UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

DEPARTAMENTO DE COMPUTAÇÃO - DC

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DOUGLAS HENRIQUE SANTANA DA SILVA

Um estudo comparativo de técnicas para a classificação contextual de companhia para sistemas de recomendação sensíveis a contexto

MONOGRAFIA

Recife

22 de Janeiro de 2019

DOUGLAS HENRIQUE SANTANA DA SILVA

Um estudo comparativo de técnicas para a classificação contextual de companhia para sistemas de recomendação sensíveis a contexto

Monografia apresentada ao curso de Bacharelado em Ciência da Computação, como parte dos requisitos necessários à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Douglas Véras e Silva

Recife

22 de Janeiro de 2019

Douglas Henrique Santana da Silva

Um estudo comparativo de técnicas para a classificação contextual de companhia para sistemas de recomendação sensíveis a contexto / Douglas Henrique Santana da Silva
. – Recife , 22 de Janeiro de 2019 -
57 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Douglas Véras e Silva

Monografia – UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE COMPUTAÇÃO - DC
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO , 22 de Janeiro de 2019 .

IMPORTANTE: ESSE É APENAS UM TEXTO DE EXEMPLO DE FICHA CATALOGRÁFICA. VOCÊ DEVERÁ SOLICITAR UMA FICHA CATALOGRÁFICA PARA SEU TRABALHO NA BILBIOTECA DA SUA INSTITUIÇÃO (OU DEPARTAMENTO).

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Douglas Henrique Santana da Silva às 14 horas do dia 22 de janeiro de 2019, no Auditório do CEAGRI-02 – Sala 07, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado **Um estudo comparativo de técnicas para a classificação contextual de companhia para sistemas de recomendação sensíveis a contexto**, orientado por Douglas Véras e Silva e aprovado pela seguinte banca examinadora:

Douglas Véras e Silva
DC/UFRPE

Rafael Ferreira Leite de Mello
DC/UFRPE

Dedico este trabalho aos meus pais, à minha irmã e minhas avós.

Agradecimentos

Agradeço primeiramente aos meus pais, Carlos e Wilma, por todo carinho, amor e apoio incondicional, e por terem feito o possível e o impossível para que me proporcionasse uma educação de qualidade.

À minha irmã Mariana e novamente à minha mãe, por terem me dado uma ajuda substancial durante o processo de experimentação.

Ao meu orientador, Douglas Véras e Silva, por toda orientação e suporte.

Ao meu amigo Vitor Rolim, por ter me ajudado e dado dicas importantes durante o desenvolvimento desta monografia, e ao meu amigo Ricardo Dantas por ter dado um suporte na etapa final do trabalho.

Ao corpo docente do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco, por terem contribuído diretamente na minha formação acadêmica.

Por último, porém não menos importante, aos meus colegas de turma, pelo companheirismo.

Resumo

Atualmente, a grande quantidade de informação tem prejudicado os usuários durante a tomada de decisões. Em face deste problema, sistemas de recomendação tem sido propostos de modo a conferir sugestões que auxiliem aos usuários em face de tal problema. Essas sugestões são ainda mais valiosas quando esses sistemas passam a sugerir itens se baseando também nos contextos ao qual o usuário está inserido. Dentre os esses contextos o de companhia pode ser destacado. Por meio da inferência do contexto de companhia o sistema poderá sugerir diferentes itens caso o usuário esteja acompanhado ou não. Um bom exemplo de sistema que possui tais características é o Sistemas de Recomendação em Domínios Cruzados e Sensíveis a Contexto (CD-CARS). Entretanto, o método de aprendizagem não supervisionada para inferência contextual de companhia no CD-CARS possui limitações. Desta forma, a presente pesquisa analisou e destacou um método de aprendizagem supervisionada que substitui a atual abordagem de classificação contextual de companhia executada no CD-CARS.

Palavras-chave: sistema de recomendação sensível a contexto, classificação de contexto, recomendação, classificação contextual de companhia.

Abstract

Nowadays, the vast amount of information has harmed users during decision making. In face of this problem, recommendation systems have been proposed in order to offer suggestions that help users to overcome such problem. These suggestions are even more valuable when these systems begin to suggest items based on the user contexts. Among these contexts, the companion context can be highlighted. Through the inference of the companion context the system may suggest different items if the user is accompanied or not. An example of a system that has such features is the CD-CARS. However, the unsupervised learning method for companion inference on CD-CARS has some limitations. In this way, the present research analyzed and highlighted a supervised learning method that can replace the current company contextual classification approach executed in the CD-CARS.

Keywords: context-aware recommendation system, context classification, recommendation, companion context classification.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

2D	Duas dimensões
CD-CARS	CROSS-DOMAIN CONTEXT-AWARE RECOMMENDER SYSTEM
FC	Filtragem Colaborativa
IDF	Inverse Document Frequency
LTS	Long-term Suport
NB	Naïve Bayes
Pos tagging	Part-of-speech tagging
SR	Sistemas de Recomendação
SRBC	Sistemas de Recomendação Baseado em Conteúdo
SRDC	Sistemas de Recomendação em Domínios Cruzados
SRFC	Sistemas de Recomendação de Filtragem Colaborativa
SRSC	Sistemas de Recomendação Sensíveis a Contexto
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TFT	Term Frequency Thresholding

Sumário

1 Introdução

O advento da Internet contribuiu substancialmente na produção de informação digital, sobrecarregando usuários com uma enorme quantidade de conhecimento, dificultando assim o acesso a conteúdo de uma forma direta. Diante deste cenário, alguns sistemas de busca de informação, popularmente conhecidos como buscadores, tais como: Google[1], Bing[2] e Yahoo[3] foram desenvolvidos de modo a facilitar o processo de pesquisa e obtenção de informações. Entretanto, o problema não foi solucionado por completo, pois ainda assim o conteúdo, que é retornado nas pesquisas, é muito extenso, impossibilitando o usuário de escolher o conteúdo de seu interesse dentre uma variedade de opções (VERBERT et al., 2012; BERGAMASCHI; GUERRA; LEIBA, 2010; YANG et al., 2016).

Assim sendo, algumas pesquisas sobre Sistemas de Recomendação (SR) foram desenvolvidas nos últimos anos (BORRÀS; ANTÔNIO MORENO; VALLS, 2014) (SASSI; MELLOULI; YAHIA, 2017), e este tipo de sistemas tem-se provado uma alternativa bastante eficiente ao lidar com tal problema de sobrecarga informacional (RICCI; ROKACH; SHAPIRA, 2015). Atualmente, esses SR têm desempenhado importantes papéis na sugestão de conteúdos em diversos *websites*, tais como: Youtube[4] (DAVIDSON et al., 2010), Amazon[5] (SMITH; LINDEN, 2017), Netflix[6] (AMATRIAIN; BASILICO, 2015) entre outros.

De acordo com Adomavicius e Tuzhilin (2015), muitos SR são Sistemas de Recomendação Baseado em Conteúdo (SRBC), ou seja, focam apenas no conteúdo, sem considerar nenhum tipo de informação contextual. Entretanto, em algumas ocasiões é necessário a incorporação de tal contexto ao processo de recomendação, de modo a fornecer melhores sugestões de itens a usuários sob determinados contextos. Tais sistemas, que consideram o contexto durante a recomendação, são determinados como Sistemas de Recomendação Sensíveis a Contexto (SRSC) (ADOMAVICIUS; TUZHILIN, 2015).

Adomavicius e Tuzhilin (2015) considera o contexto como dimensões (ex.: localização, tempo, humor, etc.) e seus atributos (ex.: país, cidade, ano, dia, etc.), que podem ser usados para adaptar as recomendações fornecidas.

Um bom exemplo de tais sistemas sensíveis a contexto, são aqueles de recomendação de músicas, que beneficiam-se de informações, tais como: tempo, humor, atividade atual e a presença de outras pessoas, para recomendar músicas a seus usuários (SCHEDL et al., 2015).

Adomavicius e Tuzhilin (2015) informa que essas informações contextuais podem ser

1 https://www.google.com/
2 https://www.bing.com/
3 https://br.yahoo.com/
4 https://www.youtube.com/
5 https://www.amazon.com.br/
6 https://www.netflix.com

obtidas de três formas:

- **Explicitamente**: Onde é solicitado que o usuário declare explicitamente as condições contextuais à medida que ele interage com os itens do sistema de recomendação (LEE; KWON, 2014; COLOMBO-MENDOZA et al., 2015) ;
- **Implicitamente:** Nesse caso os usuários não estão cientes do processo de coleta de informações contextuais pelo SRSC. Esta informação pode ser obtida de várias maneiras, um exemplo disto, são informações temporais que podem ser implicitamente obtidas através do momento em que uma avaliação é feita, ou seja, por meio de uma informação indireta (OH et al., 2014) ;
- **Inferindo:** Nesse método, as informações contextuais também são obtidas implicitamente, mas para serem obtidas é necessário o uso de algoritmos de aprendizagem de máquina, pois o contexto não pode ser obtido de maneira direta. Como por exemplo inferindo-se, por mineração de texto, o contexto de companhia em avaliação de usuários a itens (ex.: livros, músicas, hotéis, etc.) (LAHLOU et al., 2013; CAMPOS; RODRÍGUEZ-ARTIGOT; CANTADOR, 2017; LI et al., 2010).

A inferência contextual é importante, pois em determinadas ocasiões não conseguimos obter um determinado contexto de foma direta, tendo que utilizar algoritmos de classificação textual para extrair o contexto, como por exemplo, de companhia (ex.: sozinho, amigos, família, etc.).

1.1 Justificativa

Nos últimos anos, alguns trabalhos propondo diferentes métodos para inferência de informação contextual foram desenvolvidos com a finalidade de melhorar recomendações (LAHLOU et al., 2013; LI et al., 2010; BAUMAN; TUZHILIN, 2014; CAMPOS; RODRÍGUEZ-ARTIGOT; CANTADOR, 2017). Alguns destes trabalhos, além de inferir contextos tais como: localização e tempo, incluíram em seus esforços a inferência contextual de companhia, por meio de técnicas de mineração de texto e obtiveram bons resultados na inferência desse contexto.

Dentre as pesquisas que lidaram com o contexto de companhia pode-se destacar a pesquisa conduzida por Lahlou et al. (2013), onde consegue-se identificar um método para classificação de companhia em comentários extraídos do Tripadvisor [7]. Em seguida, tem-se a pesquisa realizada por Li et al. (2010), onde foi construído um classificador híbrido, para classificação de companhia. Por último, destaca-se a metodologia proposta por Campos,

[7] https://www.tripadvisor.com/

Rodríguez-Artigot e Cantador (2017), para classificação de contextos, que atribui classe de companhia pela taxonomia das palavras nos comentários.

Na tese de doutorado realizada por Silva (2016), onde é proposto um sistema de recomendação em domínios cruzados e sensível a contexto (CD-CARS), que considera a dimensão contextual de companhia em suas recomendações. O autor utiliza um método para inferência contextual de companhia, baseando-se no trabalho realizado Bauman e Tuzhilin (2014). Entretanto, a metodologia adaptada não produziu resultados tão satisfatórios quanto aos dos trabalhos mencionados anteriormente.

Diante dessas circunstâncias, surge uma oportunidade de analisar/comparar metodologias, para inferência contextual de companhia, de modo a destacar uma que fornecerá melhor classificação contextual, para ser empregadas em sistemas de recomendação sensíveis a contexto tais como o proposto por Silva (2016).

Dentre as metodologias bem sucedidas, tem-se a possibilidade de replicar a metodologia proposta por Lahlou et al. (2013) e compara-la a metodologia utilizada por Silva (2016), de modo a determinar se o método proposto por Lahlou et al. (2013) pode ser empregado ao CD-CARS, para melhorar a classificação contextual de companhia.

A proposta por Li et al. (2010) não será replicada e analisada, por está não fornecer maiores detalhamentos sobre a metodologia aplicada durante o processo de classificação contextual de companhia.

Durante o processo de análise das metodologias para classificação contextual de companhia em (LAHLOU et al., 2013; SILVA, 2016), usaremos uma base de dados contendo comentários da Amazon, referentes a itens, tais como: música, filme e livros.

1.2 Objetivo Geral

Este trabalho tem como objetivo realizar um estudo comparativo entre diferentes metodologias para classificação contextual de companhia, de forma a determinar aquela que produz melhores resultados, para que possa ser empregada em sistemas de recomendação sensíveis a contextos.

1.3 Objetivos Específicos

Dos objetivos específicos deste trabalho, destaca-se os seguintes:

- Disponibilizar uma base de dados de comentários classificados por contexto de companhia[8];

[8] Base de dados: https://github.com/douglashss/reviews_amazon_dataset/

- Disponibilizar o código desenvolvido para análise das metodologias de inferência contextual;
- Utilizar algoritmo para otimização de configuração de hiper parâmetros em modelos classificadores.

1.4 Estrutura do trabalho

Este trabalho está organizado da seguinte forma:

- No Capítulo 1, é realizada a contextualização sobre o trabalho, fornecendo: a justificativa e os objetivos do trabalho;
- No Capítulo 2, é descrita a fundamentação teórica, expondo os conceitos básicos e relevantes ao tema aqui desenvolvido;
- No Capítulo 3, é apresentado os trabalhos relacionados à pesquisa;
- No Capítulo 4, é descrita a metodologia empregada no trabalho;
- No Capítulo 5, é analisado os resultados obtidos ao final da execução do experimento;
- No Capítulo 6, é desenvolvida a conclusão, incluindo: considerações finais, linha de trabalhos futuros e dificuldades da pesquisa.

2 Fundamentação Teórica

Este capítulo explanará os tópicos abordados na pesquisa, de modo que se forneça uma melhor assimilação desses assuntos.

2.1 Sistemas de Recomendação (SR)

SR são ferramentas e técnicas de *'software'* que fornecem aos usuários sugestões de itens, os quais os usuários possam querer consumir. Essas recomendações ajudam usuários na tomada de decisões, tais como: qual filme assistir, qual álbum escutar ou qual livro ler. Como pode-se ver essas ferramentas são bastante uteis aos usuários, por minimizar o problema de sobrecarga de informação (RICCI; ROKACH; SHAPIRA, 2015).

Vários tipos de SR foram propostos, tais como:

- Sistemas de Recomendação de Filtragem Colaborativa (SRFC);
- Sistemas de Recomendação Baseado em Conteúdo (SRBC);
- Sistemas de Recomendação Sensíveis a Contexto (SRSC).

Os SRFC são sistemas que baseia suas previsões e recomendações nas classificações ou no comportamento de outros usuários do sistema (ELAHI; RICCI; RUBENS, 2016). A suposição por trás desses sistemas é que as opiniões de outros usuários podem ser selecionadas e agregadas de modo a fornecer uma previsão razoável para um usuário. Esses sistemas assumem que, se os usuários concordarem com a qualidade ou relevância de alguns itens, provavelmente concordarão sobre outros, ou seja, usuários que avaliaram os mesmos itens com avaliações semelhantes provavelmente terão preferências semelhantes(YANG et al., 2016).

Em alguns casos, as classificações de outros usuários podem não ser obrigatórias para fazer recomendações significativas. Nesses casos, as avaliações e ações dele em outros itens similares já são suficientes para descobrir recomendações significativas (GEMMIS et al., 2015). Ao contrário dos SRFC, que consideram as classificações de outros usuários além do usuário-alvo, os sistemas SRBC concentram-se principalmente nas avaliações do próprio usuário-alvo e nos atributos dos itens apreciados pelo mesmo, por exemplo, se ele classifica positivamente um filme de comédia, então o sistema o recomendará outros filmes do mesmo gênero (AGGARWAL, 2016).

Diferentemente dos SRBC e SRFC, que basicamente lidam com usuários e itens durante a sugestão, os SRSC também consideram o contexto dos usuários. Por exemplo, o tipo de um filme recomendado a uma pessoa pode ser diferente dependendo se esta planeja assisti-lo em um

sábado à noite acompanhado dos amigos ou durante a semana acompanhado dos pais. Nesse cenário, pode-se considerar que a companhia é o contexto a ser considerado na recomendação, entretanto outras informações contextuais também podem ser exploradas, tais como informações de: tempo e localização (ADOMAVICIUS; TUZHILIN, 2015). A diversidade de informações contextuais abrem um espaço para mineração de preferências de usuários inseridos em diversos contextos, possibilitando o desenvolvimento de recomendações cada vez mais personalizadas (ZHU et al., 2014).

2.2 Sistemas de Recomendação em Domínios Cruzados (SRDC)

A maioria dos sistemas de recomendação oferecem apenas recomendações em um único domínio, por exemplo, o Youtube[1] (DAVIDSON et al., 2010), que recomenda vídeos a partir de vídeos. Esses sistemas de recomendação tem sido implementados com sucesso por inúmeros websites. Entretanto, pode ser benéfico aproveitar informações de usuários disponíveis em vários domínios, para gerar modelos de usuários mais abrangentes e melhores recomendações. Ou seja, em vez de tratar cada domínio de forma independente, o conhecimento adquirido em um domínio de origem pode ser transferido e explorado por outro domínio de destino. Portanto, em um SRDC uma sugestão de um livro poderá ser derivada de um filme que o usuário avaliou positivamente (CANTADOR et al., 2015).

Para sistemas de recomendação de domínio único, fornecer sugestões relevantes de itens para novos usuários é um problema e frequentemente são necessários dados adicionais para compensar a falta de informações sobre as preferências dos usuários. Portanto, para compensar essa falta de informação, SRDC utilizam informações adicionais de domínios de origem diferentes. Essas informações auxiliares podem ser exploradas para mitigar a falta de dados históricos no domínio de recomendação de destino (FERNÁNDEZ-TOBÍAS et al., 2016).

Segundo Fernández-Tobías et al. (2012), boa parte das abordagens, para recomendação de domínios cruzados, utilizam filtragem colaborativa (FC), onde exploram as preferências do usuário (normalmente expressas como classificações explícitas para itens) e ignoram qualquer descrição baseada no conteúdo dos itens. Ele ainda afirma que a filtragem colaborativa em SRDC é mais conveniente devido à falta de homogeneidade do conteúdo dos itens de domínios diferentes.

2.3 Sistemas de Recomendação em Domínios Cruzados e Sensíveis a Contexto (CD-CARS)

Como mencionado anteriormente, a maioria das abordagens propostas para recomendação em domínio cruzado lidam com FC. Entretanto, apesar de grande parte dos

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

SRDC serem baseados em FC, nota-se que as técnicas de reconhecimento contextual ainda é pouco explorada, ou seja, a maioria dos SRDC sugerem itens, independentemente das condições contextuais dos usuários (FERNÁNDEZ-TOBÍAS et al., 2012; CANTADOR et al., 2015), olhando apenas para notas dos itens, que geralmente são representadas por tensores $\text{Nota} = (\text{ Usuário x Item })$.

Dessa forma, Silva (2016) abordou o problema de recomendação em domínios cruzados utilizando abordagens de filtragem colaborativa em conjunto com sensibilidade a contexto. Para tanto, o autor considerou as notas dos usuários como um tensor de três dimensões $Nota = (Usuário\ x\ Item\ x\ Contexto)$.

Silva (2016) define que os algoritmos CD-CARS são baseados em três paradigmas distintos e sistemáticos de recomendação sensível ao contexto:

- ***Pré-Filtragem* Contextual**: onde informações sobre o contexto atual são usadas para selecionar o conjunto relevante de avaliações de usuários. Em seguida o algoritmo de filtragem colaborativa em domínio cruzado é aplicado a essas matrizes para produzir as classificações;
- ***Pós-Filtragem Contextual***: onde inicialmente informações contextuais são ignoradas e as recomendações são selecionadas usando o algoritmo de filtragem colaborativa em domínio cruzado em todos os dados. Em seguida, as recomendações são ajustadas de acordo com as informações contextuais do usuário;
- ***Modelagem* Contextual**: onde as informações contextuais são usadas diretamente no processo de recomendação, não precisando aplicar o algoritmo de filtragem colaborativa em domínio cruzado.

A vantagem desta abordagem é justamente a melhoria da precisão das recomendações de domínio cruzado por incorporação de informação contextual (SILVA, 2016). Para tal, o CD-CARS considera, em sua recomendação de itens, três tipos de dimensões contextuais, são eles:

- **Localização** - Representada por informações geográficas do usuário, tais como: endereço, país, estado e cidade.
- **Temporal** - Representa quando um usuário consumiu um item, por exemplo, o período do dia (ex.: manhã, tarde ou noioite) ou tipo de dia (ex.: fim de semana ou dia da semana);
- **Companhia** - Representado pelo tipo de companhia (ex.: Acompanhado, Família, Amigos, Casal, Colega ou Sozinho) do usuário durante o consumo de um item.

2.4 Mineração de Texto

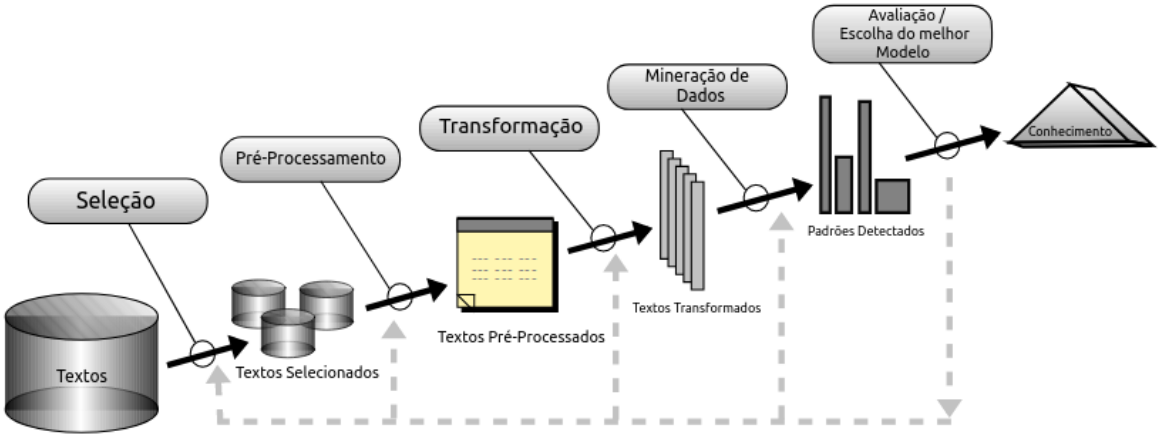
A mineração de texto pode ser definida como um conjunto de técnicas tais como: algoritmos e métodos do campo de aprendizagem de máquina e estatística, para obtenção de informações a partir de textos não estruturados. Basicamente ela pode ser compreendida como um processo que visa descobrir informações em grande quantidade de texto, por meio da identificação de padrões (RISTOSKI; PAULHEIM, 2016; HOTHO; NURNBERGER; PAASS, 2005) .

Conforme pode ser observado em (MAIMON; ROKACH, 2010; RISTOSKI; PAULHEIM, 2016), o processo de mineração de textos é dividido em cinco etapas, são elas:

- **Seleção dos textos**: É a etapa onde acontecerá a coleta da base de dados, bem como todos os dados necessários para a aplicação;
- **Pré-processamento**: Na etapa de pré-processamento acontecerá a preparação dos dados. É nesta etapa que acontecerá a limpeza de dados, separação dos termos e a remoção de ruído ou valores discrepantes. Ou seja, nesta etapa os dados serão normalizados de forma a permitir uma análise subsequente dos mesmos;
- **Transformação dos dados**: Nesta etapa é produzida uma projeção dos textos pré-processados em uma forma na qual os algoritmos de mineração de dados possam trabalhar. Isso significa transformar os dados em uma forma em que cada instância é representada por um vetor de características. Os métodos de redução de dimensionalidade também podem ser aplicados nesta etapa para reduzir o número de características textuais;
- **Mineração dos Dados**: Também conhecida como etapa de classificação, é nela onde serão aplicados métodos de predição ou descrição, para a procura de padrões;
- **Avaliação do modelo:** É a fase onde o modelo definido anteriormente serão avaliados. Nessa etapa acontecerá a validação do modelo em um conjunto de dados inédito, bem como interpretação das métricas de avaliação obtidas no processo de validação.

A Figura 1 exemplifica cada uma das etapas durante o processo de mineração de textos.

Figura 1 – Etapas do processo de Mineração de Texto



Fonte: (Petar Ristoski;Heiko Paulheim (2016))

2.4.1 Pré-processamento

Como mencionado anteriormente, o pré-processamento será a etapa onde serão aplicados métodos de limpeza, seleção e redução de volume dos dados, de forma que os prepare para que possam ser melhor utilizados em análises subsequentes realizadas por algoritmos de mineração de dados (RISTOSKI; PAULHEIM, 2016).

Na ocasião, a pesquisa beneficia-se de seis técnicas de pré-processamento textual:

- ***Tokenization****:* Processo em que é feita a divisão do texto bruto em unidades menores denominadas *tokens* (MANNING; RAGHAVAN; SCHÜTZE, 2008);
- ***Stemming:*** Procedimento cuja proposta é a obtenção do radical de cada palavra, de modo a eliminar as variações morfológicas de uma palavra (GUPTA; LEHAL, 2009);
- **Remoção de *Stopwords*:** *stopwords* são palavras que não possuem relevância na análise textual. As mais conhecidas são: preposições, pronomes, artigos, advérbios, e alguns verbos auxiliares (MORAIS; AMBRÓSIO, 2007);
- ***Part-of-speech tagging (Pos tagging)*:** É o processo em que se atribui um marcador a cada palavra, de um texto de entrada, identificando a qual classe gramatical cada palavra pertence (JURAFSKY; MARTIN, 2017b);
- **N-gramas:** São um conjunto de palavras que ocorrem dentro de uma determinada janela e, ao computar um n-grama, você geralmente move uma palavra para frente de modo que se compute o próximo n-grama. Por exemplo, para a frase “Assisti esse filme sozinho”, se n=2 (conhecido como bi-gramas), então o conjunto de n-gramas seria: “Assisti esse”, “esse filme” e “filme sozinho” (JURAFSKY; MARTIN, 2017a);

- ***Term Frequency Thresholding* (TFT):** Essa técnica é usada para eliminar palavras cujas frequências estão abaixo ou acima de um limite especificado. Este processo ajuda a melhorar o desempenho de classificação, pois termos que raramente aparecem em uma coleção de documentos possuem pouco poder discriminativo e termos que ocorrem frequentemente não têm poder discriminativo, podendo ambos serem eliminados (LAHLOU et al., 2013).

Entretanto, além das técnicas mencionadas anteriormente existem outras tais como: *Lemmatization, Dependency Parser*, *Named* Entity Recognition, entre outras (MANNING et al., 2014).

2.4.2 Transformação dos dados

A transformação dos dados é realizada após a etapa de pré-processamento. Este processo é necessário devido às limitações dos algoritmos de mineração de dados em interpretar padrões em um conjunto de atributos qualitativos (textos, datas, entre outros). Desta forma, é necessário representar na forma de atributos quantitativos (números inteiros ou reais).

Uma forma comum para esta representação é a utilização do modelo espaço vetorial, onde cada documento é representado por um vetor de elementos, e cada elemento do vetor esta associado a uma palavra da coleção de documentos. Cada elemento do vetor indica o grau de importância da palavra no conjunto de documentos (MORAIS; AMBRÓSIO, 2007)(HOTHO; NURNBERGER; PAASS, 2005).

Apesar de o modelo espaço vetorial não usar explicitamente alguma informação semântica, este modelo permite uma análise eficiente de uma grande quantidade de documentos, sendo usado em diversas abordagens de mineração de texto (HOTHO; NURNBERGER; PAASS, 2005).

Uma técnica frequentemente utilizada no cálculo de relevância de uma palavra, é a TF-IDF. Esta é uma medida estatística usada para avaliar a importância de uma palavra em um conjunto de documentos (MORAIS; AMBRÓSIO, 2007; HOTHO; NURNBERGER; PAASS, 2005).

Esta técnica é a combinação de duas técnicas estatísticas: o Term Frequency (TF) e o Inverse Document Frequency (IDF), representada pela seguinte fórmula:

$$TFIDF = TF * IDF \tag{2.1}$$

- **TF***:* Identifica a razão entre a frequência de uma palavra em um texto *(FreqTermo)*, pela quantidade de palavras no texto *(TotalTermos)*;

$$TF = (FreqTermo) \div (TotalTermos) \tag{2.2}$$

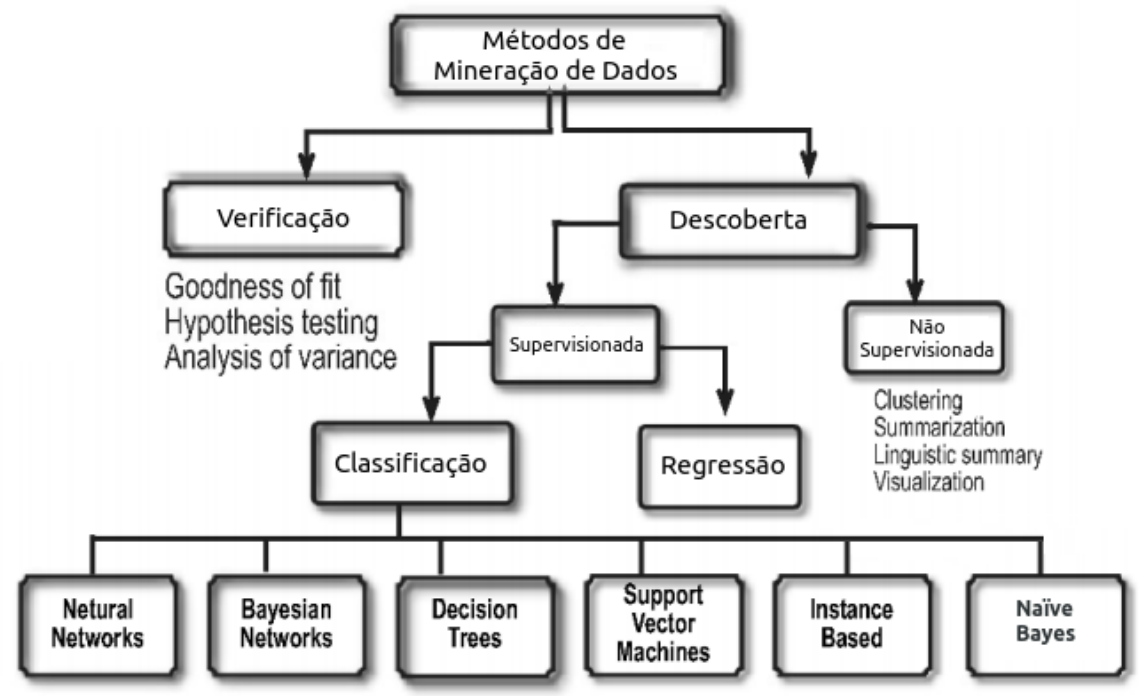
- **IDF***:* Mede o grau de importância daquela palavra considerando o conjunto de dados. Este valor é obtido pelo logaritmo da razão entre a quantidade de comentários que compõem a base de dados *(QtdRevisões)* pela quantidade de comentários que contem aquela palavra *(QtdRevisõesT).*

$$IDF = log((QtdRevises) \div (QtdRevisesT)) \tag{2.3}$$

2.4.3 Mineração dos Dados

Durante o processo de mineração de dados será aplicado de métodos preditivos (aprendizado supervisionado) ou descritivos (apredizado não supervisionado), para a descoberta de informações, conforme se pode observar na Figura 2.

Figura 2 – Métodos de Mineração de dados



Fonte: (Oded Maimon;Lior Rokach, (2010))

2.4.3.1 Métodos de aprendizagem supervisionada

Os métodos de aprendizagem supervisionada são métodos que tentam descobrir a relação entre os atributos de entrada, representados pelo vetor de características, e um atributo de saída denominado classe (ROKACH; MAIMON, 2010).

Os algoritmos de aprendizagem supervisionada utilizam um conjunto de dados de treinamento, para treinamento/construção do modelo, de forma que o mesmo possa fazer

previsões em cima de um novo conjunto de dados desconhecidos (BIRD; KLEIN; LOPER, 2009).

Os métodos supervisionados podem ser divididos em dois tipos: de classificação (classificadores) ou de regressão. Onde nos modelos de regressão o espaço de entrada é mapeado em um domínio de valor continuo, por exemplo, um conjunto Real, enquanto, os classificadores mapeiam o vetor de características a um conjunto discreto de classes pré-definidas (ROKACH; MAIMON, 2010).

Entre os algoritmos de aprendizagem supervisionada temos: *Multi Layer Perceptron, Decision Trees, Linear Models*, *Support Vector Machine* (SVM), *Naïve Bayes* (NB), entre outros (RUSSELL; NORVIG, 2009). Dentre os mencionados anteriormente, o SVM e o NB, serão destacados, pois estão sendo utilizados por trabalhos relacionados..

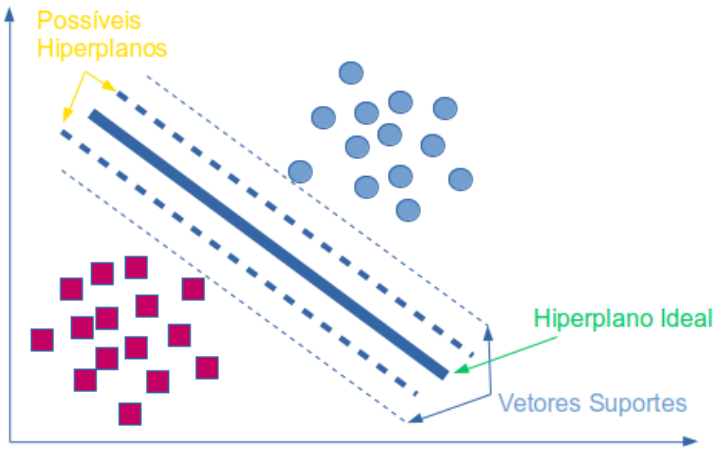
2.4.3.1.1 SVM

O SVM é um algoritmo de aprendizagem de supervisionada usado principalmente para reconhecimento de padrões e aplicado em problemas, tais como: de classificação de padrões, reconhecimento de imagem, reconhecimento de fala e categorização de texto (RUSSELL; NORVIG, 2009).

A ideia principal por trás do SVM é a construção de um hiperplano ideal, que pode ser usado para classificação de padrões linearmente separáveis. Este hiperplano ideal é um plano selecionado a partir do conjunto de planos, gerados na fase de treinamento. O objetivo principal do SVM é maximizar as margens para que possa classificar corretamente os padrões fornecidos, ou seja, quanto maior o tamanho da margem (distância entre o hiper plano e os vetores suportes), melhor ele classifica (SHMILOVICI, 2010; RUSSELL; NORVIG, 2009).

Na Figura 3 é representado o funcionamento do SVM.

Figura 3 – Representação do funcionamento do SVM



Fonte: Baseado em (Stuart J. Russell;Peter Norvig (2009))

2.4.3.1.2 NB

Os classificadores *Naïve Bayes* são baseados no Teorema de Bayes. Um classificador *Naïve Bayes* é um classificador probabilístico que usa o teorema de probabilidade de Bayes para inferir a classe de um conjunto de dados (KROCHMAL; HUSI, 2018).

Essa abordagem probabilística faz fortes suposições sobre como os dados são gerados e formulam um modelo probabilístico que incorpora essas suposições. Em seguida, eles usam uma coleção de exemplos de treinamento, para estimar os parâmetros do modelo generativo. No contexto do modelo generativo, o NB assume que todos os atributos dos exemplos são independentes um do outro. Embora essa suposição seja tomada como ingênua, em face da maioria das tarefas do mundo real demonstrar forte dependência, o algoritmo NB desempenha classificações muito bem (RUSSELL; NORVIG, 2009) .

2.4.3.2 Métodos de aprendizagem não supervisionada

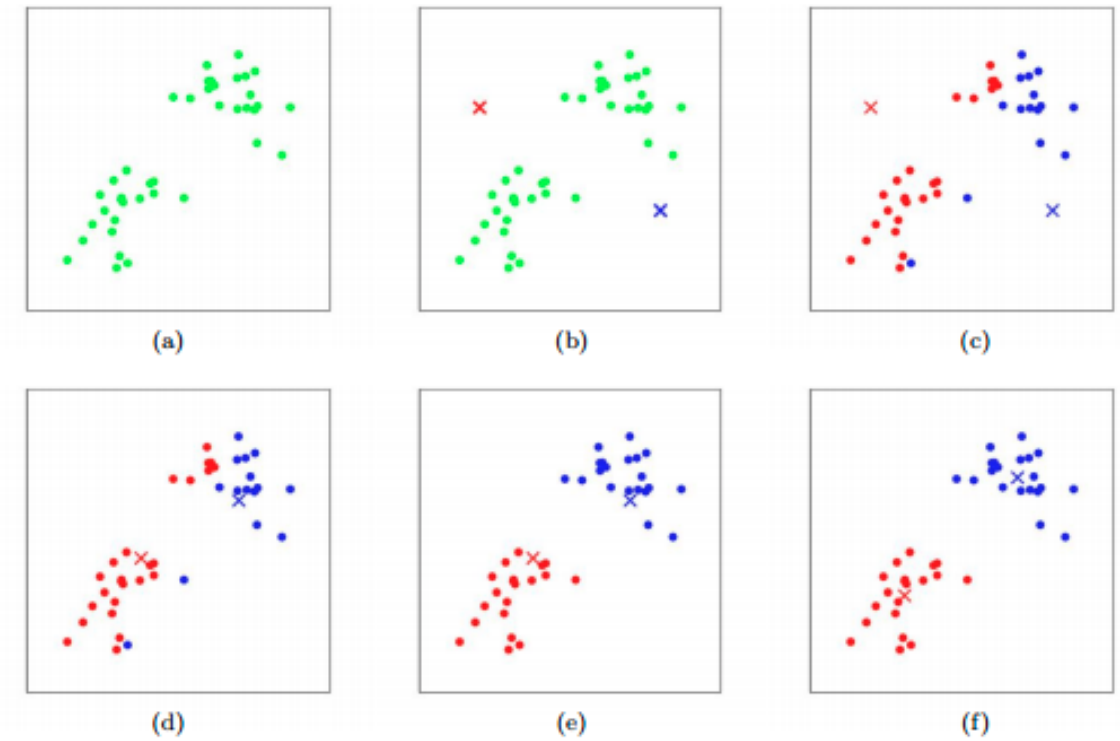
No aprendizado não supervisionado, o modelo aprende padrões na entrada, mesmo que nenhum conjunto de treinamento seja fornecido. Os algoritmos de aprendizagem não supervisionada mais comum são o de *clustering,* que realizam detecção de *clusters* potencialmente úteis de exemplos de entrada. Os *clusters* basicamente são agregações de itens que possuem características muito semelhantes, dessa forma tais modelos tentam agregar as entradas através de similaridade entre itens (ROKACH, 2010).

Entre os algoritmos de aprendizagem não supervisionada temos: *Self-Organizing Map*, *Fuzzy C-Means, K-Means,* entre outros (ROKACH, 2010). Porém, dentre eles pode-se destacar o *K-Means,* que é utilizado na pesquisa desenvolvida por Silva (2016), baseada em (BAUMAN; TUZHILIN, 2014).

2.4.3.2.1 K-Means

K-Means é um dos mais populares algoritmos de *clustering*. O algoritmo particiona os dados em K *clusters* (C_1, C_2, ..., C_k), representados por seus centros ou meios. O centro de cada *cluster* é calculado como a média de todas as instâncias pertencentes a esse *cluster* (ROKACH, 2010).

Figura 4 – Etapas da execução do K-Means por Piech (2013)



Fonte: (Chris Piech (2013))

Conforme ilustrado na Figura 4, o *K-Means* começa com um conjunto inicial de centros, escolhidos aleatoriamente ou de acordo com algum procedimento heurístico. Em cada iteração, cada instância é atribuída a um centro de *cluster* mais próximo, que é determinado de acordo com a distância euclidiana entre os dois. Em seguida, os centros dos *cluster* são recalculados, como a média de todas as instâncias pertencentes a esse *cluster* (ROKACH, 2010).

2.4.4 Avaliação do modelo

Nessa etapa o objetivo é avaliar um modelo perante um conjunto de dados inédito. Para isso, precisamos definir como metrificar o modelo e verificar o comportamento do modelo na prática. Para tal, métricas de avaliação serão definidas e técnicas de validação serão usadas, de modo a se atestar os índices de assertividade, bem como o assegurar a generalização de um determinado modelo (ZHENG, 2015; RUSSELL; NORVIG, 2009).

2.4.4.1 Métricas de Avaliação

Dentre as métricas de avaliação podemos destacar: Acurácia, Matriz confusão, Precisão, Cobertura e *F-Measure*.

2.4.4.1.1 Acurácia

Esta métrica simplesmente mede a frequência com que o classificador faz previsões corretas, e é tida como a razão entre o número de previsões corretas pelo número total de previsões realizadas (ZHENG, 2015). Abaixo temos a representação da formulá de cálculo da Acurácia:

$$\text{Acurácia} = \frac{\text{Predições Corretas}}{\text{Total de Predições}} \tag{2.4}$$

2.4.4.1.2 Matriz confusão

A Acurácia não é uma métrica suficiente para a valiar um modelo, pois esta não faz distinção entre classes. Ou seja, respostas corretas em classes distintas são tratadas igualmente, entretanto, verificar o grau de assertividade descriminado por classes é de suma importância para atestar a generalização do modelo (ZHENG, 2015).

Figura 5 – Exemplo de uma Matriz confusão

	Classificado como Positivo	Classificado como Negativo
Positivo	80	20
Negativo	5	195

Fonte: (ZHENG, 2015)

Em uma Matriz confusão a diagonal principal representa as classificações corretas, enquanto as outras posições representam os erros de classificação. Por exemplo, na Figura 5 pode-se visualizar que 80 itens foram classificados corretamente como positivo, 195 foram classificados corretamente como negativo, 20 positivos foram classificados erroneamente como negativos e 5 negativos foram classificados como positivo. Portanto, uma Matriz confusão mostra um detalhamento das classificações corretas e incorretas para cada classe.

2.4.4.1.3 Precisão e Cobertura

Na verdade, Precisão e Cobertura são duas métricas distintas, mas que frequentemente são utilizadas juntas. Precisão responde à pergunta: Dentre os itens que o classificador classificou em uma classe, quantos de fato pertencem a ela? Enquanto isso, a Cobertura responde à pergunta: Dentre os itens de uma classe, quantos foram classificados corretamente pelo classificador? (ZHENG, 2015).

A Precisão e a Cobertura são definidas pelas seguintes fórmulas:

$$\text{Precisão} = \frac{\text{Total Classificados Corretos}}{\text{Total Classificados Na Classe}} \tag{2.5}$$

$$\text{Cobertura} = \frac{\text{Total Classificados Corretos}}{\text{Total de Itens Na Classe}} \tag{2.6}$$

2.4.4.1.4 F-Measure

Segundo Zheng (2015), o *F-Measure* é definido como a média harmônica entre a Precisão e a Cobertura, conforme representado abaixo:

$$\text{F-Measure} = 2\frac{\text{Precisão} * \text{Cobertura}}{\text{Precisão} + \text{Cobertura}} \tag{2.7}$$

2.4.4.2 Validação de modelo

Para obter-se as métricas de avaliação reais de um modelo, é necessário validá-lo em um conjunto de dados inédito, de forma a reproduzir como este funcionará na prática, e determinar quão bem ele se generaliza para novos dados (RUSSELL; NORVIG, 2009). Entretanto, tudo que se tem é apenas um conjunto de dados, que será usado para treinamento do modelo, então como um modelo pode ser validado em um conjunto inédito de dados?

Para superar tais obstáculos algumas técnicas podem ser empregadas nesse processo: *Holdout cross-validation* e *K-fold cross-validation* (RUSSELL; NORVIG, 2009).

2.4.4.2.1 Holdout cross-validation

A abordagem consiste dividir os dados disponíveis em um conjunto de treinamento e um de validação, que é onde as métricas serão avaliadas. Esse método tem a desvantagem de não possuir boa cobertura. Ou seja, se for utilizado 50% dos dados para o conjunto de treinamento e a outra metade para validação, o modelo terá dificuldades em generalizar suas classificações. Por outro lado, se é reservado apenas 10% dos dados para o conjunto de validação, métricas imprecisas podem ser obtidas (RUSSELL; NORVIG, 2009).

2.4.4.2.2 K-fold cross-validation

Por outro lado, com a técnica de *K-fold cross-validation* mais dados para treinamento podem ser utilizados e ainda assim obter uma avaliação precisa. A ideia é que cada exemplo cumpra o dever duplo, como dados de treinamento e dados de validação. Primeiro, os dados são divididos em K subconjuntos iguais. Em seguida, é realizado K iterações de aprendizado. Em cada rodada, $1/K$ dos dados são mantidos como um conjunto de validação e o restante é usado como conjunto de treinamento. Dessa forma, a pontuação média dos conjuntos de testes é uma estimativa melhor do que pontuação de um único conjunto de validação (RUSSELL; NORVIG, 2009)

3 Trabalhos Relacionados

Como mencionado anteriormente, o CD-CARS proposto por Silva (2016) utiliza informações contextuais para fornecer sugestões aos usuários. Dentre as dimensões contextuais está o contexto de companhia, que é inferido das avaliações de produtos da Amazon[1] (livros, músicas e filmes), com o auxílio de um método de mineração de texto não supervisionado baseado em Bauman e Tuzhilin (2014), que se propunha em destacar informações contextuais em uma base de dados do Yelp[2] contendo avaliações de restaurantes, hotéis e *spas*.

Basicamente, o método de Bauman e Tuzhilin (2014) gera uma lista de palavras-chave ou tópicos referentes às informações contextuais nos comentários. A partir dessa lista, Silva (2016) seleciona manualmente os tópicos relacionados à dimensão contextual de companhia, e então cada tópico selecionado tem o seu grupo definido: Sozinho, Acompanhado, Família, Amigos, Casal e Colegas. Após isto, os comentários são classificados de acordo com o grupo dos tópicos relacionados ao comentário, através de uma técnica de correspondência de palavra. Nenhuma técnica de pré-processamento textual foi empregada por Silva (2016), ou por Bauman e Tuzhilin (2014).

Vale destacar que a metodologia adaptada por Silva (2016), para realização de classificação contextual de companhia, não produziu bons resultados. Na ocasião, após o processo de validação utilizando-se a técnica de *Holdout cross-validation,* o autor obteve as seguintes Acurácias: 19,67% ao classificar contexto de companhia no domínio de livros, 17% no domínio de filmes e 10,83% no domínio de música. Diante disso, o autor sugere que outros algoritmos poderiam ser utilizados de modo a melhorar o resultado da classificação dessas avaliações.

Lahlou et al. (2013) investigam o quanto preciso é possível inferir informações contextuais das avaliações dos usuários, através de abordagem de aprendizagem supervisionada e técnicas de pré-processamento textual. Os autores tentaram inferir a intenção de compra de comentários realizados pelos usuários. Para tal, duas bases de dados foram utilizadas na pesquisa: um conjunto de dados de avaliações de hotéis e outro conjunto de dados de avaliações de carros.

A pesquisa realizada por Lahlou et al. (2013) encontrou alguns desafios, entre eles: comentários mal escritos contendo diversos erros de digitação, conjunto diversificado de vocabulário, favorecendo o aparecimento de palavras com sentido ambíguo, resenhas curtas, muitos comentários avaliavam o item, mas não descreviam alguma informação contextual. Entretanto, mesmo em face de tais desafios a pesquisa conseguiu obter bons resultados, conseguindo estabelecer um método para classificação contextual, que fornecia um valor de

[1] https://www.amazon.com.br/
[2] https://www.yelp.com/

F-Measure de 72,58%, ao utilizar o processo de validação *K-fold cross-validation*.

Li et al. (2010) focam em descobrir se a inclusão de informações contextuais melhoram o desempenho da recomendação de itens. Diante disso, eles tentaram extrair o contexto de companhia de avaliações a restaurantes no Vale do Silício, e observaram com esta informação impacta nas recomendações.

Para isso, eles utilizaram abordagens de aprendizagem supervisionada em conjunto com técnicas de pré-processamento. Na ocasião, os autores utilizaram a técnica de *pos tagging*, para pré-processamento das avaliações e desenvolveram três classificadores: de regressão logística, baseado em regras e um classificador híbrido.

Ao término do processo de validação, onde foi utilizada a técnica de *K-fold cross-validation*, Li et al. (2010) conseguiram destacar um método que proporcionou um valor de *F-Measure* de 81,67%, que para eles é tido como um bom resultado. Além disso, concluíram que o contexto é fator importante, pois afeta diretamente as escolhas dos usuários. Entretanto, apesar dos bons resultados obtidos, os autores informam que a extração do contexto de companhia é mais desafiadora e geralmente produzem resultado significativamente inferiores quando comparado a outros contextos.

Campos, Rodríguez-Artigot e Cantador (2017) propôs uma abordagem, que realiza um mapeamento entre palavras e as categorias em uma taxonomia extraída do DBpedia[3]. Para tanto, são realizados dois passos: primeiro, selecionam-se as palavras que podem representar o contexto em avaliação, e então, se possível, tais palavras são mapeadas nas categorias de taxonomia.

As palavras selecionadas são aquelas que expressam declarações e opiniões na avaliação. Por exemplo, “Eu assisti o filme com os meus filhos em casa”, onde filhos e casa seriam anotados como contextos de companhia e de localização, respectivamente. Para atingir este objetivo, os autores utilizam a técnica de pré-processamento denominada *pos tagging*.

Campos, Rodríguez-Artigot e Cantador (2017) perceberam que a abordagem proposta obteve uma alta porcentagem de mapeamentos corretos, atingindo uma Acurácia de 84,2% no mapeamento contextual em uma base de dados de avaliações a itens (livros, música e filmes) da Amazon.

Dentre as metodologias de abordagem supervisionada, que tiveram bons resultados, a realizada por Lahlou et al. (2013) foi a selecionada para ser replicada e comparada a técnica utilizada por Silva (2016). A escolha por esta metodologia se dá por ela apresentar melhor reprodutibilidade que a metodologia implementada por Li et al. (2010).

[3] https://wiki.dbpedia.org/

Tabela 1 – Tabela das diferenças entre os trabalhos.

Trabalhos	**Base de dados**	**Abordagens utilizadas**	**Faz pré-processamento**	**Classifica contexto de companhia**	**Método Validação**	**Acurácia**	**F-Measure**
Bauman e Tuzhilin (2014)	Avaliação de restaurantes, hotéis e *spas*	Não Supervisionada	Não	Não	-	-	-
Silva (2016)	Avaliações de livros, CDs de música e filmes	Não supervisionada / Correspondência de palavras	Não	Sim	*Holdout cross-validation*	19,67% 17,00% 10,83%	-
Lahlou et al. (2013)	Avaliações de hotéis e de carros	Supervisionada	Sim	Sim	*K-fold cross-validation*	-	72,58%
Li et al. (2010)	Avaliações de restaurantes	Supervisionada	Sim	Sim	*K-fold cross-validation*	-	81,67%,
Campos, Rodríguez-Artigot e Cantador (2017)	Avaliações de livros, música e filmes	Correspondência de palavras	Sim	Sim	*K-fold cross-validation*	-	84,2%

Fonte: O Autor

4 Metodologia

Como mencionado na Seção 2, a presente pesquisa irá analisar/comparar diferentes metodologias, para inferência contextual de companhia, de modo a destacar aquela que fornecerá melhor classificação contextual, para serem empregadas em sistemas de recomendação sensíveis a contexto tais como o proposto por Silva (2016).

Será replicada a metodologia proposta por Lahlou et al. (2013), para ser comparada a metodologia utilizada por Silva (2016), que foi baseada em (BAUMAN; TUZHILIN, 2014), para determinar se o método proposto por Lahlou et al. (2013) pode ser empregado ao CD-CARS, para melhorar a classificação contextual de companhia.

Portanto, este capítulo descreverá as ferramentas utilizadas no desenvolvimento do trabalho, bem como a execução do experimento de análise.

4.1 Ambiente Experimental

Nesta seção serão fornecidas informações sobre o ambiente ao qual o experimento foi submetido, ou seja, informações relativas às tecnologias utilizadas e a base de dados que foi utilizada durante o experimento.

4.1.1 Tecnologias utilizadas

Para replicação das abordagens, foi utilizada a versão 3.7.2 da linguagem de programação Python[1]. Também foi utilizada uma biblioteca Python chamada *scikit learn*[2], utilizada para fazer mineração de dados.

Essa biblioteca funciona como um arcabouço, fornecendo várias ferramentas prontas para os pesquisadores, tais como:

- Algoritmos de classificação textual;
- Ferramentas para pré-processamento de dados;
- Ferramentas de métricas para análise;
- Ferramentas de recuperação de informação.

[1] https://www.python.org/
[2] http://scikit-learn.org/stable/index.html

Além do *scikit learn*, também é utilizado um conjunto de bibliotecas e programas chamado NLTK[3], para o processamento de linguagem natural e da Matplotlib[4], que é uma biblioteca para plotagem 2D de dados.

4.1.2 Base de dados

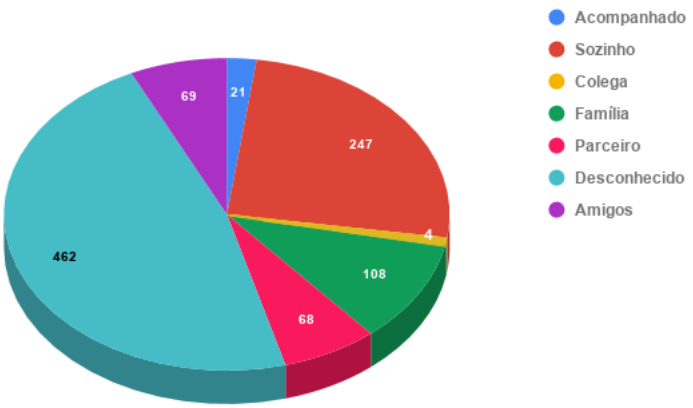
Os dados utilizados na pesquisa foram obtidos de (SILVA, 2016), que fez a extração de duas base de dados através de outro conjunto utilizado em (LESKOVEC; ADAMIC; HUBERMAN, 2007). Os conjuntos de dados extraídos por Silva (2016) continham avaliações em inglês sobre diferentes itens da Amazon[5], tais como: livros, músicas e filmes.

A base de dados extraída por Silva (2016) é extensa, contendo algumas dezenas de milhares de avaliações. Entretanto, o conjunto de dados da presente pesquisa teve que ser limitado a um tamanho menor, devido ao árduo processo de classificação manual dos mesmos, que foi executado por apenas um pesquisador.

Na ocasião foram selecionados ao todo 979 comentários, onde: 300 avaliações são referentes a livros, 300 avaliações relacionadas a música e 379 avaliações de filmes. Os domínios dos comentários foram aglutinados de modo a compor-se uma maior massa de comentários.

As classes contextuais das avaliações foram baseadas nas classes contextuais de companhia definidas em (SILVA, 2016). Na Figura 6, tem-se uma distribuição da base de comentários por classe contextual de companhia.

Figura 6 – Total de avaliações de itens nas três bases de dados



Fonte: O Autor

Como pode-se identificar na Figura 6, cerca de 47,2% das avaliações são pertencentes a classe “Desconhecido”, ou seja, são comentários que não descreve nenhum tipo contextual

[3] https://www.nltk.org/
[4] https://matplotlib.org/
[5] https://www.amazon.com.br/

de companhia, como por exemplo: "*Esse livro conta uma história maravilhosa. Conta uma aventura medieval magnifica!*". Dessa forma, foi decidido por não considerar tais comentários na base de dados.

Também foram desconsideradas as avaliações da classe contextual "Colega", traduzido do inglês *Colleague* referente a colegas de classe ou de trabalho, por representarem uma parcela de apenas 0,4% da base de comentários, não chegando a nem 1% da base inicial. Dessa forma, com essa baixa quantidade de avaliações não seria possível treinar de forma satisfatória os algoritmos de classificação, para predição deste tipo de classe.

Desta forma, a base final de dados[6] é composta por um total de 513 avaliações, como descrito na Tabela 2.

Tabela 2 – Conjunto final de dados

Classe	**Quantidade**
Acompanhado	21
Família	108
Amigos	69
Casal	68
Sozinho	247
Total	513

Fonte: O Autor

Um comentário pode ser definido do tipo:

- **Acompanhado**: Quando se consegue inferir que o usuário estava acompanhado no momento do consumo do item, mas não se consegue definir se foi com: um parente, um amigo ou um cônjuge;
- **Família**: Quando é identificado que o indivíduo consumiu o item acompanhado de algum familiar (tio, primo, pai, mãe, entre outros);
- **Amigos**: Quando fica evidenciado que a pessoa estava acompanhada de amigos;
- **Casal**: Quando se consegue determinar que o individuo descreveu que estava acompanhado do: esposo, esposa, namorado ou namorada;
- **Sozinho**: Quando se nota que o usuário consumiu o item sozinho.

No Apêndice B pode ser visualizado exemplos dos comentários por classe.

[6] https://github.com/douglashss/reviews_amazon_dataset

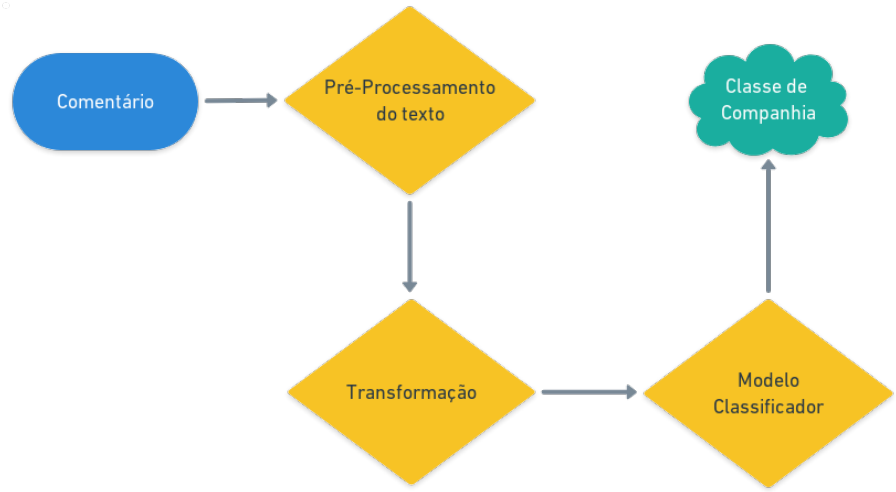
4.2 Experimento

Como mencionando no início deste capítulo, o experimento em questão consistirá na análise de duas abordagens para inferência contextual de companhia.

4.2.1 Metodologia de inferência contextual de companhia em Lahlou et al. (2013)

Nesta abordagem inicialmente os documentos serão submetidos a técnicas de pré-processamento, após o pré-processamento cada documento será transformado/representado em um modelo vetorial e por fim, acontecerá a classificação contextual dos comentários. Na Figura 7 pode ser observada uma representação visual do passo a passo da metodologia realizada por Lahlou et al. (2013).

Figura 7 – Etapas do processo de classificação dos comentários realizado por Lahlou et al. (2013)



Fonte: O Autor

4.2.1.1 Pré-processamento

As técnicas de pré-processamento utilizadas por Lahlou et al. (2013) foram: *tokenization*, *stemming*, remoção de *stopwords*, *pos tagging*, n-gramas e TFT. Os autores geraram sete configurações de combinação dessas técnicas, que estão representadas na Tabela 3.

Tabela 3 – Configurações das técnicas de pré-processamento.

I	Stemming + Unigrama
II	Stemming + UniBigrama
III	Stemming + UniBiTrigrama
IV	Stemming + TFT + Unigrama
V	Stemming + TFT + UniBigrama
VI	Stemming + TFT + UniBiTrigrama
VII	TFT + UniBiTrigrama

Fonte: (F. Z. Lahlou et al. (2013))

Para todas as configurações testadas, foi executado um método de normalização de texto, onde todas as palavras foram convertidas para minúsculas e tiveram os caracteres acentuados convertidos para sua forma sem acento. Além disso, os textos tiveram os sinais de pontuação e algumas *stopwords* removidas, conforme proposto por Lahlou et al. (2013).

Durante o processo de remoção de *stopwords*, se teve o cuidado de não remover palavras caracterizadas como pronomes, pois se considera que esta classe gramatical é relevante para o problema de classificação contextual de companhia. Para tal, foi utilizada a ferramenta de *pos tagging*, para marcar as palavras com sua respectiva classe gramatical, facilitando no controle de remoção das *stopwords.*

O *stemming* é utilizado com a finalidade de diminuir as variações das palavras por meio da radicalização destas, colaborando assim no aumento do grau de relevância das mesmas.

A técnica de TFT, é utilizada para remover termos com pouca relevância, e que ocorreram apenas uma vez no conjunto de dados.

Além disso, em determinadas configurações é usado unigramas ou combinações de unigramas e bigramas, bem como de unigramas, bigramas e trigramas, para que se enriqueça ainda mais as características extraídas dos comentários.

Mais detalhes, sobre a replicação das técnicas de pré-processamento, podem ser observados no Apêndice B.

4.2.1.2 Transformação

Após o pré-processamento, é realizada a transformação dos comentários pré-processados em um modelo que seja entendido pelos algoritmos de classificação. Este processo consiste no mapeamento dos termos em um modelo que possa ser interpretado pelos algoritmos de classificação.

Para a transformação dos comentários pré-processados é utilizado o modelo espaço vetorial, mencionado na Seção 2.4.2, e para cálculo de relevância de palavras, é utilizada a técnica de TF-IDF, também citada na Seção 2.4.2, conforme proposto por Lahlou et al. (2013).

Para isso, duas ferramentas foram utilizadas: o *TfidfVectorizer* e o *StemmedTfidfVectorizer*. A primeira é fornecida pelo *scikit learn,* enquanto a segunda foi construída baseando-se no *TfidfVectorizer,* que diferentemente dela não aplica a técnica de *stemming* aos textos. Mais detalhes de utilização e configuração podem ser vistos no Apêndice B.

4.2.1.3 Classificação

Na etapa de classificação contextual dos comentários Lahlou et al. (2013) faz a utilização de dois classificadores: o NB e o SVM, ambos algoritmos de aprendizagem supervisionada, e já implementados na biblioteca *scikit learn*.

Entretanto, na metodologia proposta por Lahlou et al. (2013), não são fornecidos os parâmetros de configurações utilizados pelos modelos classificadores.

Portanto, a presente pesquisa procurou definir os parâmetros de configuração de ambos classificadores, por meio da aplicação da técnica de otimização de hiper parâmetros. A otimização visa definir um conjunto ótimo de parâmetros para configuração de um determinado modelo classificador. Para tal, foi empregada a técnica mais utilizada para esse processo de otimização denominada *grid search* (BERGSTRA; BENGIO, 2012). Esta técnica simplesmente executa uma pesquisa exaustiva através de um subconjunto de hiper parâmetros, especificados manualmente.

Após a aplicação do *grid search,* o modelo classificador NB teve os parâmetros de configuração: *alpha* ajustado para '0,1' e o *fit_prior* para '*False*'. Por sua vez*,* o modelo de classificação SVM teve os parâmetros: *C* ajustado para '1', o *gamma* definido como '0,1' e o *kernel* escolhido foi '*rbf*'.

Diferentemente da metodologia empregada por Lahlou et al. (2013), que utilizou a técnica de validação *K-fold cross-validation*, a presente pesquisa utilizou o método *Holdout cross-validation*, descrito na Seção 2.4.4.2.1. Este método foi utilizado de forma a se manter o padrão dos métodos de validação empregado na análise das metodologias em (SILVA, 2016; LAHLOU et al., 2013). Desta forma, durante a validação a base de dados foi dividida em duas partes: a primeira parte foi utilizada no processo de treinamento dos classificadores, enquanto a segunda foi utilizada para validar os modelos finais.

4.2.2 Metodologia de inferência contextual de companhia em Silva (2016)

Nesta abordagem a classificação contextual de companhia é obtida pela utilização de um algoritmo de mineração de texto não supervisionado, em conjunto com um classificador baseado em correspondência de palavras. Esta metodologia foi baseada em Bauman e Tuzhilin (2014),

cujo proposito era a descoberta de informações contextuais em avaliações de usuários para fins de recomendação.

Inicialmente, Silva (2016), por meio de algoritmo de aprendizagem não supervisionada, separa as avaliações em duas categorias: especificas, que descrevem detalhes específicos de um item, e genéricas, que descrevem comentários gerais sobre um item. Essa divisão é realizada, pois segundo Bauman e Tuzhilin (2014) tópicos descrevendo as informações contextuais aparecem mais frequentemente em avaliações específicas do que em genéricas.

Para a separação das avaliações nas categorias mencionadas anteriormente, foi utilizado o método de agrupamento *K-means,* que teve a quantidade de centroides definida em 2. Para tal, um conjunto de características foram determinadas para as avaliações, são elas:

- LogSentences: Logaritmo do número de frases na avaliação mais 1.
- LogWords: Logaritmo do número de palavras na avaliação mais 1.
- VBDsum: Logaritmo do número de verbos no passado, que a avaliação possui mais 1.
- Vsum: Logaritmo do número de verbos na avaliação mais 1.
- VRatio - A razão entre VBDsum and Vsum $\frac{VBDsum}{Vsum}$.

Vale salientar que

Após a separação das avaliações, é gerada uma lista de tópicos relevantes, a partir das avaliações especificas. Para tal, é executada uma filtragem de substantivos, respeitando as seguintes restrições:

1) Identifique, para cada comentário R_i, o conjunto de substantivos N_i que aparecem nele.
2) Determine, para cada substantivo n_k, suas frequências ponderadas $w^s(n_k)$ correspondentes às avaliações específicas, conforme:

$$w^s(n_k) = \frac{|R_i : R_i \in specific \wedge n_k Ni|}{|Ri : Ri \in specific|} \tag{4.1}$$

3) Filtre as palavras n_k que no geral têm baixa frequência, conforme:

$$w(n_k) = \frac{|R_i : n_k Ni|}{|Ri : Ri \in specific|} < \alpha \tag{4.2}$$

onde α é um limite. Por exemplo, $a = 0,005$.

4) Após a filtragem, defina o conjunto de sentidos, para cada substantivo restante, usando o WordNet (MILLER, 1995).

5) Combine os sentidos em grupos g_t por significados próximos usando a distância da taxonomia WordNet.

6) Determine, para cada grupo, suas frequências ponderadas $w^s(g_t)$, através das frequências de seus itens, conforme:

$$w^s(n_k) = \frac{|R_i : R_i \in specific \wedge g_t \cap N_i \neq \oslash|}{|Ri : Ri \in specific|} \tag{4.3}$$

7) Ordene os grupos por suas frequências ponderadas $w^s(g_t)$ em ordem decrescente.

Após gerar a lista de grupos(ou tópicos) ordenados, a presente pesquisa selecionou manualmente apenas os tópicos relacionados à dimensão contextual de companhia, conforme proposto por Silva (2016). Na ocasião, os tópicos estavam associados com os seguintes grupos de contexto: Acompanhado, Amigos, Casal, Família, Colegas, e Sozinho.

Para classificação dos comentários, um classificador baseado em correspondência de palavras(ou tópicos) foi criado, baseando-se no trabalho de Silva (2016).

Basicamente o classificador percorre uma lista de tópicos selecionados, e caso haja a ocorrência de um tópico no comentário, este será classificado com o respectivo grupo do tópico, caso contrário, será classificado com o contexto desconhecido. Por exemplo, se na listagem existe o tópico "Mãe" classificado como "Família", caso o comentário tenha a ocorrência da palavra "Mãe" o mesmo sera classificado como "Família". Mais detalhes sobre o funcionamento do classificador podem ser vistos no Apêndice D.

Para validação desta metodologia, foi utilizada a técnica de *Holdout cross-validation,* mencionada na Seção 2.4.4.2.1. Na ocasião a base de dados foi dividida ao meio. Desta forma, a primeira metade foi utilizada no processo da geração de tópicos contextuais de companhia, enquanto a segunda metade utilizada para validar o modelo final gerado.

4.2.3 Métricas de avaliação

As métricas adotadas para o processo de validação dos resultados da análise foram: Acurácia*, F-measure* e a Matriz confusão. A métrica de Acurácia foi adotada, pois essa foi a métrica utilizada por Silva (2016) em seu processo de avaliação. Por sua vez, a métrica de *F-Measure* foi a métrica adota por Lahlou et al. (2013). Portanto, de forma a padronizar-se a análise de ambas metodologias, tanto os valores da Acurácia quanto do *F-Measure* serão considerados no processo de análise de resultados.

Já a Matriz confusão será utilizada com a finalidade de garantir uma visualização mais detalhada das predições realizadas por ambas metodologias replicadas na presente pesquisa. Esse detalhamento, poderá ajudar na identificação de possíveis problemas de predição de determinadas classes. Na matriz confusão, será considerado que quanto maiores os valores da diagonal principal, melhor o modelo será.

5 Resultados

Esta seção apresenta os resultados obtidos no experimento descrito no Capítulo 4.

5.1 Resultados da classificação seguindo a metodologia proposta por Lahlou et al. (2013)

Na Tabela 4 é apresentado as Acurácias de cada configuração de técnicas de pré-processamento, separados por classificador.

Tabela 4 – Acurácias obtidas ao replicar a metodologia proposta porLahlou et al. (2013)

Configurações de pré-processamento	**NB**	**SVM**
Config1 - Stemming + Unigrama	**61,25%**	70,04%
Config2 - Stemming + TFT + Unigrama	65,13%	71,54%
Config3 - Stemming + UniBigrama	62,90%	72,70%
Config4 - Stemming + TFT + UniBigrama	67,01%	73,04%
Config5 - Stemming + UniBiTrigrama	63,23%	72,99%
Config6 - Stemming + TFT + UniBiTrigrama	67,95%	**74,32%**
Config7 - TFT + UniBiTrigrama	66,52%	72,24%

Fonte: O Autor

Conforme pode-se observar, todas as Acurácias obtidas pelo classificador SVM foram superiores quando comparadas àquelas obtidas pelo NB. Além disso, tanto a Config-4, quanto a Config-6, em conjunto com o classificador SVM, foram as que produziram melhores Acurácias durante a experimentação, tendo ambas fornecido uma Acurácia de 74,32%. Todavia, a utilização da Config-1 em conjunto com o classificador NB foi a combinação que produziu a pior Acurácia, durante o experimento.

Por sua vez, a Tabela 5 representa os valores de *F-Measure* obtidos, para cada configuração de pré-processamento e também dividida por classificador.

Tabela 5 – *Valores de F-Measure* obtidos ao replicar a metodologia proposta porLahlou et al. (2013)

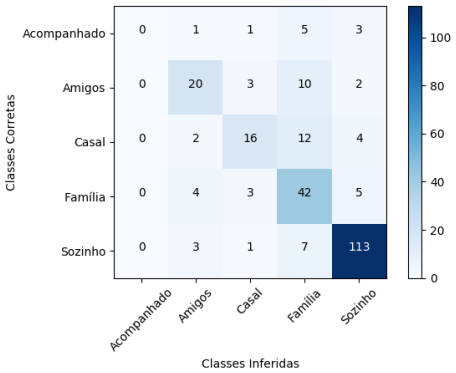
Configurações de pré-processamento	NB	SVM
Config1 - Stemming + Unigrama	**35,95%**	48,20%
Config2 - Stemming + TFT + Unigrama	38,45%	49,52%
Config3 - Stemming + UniBigrama	37,60%	52,04%
Config4 - Stemming + TFT + UniBigrama	39,00%	52,95%
Config5 - Stemming + UniBiTrigrama	38,33%	53,62%
Config6 - Stemming + TFT + UniBiTrigrama	41,02%	**54,27%**
Config7 - TFT + UniBiTrigrama	39,25%	52,37%

Fonte: O Autor

Como se pode visualizar na tabela anterior, os valores de *F-Measure* no geral apresentaram um valor bem abaixo, quando comparado a métrica de Acurácia. Além disso, ao analisar os valores de *F-Measure*, confirma-se que a Config-6 é a melhor configuração, quando utilizadas em conjunto com o SVM, produzindo um valor de *F-Measure* de 54,27%.

Ao se verificar a matriz confusão da Config-6 utilizada em conjunto com o SVM, representada da Figura 8, percebe-se que, apesar do péssimo resultado ao classificar o contexto "Acompanhado", no geral as classes foram bem classificadas. Olhando para a diagonal principal nota-se que na maior parte das vezes o classificador produziu mais acertos de que erros.

Figura 8 – Matriz confusão da Config-6 com SVM



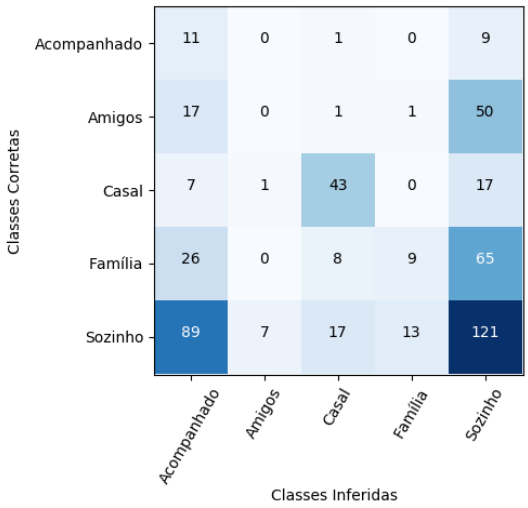
Fonte: O Autor

5.2 Resultados da classificação seguindo a metodologia desenvolvida em Silva (2016)

Ao analisar a metodologia utilizada por Silva (2016), que foi baseada em (BAUMAN; TUZHILIN, 2014), percebe-se que os resultados não foram tão satisfatórios, quando comparado à metodologia executada por Lahlou et al. (2013). Na ocasião, a metodologia utilizada por Silva (2016) produziu uma Acurácia de 35,8% e um valor de *F-Measure* igual a 19,5%.

Além disso, quando se verifica a Matriz confusão, representada na Figura 9, evidencia-se que, apesar de obter bons resultados ao classificar os contextos: "Acompanhado" e "Casal", em outras classes o resultado não foi tão satisfatório. No geral a abordagem adotada por Silva (2016) produziu mais erros que acertos, confirmando o resultado não satisfatório reportado pelo autor. Além disso, a Matriz confusão reflete bem o baixo valor de F-Measure obtido.

Figura 9 – Matriz confusão do Classificador por correspondência de palavras



Fonte: O Autor

5.3 Análise dos Resultados

Analisando detalhadamente os resultados obtidos na Seção 5.1, é possível extrair algumas informações. Primeiramente, pode-se notar que a utilização da técnica de TFT, que é utilizada para remover termos que ocorreram apenas uma vez no conjunto de dados, pode promover melhoras no nível de classificação em ambos modelos classificadores, conforme comparação dos pares: (Config-1, Config-2), (Config-3, Config-4) e (Config-5, Config-6). Esses termos, que ocorrem apenas uma vez no conjunto de dados, não contribuem na classificação, pois aparecem em um documento no conjunto de treinamento ou no de teste. Por esse motivo, pode-se dizer que eles não têm poder discriminatório, portanto podem ser desconsiderados.

Outro detalhe importante que pode ser observado, ainda na mesma metodologia proposta por Lahlou et al. (2013), é a utilização de n-gramas. Provavelmente o uso de combinação de

unigramas e bigramas, bem como de unigramas, bigramas e trigramas produziram bons efeitos nos resultados finais dos modelos. Isso pode ter sido ocasionado pelo fato da combinação de n-gramas promover o enriquecimento das características extraídas dos comentários. O impacto desta técnica de pré-processamento fica evidente quando se compara os seguintes pares: (Config-1, Config-3), (Config-2, Config-4), (Config-3, Config-5) e (Config-4, Config-6).

Outro impacto positivo, nos resultados finais obtidos na Seção 5.1, foi causado pela utilização da técnica de pré-processamento denominada *stemming*. Esta melhora pode ser percebida, ao se comparar o par de configuração (Config-6, Config-7). O incremento pode ter sido proporcionado pelo fato desta técnica diminuir as variações das palavras por meio da radicalização, colaborando assim no aumento do grau de relevância das mesmas.

No geral, embora seja possível selecionar um método que tenha fornecido uma Acurácia relativamente boa, os valores de *F-Measure*, ao se replicar a metodologia proposta por Lahlou et al. (2013), foram bem abaixo, quando comparados àqueles obtidos na pesquisa original (LAHLOU et al., 2013). Na pesquisa Lahlou et al. (2013) destaca um método que obteve um valor de *F-Measure* igual a 72,58%. Diante deste contexto, a perspectiva da presente pesquisa era obter um resultado semelhante ao aplicar tal metodologia em outra base de dados.

A primeira hipótese para tal desempenho, é que tais resultados podem ter acontecido devido ao desbalanceamento da quantidade de comentários por classes, conforme pode ser evidenciado na Tabela 2. Uma segunda hipótese, é que a quantidade de avaliações talvez não tenha sido suficiente para prover um bom treinamento para o modelo. Além disso, tanto o desbalanceamento quanto a baixa quantidade de avaliações do tipo “Acompanhado”, pode ter sido determinante para o péssimo desempenho ao classificar comentários desse tipo de classe.

Por sua vez, os baixos resultados obtidos, ao se aplicar a metodologia descrita em Silva (2016) à base de dados da presente pesquisa, podem ter sido gerados devido a abordagem empregada. Certamente, a utilização de um classificador contextual de companhia baseado em correspondência de tópicos não seja a melhor abordagem para resolver tal problemática. Além disso, o fato de tal abordagem não fazer o uso de técnicas de pré-processamento textual pode ter interferido nos resultados.

Comparando ambas metodologias, pode-se concluir que a metodologia proposta por Lahlou et al. (2013) de fato proporcionou um resultado melhor para classificação contextual de companhia, quando comparada com a metodologia utilizada em Silva (2016). Entretanto, ambas metodologias ficaram abaixo do esperado, que era obter um valor de *F-Measure em torno de* 72,58%, conforme apresentado no trabalho original de Lahlou et al. (2013).

6 Conclusão

A proposta da presente pesquisa, foi analisar metodologias para classificação contextual de companhia. Para tal, duas metodologias foram selecionadas. A primeira foi a proposta por Lahlou et al. (2013), e a segunda foi a metodologia, baseada em (BAUMAN; TUZHILIN, 2014), que foi utilizada na pesquisa do CD-CARS desenvolvida por Silva (2016). O objetivo da análise é atestar se o método proposto por Lahlou et al. (2013) pode ser incorporado ao CD-CARS (SILVA, 2016), para que tal RS possa produzir melhores sugestões, ao considerar a dimensão contextual de companhia.

Para tanto, ambas metodologias foram replicadas, e então foram analisadas algumas métricas, tais como: Acurácia, *F-Measure* e Matriz confusão.

Ao término da análise é observado que embora a metodologia proposta por Lahlou et al. (2013) tenha produzido resultados poucos satisfatórios, quando comparado aos obtidos na pesquisa original (LAHLOU et al., 2013), ela se saiu melhor que a técnica utilizada no CD-CARS (SILVA, 2016). Na ocasião, a metodologia proposta por Lahlou et al. (2013) proporcionou um aumento da Acurácia em 107,60% e um aumento do *F-Measure* em 178,31%, quando comparada àquela utilizada por Silva (2016).

Portanto, pode-se concluir que a metodologia em (LAHLOU et al., 2013), pode sim ser uma boa opção como método para classificação contextual de companhia no CD-CARS (SILVA, 2016), de modo a melhorar ainda mais as sugestões fornecidas ao CD-CARS.

6.1 Desafios e Limitações

O principal desafio para a pesquisa foi referente a escassez de informações contextuais de companhia nos comentários. Como o propósito principal de uma avaliação é expressar opiniões sobre os itens, muitas vezes os usuários não descreviam o contexto e expressavam apenas suas considerações sobre o item. Desta forma, ao classificar os comentários manualmente muitos deles tinham o contexto determinado como desconhecido, e portanto, terminavam sendo descartados da nossa base de dados final. Diante disto, nossa base de dados sofreu uma redução de 979 para apenas 513 comentários, conforme pode ser observado na Seção 4.2.1.2.

Outro desafio também encontrado foi o balanceamento das quantidades de comentários por classe contextual de companhia, conforme pode ser observado na Seção 4.2.1.2. Devido ao desbalanceamento, comentários da classe contextual “Colega” tiveram que ser descartados, pois com uma quantidade pequena não seria possível treinar satisfatoriamente os algoritmos de classificação, para predição deste tipo de classe.

Devido a esses desafios, algumas limitações surgiram na pesquisa. Talvez, a quantidade

limitada de comentários, tenha interferido no poder de generalização e classificação dos algoritmos utilizado na metodologia proposta por Lahlou et al. (2013), impossibilitando melhores resultados.

Outro fator limitante é que a análise poderia ter incluído outras metodologias mais recentes, como a realizada por Campos, Rodríguez-Artigot e Cantador (2017), de forma que a análise tivesse uma melhor cobertura ao analisar outros métodos para classificação contextual de companhia.

Além disso, ambas as métodologias aqui replicadas, não lidam com o problema de ambiguidade da palavras. Ambos os métodos pressupõe que as palavras possuem um único vetor semântico. Isso pode ser um problema para a classificação contextual, pois dependendo do contexto da palavra, as mesmas podem não estar associadas ao contexto de companhia. Portanto, utilização de técnicas para desambiguidade de sentido das palavras podem proporcionar melhores resultados, na proposta de Lahlou et al. (2013).

Outro ponto importante, é com relação ao processo de *stemming*, que pode ser custoso em um contexto real de uma base de dados possuido milhares de comentários. A grande quantidade de comentários que serião pré-processados, poderia exigir uma grande quantidade de memória, inviabilizando assim o processo de stemming em grande lotes de dados.

6.2 Trabalhos Futuros

- Revalidação desta pesquisa de análise utilizando um conjunto de dados melhor balanceado e com uma massa de comentários maior, pois a base de dados utilizada pode de alguma forma ter interferido nos resultados finais da pesquisa;
- Aplicar a metodologia proposta por Lahlou et al. (2013) ao CD-CARS proposto por Silva (2016), e analisar se acontece melhoria nas recomendações realizadas pelo CD-CARS. Embora a proposta de Lahlou et al. (2013), tenha proporcionado resultados poucos satisfatórios na presente pesquisa, esta metodologia se mostrou melhor que a abordagem de classificação contextual de companhia desenvolvida por Silva (2016);
- Utilizar outros algoritmos de classificação tal qual o *Random Forest*, além do *Naïve Bayes* e o *SVM;*
- Utilizar outras características textuais tais como conjugação verbal, pois verbos no plural podem colabora na identificação de que o usuário estava acompanhado de alguém.
- Utilizar e analisar algoritmos de *deep learning*, tais como redes neurais convolucionais, para a classificação contextual de companhia;
- Analisar as metodologias aqui replicadas com outros trabalhos relacionados mais recentes, tais como (CAMPOS; RODRÍGUEZ-ARTIGOT; CANTADOR, 2017).

- Validar a classificação manual dos comentários, da base utilizada nesta pesquisa, com outros especialistas;
- Utilizar outras bases de dados, como por exemplo a do Yelp[1], para validação das metodologias aqui replicadas em diferentes bases de dados;
- Adaptar a proposta de Lahlou et al. (2013), para executar uma classificação em duas etapas: primeiro para determinar se o contexto de companhia do comentário é "Desconhecido", e segundo caso não seja informar qual o devido contexto. Pois em bases reais, pode acontecer de muitos comentários possuírem o contexto "Desconhecido".
- Adaptar a proposta de Lahlou et al. (2013), para utilizar técnica de desambiguidade de sentidos de palavras, como por exemplo aquela proposta por Chen, Liu e Sun (2014).

[1] https://www.yelp.com/

Referências

ADOMAVICIUS, G.; TUZHILIN, A. Context-Aware Recommender Systems. In: ____. *Recommender Systems Handbook*. 2. ed. [S.l.]: Springer, 2015. cap. 6, p. 191 – 226. Citado 2 vezes nas páginas 13 e 18.

AGGARWAL, C. C. Content-Based Recommender System. In: ____. *Recommender Systems*. [S.l.: s.n.], 2016. cap. 4, p. 139 – 166. ISBN 978-3-319-29659-3. Citado na página 17.

AMATRIAIN, X.; BASILICO, J. Recommender Systems in Industry: A Netflix Case Study. In: ____. *Recommender Systems Handbook*. 2. ed. Nova Iorque: Springer, 2015. cap. 11, p. 385 – 419. ISBN 978-1-4899-7637-6. Citado na página 13.

BAUMAN, K.; TUZHILIN, A. Discovering Contextual Information from User Reviews for Recommendation Purposes. In: CBRECSYS 2014 - WORKSHOP ON NEW TRENDS IN CONTENT-BASED RECOMMENDER SYSTEMS, 2014, Silicon Valley, CA, USA. Silicon Valley, CA, USA., 2014. p. 2 – 8. Citado 10 vezes nas páginas 14, 15, 25, 29, 31, 32, 37, 38, 42 e 44.

BERGAMASCHI, S.; GUERRA, F.; LEIBA, B. Guest Editors' Introduction: Information Overload. *IEEE Internet Computing*, IEEE, v. 14, n. 6, p. 10 – 13, Novembro 2010. ISSN 1089-7801. Disponível em: <http://ieeexplore.ieee.org/abstract/document/5617056/>. Acesso em: 01/09/2017. Citado na página 13.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, p. 281 – 305, Fevereiro 2012. Disponível em: <http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>. Citado na página 37.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. 1. ed. [S.l.]: O'Reilly Media, 2009. ISBN 9780596516499. Citado na página 24.

BORRÀS, J.; ANTÔNIO MORENO; VALLS, A. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, v. 41, n. 16, p. 7370 – 7389, Novembro 2014. Citado na página 13.

CAMPOS, P. G.; RODRÍGUEZ-ARTIGOT, N.; CANTADOR, I. Extracting context data from user reviews for recommendation: A Linked Data approach. In: *ComplexRec*. [S.l.: s.n.], 2017. Citado 5 vezes nas páginas 14, 15, 30, 31 e 45.

CANTADOR, I. et al. Cross-Domain Recommender Systems. In: ____. *Recommender Systems Handbook*. 2. ed. [S.l.]: Springer, 2015. cap. 27, p. 919 – 959. ISBN 978-1-4899-7637-6. Citado 2 vezes nas páginas 18 e 19.

CHEN, X.; LIU, Z.; SUN, M. A Unified Model for Word Sense Representation and Disambiguation. In: LINGUISTICS, A. for C. (Ed.). *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1025 – 1035. Citado na página 46.

COLOMBO-MENDOZA, L. O. et al. RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, Elsevier, v. 42, n. 3, p. 1202 – 1222, 2015. Citado na página 14.

DAVIDSON, J. et al. The YouTube video recommendation system. In: ‘10, R. (Ed.). *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys ‘10*. Nova Iorque,: ACM, 2010. p. 293 – 296. Disponível em: <http://doi.acm.org/10.1145/1864708.1864770>. Citado 2 vezes nas páginas 13 e 18.

ELAHI, M.; RICCI, F.; RUBENS, N. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, Elsevier, p. 29 – 50, Junho 2016. Citado na página 17.

FERNÁNDEZ-TOBÍAS, I. et al. Cross - domain recommender systems: A survey of the State of the Art. In: *Spanish Conference on Information Retrieval*. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 18 e 19.

FERNÁNDEZ-TOBÍAS, I. et al. Accuracy and Diversity in Cross-domain Recommendations for Cold-start Users with Positive-only Feedback. *RecSys ’16 Proceedings of the 10th ACM Conference on Recommender Systems*, ACM, New york, p. 119 – 122, Setembro 2016. ISSN 978-1-4503-4035-9. Citado na página 18.

GEMMIS, M. de et al. Semantics-Aware Content-Based Recommender Systems. In: ____. *Recommender Systems Handbook*. 2. ed. New york: Springer, 2015. cap. 4, p. 119 – 159. ISBN 978-1-4899-7636-9. Citado na página 17.

GUPTA, V.; LEHAL, G. S. A Survey of Text Mining Techniques and Applications. *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE*, v. 1, p. 60 – 76, Agosto 2009. Citado na página 21.

HOTHO, A.; NURNBERGER, A.; PAASS, G. A Brief Survey of Text Mining. *Journal for Language Technology and Computational Linguistics*, p. 19 – 62, Maio 2005. Citado 2 vezes nas páginas 20 e 22.

JURAFSKY, D.; MARTIN, J. H. Language Modeling with N-grams. In: ____. *Speech and Language Processing*. 3. ed. [s.n.], 2017. cap. 4, p. 35 – 59. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>. Citado na página 21.

JURAFSKY, D.; MARTIN, J. H. Part-of-Speech Tagging . In: ____. *Speech and Language Processing*. 3. ed. [s.n.], 2017. cap. 10, p. 1 – 25. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/10.pdf>. Citado na página 21.

KROCHMAL, M.; HUSI, H. Knowledge Discovery and Data Mining. In: ____. *Integration of Omics Approaches and Systems Biology for Clinical Applications*. 1. ed. [S.l.]: John Wiley & Sons, Inc, 2018. cap. 14, p. 233 – 247. Citado na página 25.

LAHLOU, F. Z. et al. Inferring Context from Users’ Reviews for Context Aware Recommendation. In: ____. *Research and Development in Intelligent Systems XXX*. 1. ed. [S.l.]: Springer International Publishing, 2013. p. 227 – 239. Citado 21 vezes nas páginas 8, 9, 12, 14, 15, 22, 29, 30, 31, 32, 35, 36, 37, 39, 40, 41, 42, 43, 44, 45 e 46.

LEE, H.; KWON, J. Personalized TV Contents Recommender System Using Collaborative Context tagging-based User’s Preference Prediction Technique. *International Journal of Multimedia and Ubiquitous Engineering*, v. 9, n. 5, p. 231 – 240, 2014. Citado na página 14.

LESKOVEC, J.; ADAMIC, L. A.; HUBERMAN, B. A. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, v. 1, n. 1, p. 5 –, Maio 2007. Citado na página 33.

LI, Y. et al. Contextual recommendation based on text mining. In: THE 23RD INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS (COLING 2010), 2010, Beijing, China. Beijing, China: Association for Computational Linguistics, 2010. p. 692 – 700. Citado 4 vezes nas páginas 14, 15, 30 e 31.

MAIMON, O.; ROKACH, L. Introduction to Knowledge Discovery and Data Mining. In: ____. *Data Mining and Knowledge Discovery Handbook*. 2. ed. [S.l.]: Srpinger, 2010. cap. 1, p. 1 – 15. Citado na página 20.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. The term vocabulary and postings lists. In: ____. *Introduction to Information Retrieval*. 1. ed. Cambridge University Press, 2008. cap. 2, p. 19 – 47. ISBN 0521865719. Disponível em: <https://nlp.stanford.edu/IR-book/pdf/02voc.pdf>. Citado na página 21.

MANNING, C. D. et al. The Stanford CoreNLP Natural Language Processing Toolkit. 2014. Disponível em: <https://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf>. Citado na página 22.

MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, v. 38, n. 11, p. 39 – 41, 1995. Citado na página 38.

MORAIS, E. A. M.; AMBRÓSIO, A. P. L. *Mineração de Textos*. [S.l.], 2007. Citado 2 vezes nas páginas 21 e 22.

OH, S. et al. Comparison of Techniques for Time Aware TV Channel Recommendation. *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, IEEE, p. 989 – 992, 2014. ISSN 978-1-4799-5955-6. Citado na página 14.

PIECH, C. *K Means*. 2013. Disponível em: <http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>. Citado 2 vezes nas páginas 8 e 26.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender Systems: Introduction and Challenges. In: ____. *Recommender Systems Handbook*. 2. ed. Nova Iorque: Springer, 2015. cap. 1, p. 1 – 34. ISBN 978-1-4899-7637-6. Citado 2 vezes nas páginas 13 e 17.

RISTOSKI, P.; PAULHEIM, H. Semantic Web in data mining and knowledge discovery: A comprehensive survey. *WebSemantics: Science, Services and Agents on the World Wide Web*, Elsevier, v. 36, p. 1 – 22, Janeiro 2016. Disponível em: <https://doi.org/10.1016/j.websem.2016.01.001>. Citado 2 vezes nas páginas 20 e 21.

ROKACH, L. A survey of Clustering Algorithms. In: ____. *Data Mining and Knowledge Discovery Handbook*. 2. ed. [S.l.]: Springer, 2010. cap. 14, p. 271 – 298. Citado 2 vezes nas páginas 25 e 26.

ROKACH, L.; MAIMON, O. Supervised Learning. In: ____. *Data Mining and Knowledge Discovery Handbook*. 2. ed. [S.l.]: Springer, 2010. cap. 8, p. 133 – 147. Citado 2 vezes nas páginas 23 e 24.

RUSSELL, S. J.; NORVIG, P. Learning from examples. In: ____. *Artificial Intelligence*: A modern approach (3rd edition). 3. ed. [S.l.]: Pearson, 2009. cap. 18, p. 693 – 767. ISBN 0136042597. Citado 4 vezes nas páginas 24, 25, 26 e 28.

SASSI, I. B.; MELLOULI, S.; YAHIA, S. B. Context-aware recommender systems in mobile environment: On the road of future research. *Information Systems*, v. 72, p. 27 – 61, Dezembro 2017. Citado na página 13.

SCHEDL, M. et al. Music Recommender Systems. In: ____. *Recommender Systems Handbook*. 2. ed. Nova Iorque: Springer, 2015. cap. 13, p. 453 – 492. ISBN 978-1-4899-7637-6. Citado na página 13.

SHMILOVICI, A. Support Vector Machines. In: ____. *Data Mining and Knowledge Discovery Handbook*. 2. ed. [S.l.]: Springer, 2010. cap. 12, p. 229 – 247. Citado na página 24.

SILVA, D. V. e. *CD-CARS*: CROSS-DOMAIN CONTEXT-AWARE RECOMMENDER SYSTEMS. 2016. 240 p. Tese (Ciência da Computação) — Universidade Federal de Pernambuco, Recife. Citado 16 vezes nas páginas 12, 15, 19, 25, 29, 30, 31, 32, 33, 37, 38, 39, 42, 43, 44 e 45.

SMITH, B.; LINDEN, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, IEEE, v. 21, n. 3, p. 12 – 18, Maio 2017. Citado na página 13.

VERBERT, K. et al. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *IEEE Transactions on Learning Technologies*, IEEE, v. 5, n. 4, p. 318 – 335, Abril 2012. Citado na página 13.

YANG, Z. et al. A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications. *IEEE Access*, IEEE, v. 4, p. 3273 – 3287, Maio 2016. ISSN 2169-3536. Citado 2 vezes nas páginas 13 e 17.

ZHENG, A. *Evaluating Machine Learning Models A Beginner's Guide to Key Concepts and Pitfalls*. 1. ed. [S.l.]: O'Reilly Media, 2015. Citado 3 vezes nas páginas 26, 27 e 28.

ZHU, H. et al. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *Journal ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 5, n. 4, p. 58:1 – 58:27, Dezembro 2014. Citado na página 18.

Apêndices

APÊNDICE A – Exemplos de avaliações por categoria

Tabela 6 – Exemplos de avaliações por categoria

Avaliação	Categoria
Watch for the song no me queda mas by far the highlight of the album is the song no me queda mas. i used to play it for my students while teaching in albania. we had a lesson where each student had to bring a song we would listen to it and we would explain the personal significance it had for us. the song brings gentleness to my heart the sweetness that was selena the simple and adorned style she sometimes could do so well. perhaps what impresses me the most about selena is the amazing variety of styles she could sing in. my only wish she could have sung more songs like the beautiful lo me queda mas.	Acompanhado
This is not a cartoon! first let me say that my kids have warmed up to this video and now watch it. it seems to be a good bedtime video. but it is not an animated cartoon the way you expect. it is more like pictures with people reading the story. some kids might like that but it took mine awhile to get into it.	Família
Great film i loved this movie! we were watching it at a friend's house great music too! i wanna look for the soundtrack if there is one. it was sad at times but overall a great piece of work!!!	Amigos
You'll fall in love every night!! i can't say enough about this cd. i am not a big rod stewart fan but this has always been mine and my husband's cd. it is so awesome. every song on it is great. it is especially nice for a romantic evening. any time i hear a song from this cd i am flooded with memories. don't pass this one up!!	Casal
Amazing acting of nicholson hunt and kinnear i don‘t get tired of watching this movie!what can we say about jack nicholson? and i‘m in love with helen hunt. little things in ordinary life can inspire a great story.	Sozinho

APÊNDICE B – Pré-processamento e Vetorização de documentos

```
from copy import copy
from nltk import pos_tag, SnowballStemmer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

# ##############################################################################
# # Lista personalizada de Stopwords. Pronomes não são considerados como   #
# # Stopwords                                                                  #
# ##############################################################################

STOP_WORDS = [
    word for word in stopwords.words('english')
    if pos_tag([word], tagset='universal')[0][1] not in ['PRON']
]

class StemmedTfidfVectorizer(TfidfVectorizer):
    """
    Classe que assim como o TfidfVectorizer é responsável pela
    representação dos documentos no modelo espaço vetorial.

    A diferença é que o StemmedTfidfVectorizer executa a técnica
    de stemming, enquanto o TfidfVectorizer não.
    """
    def build_analyzer(self):
        stemmer = SnowballStemmer(language='english')
        analyzer = super(TfidfVectorizer, self).build_analyzer()
        return lambda doc: (stemmer.stem(w)
                            for w in analyzer(doc)
                            if not w.isdecimal())
```

```
def vetorizador_texto(texto, vetorizador):
    """
    Método para vetorizar uma lista de textos
    """
    vetorizador = copy(vetorizador)
    texto = vetorizador.fit_transform(texto).toarray()
    return texto, len(vetorizador.vocabulary_)

# ##############################################################################
# # O Term Frequency Thresholding (TFT) é definido pelo paramentro min_df.     #
# ##############################################################################

# ##############################################################################
# # O N-grama é definido pelo parâmetro ngram_range.                           #
# # ngram_range = (1, 2)  gera unigramas e bigramas                            #
# # ngram_range = (1, 3)  gera unigramas, bigramas e trigramas                 #
# # quando omitido apenas unigramas são gerados                                #
# ##############################################################################

# ##############################################################################
# # Por padrão o StemmedTfidfVectorizer e o TfidfVectorizer ja utilizando      #
# # a técnica de tf-idf                                                        #
# ##############################################################################

parametros_padrao = {'strip_accents': 'unicode', 'stop_words': STOP_WORDS}

# Stemming + Unigrama
vetorizador1 = StemmedTfidfVectorizer(
    use_idf=False, **parametros_padrao,
)

# Stemming + TFT + Unigrama
vetorizador2 = StemmedTfidfVectorizer(
    use_idf=False, min_df=2, **parametros_padrao,
)



# Stemming + Unigrama Bigrama
```

```
vetorizador3 = StemmedTfidfVectorizer(
    ngram_range=(1, 2), **parametros_padrao,
)

# Stemming + TFT + Unigrama Bigrama
vetorizador4 = StemmedTfidfVectorizer(
    min_df=2, ngram_range=(1, 2), **parametros_padrao,
)

# Stemming + Unigrama Bigrama Trigrama
vetorizador5 = StemmedTfidfVectorizer(
    ngram_range=(1, 3), **parametros_padrao,
)

# Stemming + TFT + Unigrama Bigrama Trigrama
vetorizador6 = StemmedTfidfVectorizer(
    min_df=2, ngram_range=(1, 3), **parametros_padrao,
)

# TFT + Unigrama Bigrama Trigrama
vetorizador7 = TfidfVectorizer(
    min_df=2, ngram_range=(1, 3), **parametros_padrao,
)
```

APÊNDICE C – Otimização de hiper parâmetros

```
from sklearn import svm, naive_bayes, model_selection

def svc_param_selection(X, y):
    param_grid = {
        'C': [0.001, 0.01, 0.1, 1, 10],
        'gamma': [0.001, 0.01, 0.1, 1],
        'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    }

    grid_search = model_selection.GridSearchCV(
        svm.SVC(),
        param_grid,
    )
    grid_search.fit(X, y)

    return grid_search.best_params_

def nb_param_selection(X, y):
    param_grid = {
        'alpha': [0.001, 0.1, 0.5, 1],
        'fit_prior': [True, False],
    }

    grid_search = model_selection.GridSearchCV(
        naive_bayes.MultinomialNB(),
        param_grid,
    )
    grid_search.fit(X, y)

    return grid_search.best_params_
```

APÊNDICE D – Método para classificação de contexto de companhia por correspondência de tópicos

```
from collections import OrderedDict

reviews = open('reviews.txt')
topics_and_their_groups = open(
    'selected_contextual_topics_labeled_with_groups.txt'
)

topics_dict = OrderedDict()
for line in topics_and_their_groups.readlines():
    group_of_companion, topic = line.replace('\n', '').split('|')
    topics_dict[topic] = group_of_companion

reviews_classes = []
for line in reviews.readlines():
    tokens = set(line.split('\t')[-1].replace('\n', '').split('|'))

    review_group = 'DESCONHECIDO'
    for topic, group_of_companion in topics_dict.items():
        if topic in tokens:
            review_group = group_of_companion
            break

    reviews_classes.append(review_group)

with open('result.txt', 'w') as result:
    result.writelines('\n'.join(reviews_classes))
```