o sucesso ou insucesso de uma música não é definido somente pelo conteúdo de sua letra, pois, é sabido que o ritmo, a produção e o marketing são grandes influenciadores para isso. Nestas condições, letras que tratam do mesmo tema, como amor, traição, podem estar rotuladas no *corpus* em classes diferentes. Mas, como dito anteriormente, a letra carrega a maior parte dos significados semânticos de uma música. Então, o cuidado com a escrita, as palavras e expressões utilizadas são considerados características determinantes para que o classificador possa capturar padrões e seja capaz de categorizá-las.

Foram experimentadas três abordagens para extração e representação de características: bag-of-words, Doc2vec e o modelo híbrido BOW + Doc2vec. A classificação das letras representadas pelo modelo bag-of-words obteve menores resultados de acurácia em comparação com as outras representações. Com o modelo híbrido de representação, a classificação alcançou os melhores resultados nos testes em que a base foi particionada em proporções treino/teste, porém, quanto aos experimentos com a base completa e validação cruzada leave-one-out, Doc2vec apresentou a melhor acurácia. Estes dois últimos são modelos de representação que utilizam características semânticas e mostraram-se melhor aplicáveis à tarefa proposta neste trabalho. Entretanto, o modelo híbrido tem implementação mais complexa que o Doc2vec, que conta com ferramentas capazes de facilitar o processo de representação.

O *corpus* criado para os experimentos deste trabalho é considerado pequeno, o que significa que os resultados podem melhorar com a ampliação desta base de dados. Para isto, mais pesquisas podem ser feitas em sites de crítica especializada e também pode haver um trabalho conjunto entre API's de sites com conteúdo musical a fim de descobrir as músicas com mais *likes* ou *dislikes* avaliadas pelo grande público. Esta ampliação pode ocorrer levando-se em consideração variados estilos musicais, assim como já ocorre neste trabalho, ou considerar algum estilo específico.

Baseado nos resultados satisfatórios obtidos com a classificação baseada em

letra, em trabalhos futuros há a possibilidade de o áudio ser considerado, podendo ter características extraídas com a utilização de técnicas como MPEG7 (MANJUNATH; SALEMBIER; SIKORA, 2002) e MFCC (MERMELSTEIN, 1976). A classificação envolvendo letra e áudio, hipoteticamente, pode alcançar resultados melhores e, assim, ser mais aplicável em um cenário real.

Referências

ABBURI, H.; AKKIREDDY, E. S. A.; GANGASHETTY, S. V.; MAMIDI, R. Multimodal sentiment analysis of telugu songs. In: *Proceedings of the 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2016)*. [S.l.: s.n.], 2016. p. 48–52. Citado na página 21.

ADOLFO, A. *Composiçao, Uma Discussão...* LUMIAR, 1997. ISBN 9788585426408. Disponível em: <https://books.google.com.br/books?id=1RAhFupGD_sC>. Citado na página 16.

AGUIAR, R. F.; PRATI, R. C. Incorporação de representação vetorial distribuída de palavras e parágrafos na classificação de sms spam. 2015. Citado 2 vezes nas páginas 20 e 40.

ALMEIDA, J. D.; SILVA, A. C.; PAIVA, A.; TEIXEIRA, J. A. Metodologia computacional para detecção e diagnóstico automáticos, e planejamento cirúrgico do estrabismo. In: *XXIX Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2009. Citado na página 33.

BENGIO, Y.; SCHWENK, H.; SENÉCAL, J.-S.; MORIN, F. Gauvain, jean-luc. neural probabilistic language models. *Innovations in Machine Learning*, p. 137–186, 2006. Citado na página 31.

CASTRO, G. G. Web music: produção e consumo de música na cibercultura. *Comunicação Mídia e Consumo*, v. 1, n. 2, p. 7–19, 2008. Citado na página 16.

CHANG CHIH-CHUNG ; LIN, C.-J. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 2, n. 3, p. 27, 2011. Citado 2 vezes nas páginas 36 e 41.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado 2 vezes nas páginas 16 e 33.

CRISTIANINI, N.; SHAWE-TAYLOR, J. *An introduction to support vector machines and other kernel-based learning methods*. [S.l.]: Cambridge university press, 2000. Citado na página 33.

DIAZ-GALIANO, M.; MONTEJO-RÁEZ, A. Participación de sinai dw2vec en tass 2015. *Comité organizador*, p. 59, 2015. Citado na página 20.

DOSCIATTI, M. M.; FERREIRA, E. C. L. P. C. Identificando emoções em textos em portugues do brasil usando maquina de vetores de suporte em soluçao multiclasse. 2013. Citado na página 19.

FELL, M.; SPORLEDER, C. Lyrics-based analysis and classification of music. In: *COLING*. [S.l.: s.n.], 2014. p. 620–631. Citado 3 vezes nas páginas 21, 38 e 39.

FRANCA, T.; OLIVEIRA, J. Análise de sentimento de tweets relacionados aos protestos que ocorreram no brasil entre junho e agosto de 2013. In: *III Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*. [S.l.: s.n.], 2014. Citado na página 20.

GASPERIN, C. V.; LIMA, V. L. S. Fundamentos do processamento estatístico da linguagem natural. *Trabalho Individual, PUC-RS*, 2000. Citado 2 vezes nas páginas 19 e 24.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>. Citado 4 vezes nas páginas 21, 36, 39 e 41.

HARRIS, Z. S. Distributional structure. *Word*, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954. Citado 3 vezes nas páginas 15, 28 e 39.

HU, X.; DOWNIE, J. S.; EHMANN, A. F. Lyric text mining in music mood classification. *American music*, v. 183, n. 5,049, p. 2–209, 2009. Citado 3 vezes nas páginas 21, 38 e 39.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 41.

LAU, J. H.; BALDWIN, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016. Citado 2 vezes nas páginas 32 e 40.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014. Disponível em: <http://arxiv.org/abs/1405.4053>. Citado 6 vezes nas páginas 15, 29, 30, 31, 32 e 40.

LILLEBERG, J.; ZHU, Y.; ZHANG, Y. Support vector machines and word2vec for text classification with semantic features. In: IEEE. *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*. [S.l.], 2015. p. 136–140. Citado 2 vezes nas páginas 20 e 40.

LORENA, A. C. *Investigação de estratégias para a geração de máquinas de vetores de suporte multiclasses*. Tese (Doutorado) — Universidade de São Paulo, 2006. Citado na página 34.

MAHEDERO, J. P.; MARTÍNEZ, Á.; CANO, P.; KOPPENBERGER, M.; GOUYON, F. Natural language processing of lyrics. In: ACM. *Proceedings of the 13th annual ACM international conference on Multimedia*. [S.l.], 2005. p. 475–478. Citado 2 vezes nas páginas 21 e 38.

MANJUNATH, B. S.; SALEMBIER, P.; SIKORA, T. *Introduction to MPEG-7: multimedia content description interface*. [S.l.]: John Wiley & Sons, 2002. v. 1. Citado na página 50.

MARTINS, C. A. Uma abordagem para pré-processamento de dados textuais em algoritmos de aprendizado. *USP–São Carlos*, 2003. Citado na página 27.

MEDEIROS, E. A. de. Técnica de aprendizagem de máquina para categorização de textos. 2004. Citado 3 vezes nas páginas 19, 23 e 26.

MERMELSTEIN, P. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, v. 116, p. 374–388, 1976. Citado na página 50.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado 2 vezes nas páginas 29 e 31.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010. p. 45–50. <http://is.muni.cz/publication/884893/en>. Citado 3 vezes nas páginas 36, 40 e 43.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information processing & management*, Elsevier, v. 24, n. 5, p. 513–523, 1988. Citado 2 vezes nas páginas 29 e 39.

SANTOS, J. C. A. dos; RIBEIRO, T.; FAVERO, E.; QUEIROZ, J. Aplicação de um método lsa na avaliação automática de respostas discursivas. In: *Anais do Workshop de Desafios da Computação Aplicada à Educação*. [S.l.: s.n.], 2012. p. 10–19. Citado na página 19.

SILVA, T. P.; SANTOS, I.; ALMEIDA, T. A.; HIDALGO, J. G. Normalização textual e indexação semântica aplicadas na filtragem de sms spam. *Proc. of the 11st ENIAC*, p. 1–6, 2014. Citado na página 20.

TROTTA, F. Música e mercado: a força das classificações. *Contemporanea-Revista de Comunicação e Cultura*, v. 3, n. 2, 2009. Citado na página 16.

WEISS, S. M.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. *Text mining: predictive methods for analyzing unstructured information*. [S.l.]: Springer Science & Business Media, 2010. Citado na página 23.

Anexos

ANEXO A – Lista de Stopwords

Durante a etapa de pré processamento do *corpus* formado pelas letras de músicas, realizou-se o procedimento de redução da dimensionalidade do espaço de características por meio da técnica de remoção de stopwords. A lista de stopwords (*stoplist*) utilizada neste trabalho possui palavras em português brasileiro e foi extraída de um repositório no site *GitHub*[1].

A *stoplist* contém as palavras: de, a, o, que, e, do, da, em, um, para, é, com, não, uma, os, no, se, na, por, mais, as, dos, como, mas, foi, ao, ele, das, tem, à, seu, sua, ou, ser, quando, muito, há, nos, já, está, eu, também, só, pelo, pela, até, isso, ela, entre, era, depois, sem, mesmo, aos, ter, seus, quem, nas, me, esse, eles, estão, você, tinha, foram, essa, num, nem, suas, meu, às, minha, têm, numa, pelos, elas, havia, seja, qual, será, nós, tenho, lhe, deles, essas, esses, pelas, este, fosse, dele, tu, te, vocês, vos, lhes, meus, minhas, teu, tua, teus, tuas, nosso, nossa, nossos, nossas, dela, delas, esta, estes, estas, aquele, aquela, aqueles, aquelas, isto, aquilo, estou, está, estamos, estão, estive, esteve, estivemos, estiveram, estava, estávamos, estavam, estivera, estivéramos, esteja, estejamos, estejam, estivesse, estivéssemos, estivessem, estiver, estivermos, estiverem, hei, há, havemos, hão, houve, houvemos, houveram, houvera, houvéramos, haja, hajamos, hajam, houvesse, houvéssemos, houvessem, houver, houvermos, houverem, houverei, houverá, houveremos, houverão, houveria, houveríamos, houveriam, sou, somos, são, era, éramos, eram, fui, foi, fomos, foram, fora, fôramos, seja, sejamos, sejam, fosse, fôssemos, fossem, for, formos, forem, serei, será, seremos, serão, seria, seríamos, seriam, tenho, tem, temos, tém, tinha, tínhamos, tinham, tive, teve, tivemos, tiveram, tivera, tivéramos, tenha, tenhamos, tenham, tivesse, tivéssemos, tivessem, tiver, tivermos, tiverem, terei, terá, teremos, terão, teria, teríamos, teriam.

[1] https://gist.github.com/alopes/5358189