Classificação automática de questões do ENADE utilizando o algoritmo KNN

Samuel Oliveira, Ismar Frango

[1]Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie (UPM)
São Paulo – SP – Brasil

Abstract. *Most of the digitally generated data is in unstructured format, more precisely in text, because of this the task of extracting, analyzing and organizing this data becomes essential and works related to text classification have been gaining more and more prominence. The generation of textual data can occur in several ways due to the vast amount of documents available in digital form such as news sites, customer reviews, online encyclopedias, social networks, etc.. In Brazil there are exams applied nationally in order to evaluate the country's level of education in stages of education. For higher education, we have the National Student Performance Exam (ENADE), and there are several works statistically analyzing the results of this test using microdata containing test scores, number of correct answers per question, answers to questionnaires, etc. On the other hand, there are few studies exploring the content and questions of the test itself in its raw text format. In this context, this work aims to apply and evaluate the k-nearest neighbors (KNN) algorithm in the task of classifying the questions of ENADE for the Computer Science course in its five years of application in previously defined themes.*

Resumo. *Grande parte dos dados gerados digitalmente estão em formato não estruturado, mais precisamente em texto, por conta disso a tarefa de extrair, analisar e organizar esses dados se torna essencial e trabalhos relacionados a classificação de texto vem ganhando cada vez mais destaque. A geração de dados textuais pode ocorrer de diversas formas devida a vasta quantidade de documentos disponíveis em forma digital como site de noticias, avaliações de clientes, enciclopédias online, redes sociais, etc. e apesar de serem mais difíceis de lidar, uma vez compreendidos os dados textuais podem resultar em uma fonte rica e complementar aos dados estruturados mais convencionais. No âmbito educacional, podemos citar as avaliações como fonte textual, no Brasil existem exames aplicados nacionalmente afim de avaliar o nível educacional do país em diferentes estágios da educação. Direcionado ao ensino superior temos o Exame Nacional de Desempenho dos Estudantes (ENADE), há diversos trabalhos analisando estatisticamente os resultados desta prova usando como base os microdados contendo as notas dos exames, quantidade de acertos por questão, respostas aos questionários, etc. Em contrapartida, existem poucos trabalhos explorando o conteúdo e questões da prova em si, em seu formato bruto de texto. Nesse contexto, o presente trabalho tem como objetivo aplicar e avaliar o algoritmo k-nearest neighbors (KNN) na tarefa de classificação das questões do ENADE para o curso de Ciências da Computação, em seus cinco anos de aplicação em temas previamente definidos.*

1. Introdução

Exames nacionais são ferramentas importantes na avaliação da educação de um país em diferentes níveis. Além de fornecerem indicadores de qualidade de ensino essenciais para o subsídio de novas políticas púbicas educacionais, também são fontes claras e confiáveis aos gestores, pesquisadores, educadores, e púbico em geral que podem servir como norteadoras para compreender e aprimorar a qualidade de ensino e aprendizagem em seus ambientes.

As avaliações são aplicadas desde o nível básico até o nível superior de educação, em relação ao último temos as provas do Exame Nacional de Desempenho dos Estudantes (ENADE), o qual faz parte do Sistema Nacional de Avaliação da Educação Superior (SINAES), a prova avalia o rendimento dos concluintes dos cursos de graduação em relação aos conteúdos programáticos previstos nas diretrizes curriculares dos seus respectivos cursos, o desenvolvimento de competências e habilidades necessárias ao aprofundamento da formação geral e profissional, e o nível de atualização dos estudantes com relação à realidade brasileira e mundial. Além de formular e aplicar a prova o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), órgão responsável pela prova, divulga também relatórios das provas para cada curso avaliado naquele ano. Os chamados "Relatórios síntese de área" apresentam estatísticas e observações detalhadas a partir dos resultados dos estudantes de cada graduação. As análises são feitas tanto em cima das questões, mesurando quantitativamente erros e acertos e comentando a respeito da dificuldade de questões específicas por exemplo, quanto em cima dos questionários que compõe a prova, analisando os aspectos demográficos e socioeconômicos extraídos das respostas. [Ferreira 2014]

As provas são compostas por uma parte de Formação Geral, comum aos cursos de todas as áreas, e uma parte de Componente Específico, própria de cada área de avaliação. Em relação a primeira, são 10 questões, sendo 2 discursivas e 8 objetivas, já no componente específico temos 30 questões, sendo 3 discursivas e 27 objetivas. Focaremos no componente específico da prova, mais precisamente no componente específico de Ciências da Computação nos anos 2005, 2008, 2011, 2014 e 2017. [Ferreira 2014]

Utilizando princípios de mineração de dados, processamento de linguagem natural e aprendizado de máquina supervisionado, este trabalho tem como objetivo aplicar o algoritmo KNN na base de questões extraídas e avaliar seus resultados na tarefa de classificação dessas questões em temas previamente selecionados.

2. Referencial Teórico

Esta seção apresenta conceitos gerais envolvendo mineração de dados, aprendizado de máquina, representação e classificação de textos e trabalhos semelhantes encontrados na literatura.

2.1. Mineração de Dados

As definições para o termo Mineração de Dados podem variar de acordo com o campo de conhecimento que as apresentam. Portanto, aqui nos atentaremos a duas em específico, a definição de Mineração de Dados no campo da aprendizagem de máquina e no de Mineração de Texto.

Na perspectiva do primeiro, em [Fayyad et al. 1996] a definição é dada da seguinte maneira: ”Mineração de Dados é um passo no processo de Descoberta de Conhecimento que consiste na realização da análise dos dados e na aplicação de algoritmos de descoberta que, sob certas limitações computacionais, produzem um conjunto de padrões de certos dados.”

A obra citada anteriormente nos dá os fundamentos da Mineração de Dados, porém foca em sistemas de armazenamentos de dados estruturados como os bancos de dados, como neste trabalho estamos considerando provas e relatórios disponibilizados na web no formato PDF, cujo conteúdo é majoritariamente na forma de texto, é preciso buscar o embasamento em ramos da Mineração de Dados que lide com esse tipo de dados não estruturados, neste ponto temos a Mineração de Textos. De acordo com [Tan et al. 1999] Mineração de Textos, em inglês Text Mining (TM), refere-se geralmente ao processo de extração de padrões interessantes e não triviais ou conhecimento de documentos textuais não estruturados e pode ser visto como uma extensão de Mineração de Dados ou KDD.

2.2. Aprendizado de Máquina

Podemos definir Aprendizado de Máquina (AM) como: ”[...] é uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores. [...]”[e José Augusto Baranauskas 2003]

Dentro de AM existem diversas outras subáreas, focaremos aqui naquelas que fazem mais sentido para se trabalhar junto a mineração de textos. Aprendizado supervisionado e Classificação, as quais serão discutidas com mais detalhes a seguir.

2.3. Aprendizado supervisionado

Aprendizado de máquina supervisionado ou induzido refere-se ao método onde os resultados esperados são fornecidos ao sistema. É o processo de aprender um conjunto de regras a partir de instâncias/exemplos, de maneira geral, criar um classificador capaz de generalizar a partir de novos exemplos [Kotsiantis et al. 2007]

[Kotsiantis et al. 2007] define alguns passos para a aplicação do AM supervisionado em problemas reais. O primeiro passo seria a coleta do dataset, se um expert na área estiver disponível ele(a) pode sugerir quais campos são os mais pertinentes nos dados em questão, no nosso caso quais termos classificariam cada matéria por exemplo, se não o método de “força bruta” pode ser usado, que consiste em avaliar tudo disponível na esperança de que as características relevantes sejam isoladas. Entretanto, o autor menciona que esse método não é adequado para AM por indução, uma vez que resulta em grande parte de ruídos e valores faltantes, portanto exige um pré-processamento significativo. O segundo passo é justamente a preparação e pré-processamento dos dados, no caso de textos, estamos nos referindo na maior parte às técnicas pertencentes ao processamento de linguagem natural (PLN) que será apresentada ainda nessa seção.

2.4. Classificação

Como dito anteriormente, a Classificação é uma subárea do AM Supervisionado. Seu objetivo, resumidamente, é atribuir rótulos a determinadas entradas no sistema. Sistemas de

Classificação são usados geralmente quando as previsões são de natureza distinta, neste caso a natureza das previsões irão se distinguir em razão do campo de conhecimento em que a questão analisada se encaixa. Existem diversos algoritmos de aprendizagem voltados a classificação, por estarmos tratando de dados puramente textuais e não estruturados, a escolha de qual usar parte da primeira etapa que é o pré-processamento do texto afim de facilitar sua manipulação no sistema.g

[Patra and Singh 2013] Estabelece alguns estágios para o processo de classificação em texto. A Figura 1 é uma adaptação desse processo feita por Alencar em [Silva 2020] e se encaixa no proposito deste trabalho.

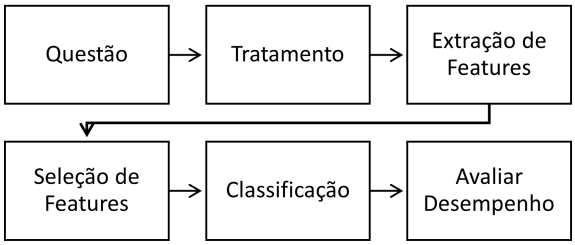

Figura 1. Estágios da classificação (Autor)

2.5. k-nearest neighbors (KNN)

KNN é um dos algoritmos mais simples e mais utilizados no aprendizado de máquina. Seu uso consiste na classificação por votação de vários exemplos de treinamento rotulados com suas menores distâncias de cada objeto, sendo assim, tradicionalmente é utilizada a distância euclidiana para esse cálculo [Patra and Singh 2013].

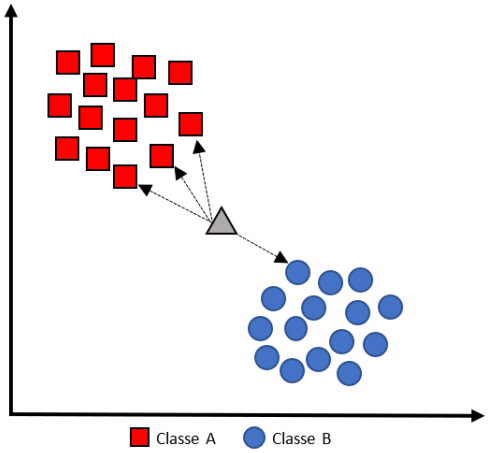

Figura 2. Arquitetura do modelo KNN para duas classes (Autor)

Dado um documento como entrada, o algoritmo ranqueia seus vizinhos mais próximos entre os documentos de treinamento, e utiliza as categorias dos vizinhos com

os melhores ranques para definir sua categoria. A pontuação de similaridade entre o documento de entrada e seus vizinhos é usada como peso de cada uma de suas categorias, e a soma dos pesos das categorias para os k vizinhos próximos são usados para ranquear as categorias [Yang 1999]. Logo, o "k"no nome faz referência a quantidade de "vizinhos"(*kneighbors* no inglês) que serão comparados no momento da classificação do objeto, a escolha do k é importante, pois ira suavizar os limites entre as classes presentes, geralmente o usado $\sqrt{n}$, onde n é a quantidade de amostras no conjunto de dados.

A Figura 2 ilustra simplificadamente o algoritmo KNN em um plano 2D, onde um objeto ainda não classificado será avaliado em relação a seu pertencimento à classe A ou B. Entretanto, em tarefas reais envolvendo classificação é quase impossível não se deparar com casos em que haja *multi-label*, no qual uma classe pode pertencer a mais de uma categoria. Em notícias, uma mesma reportagem pode se categorizar em saúde e esporte; em classificação de imagens, uma mesma pode conter gato e cachorro. Assim, no caso desse trabalho não é diferente, uma mesma questão pode pertencer a mais de um conteúdo. Exitem diversas maneiras de endereçar esse casos nos algoritmos de aprendizados, geralmente eles se dividem em duas formas conhecidas na literatura [Katakis et al. 2008]:

1. **Transformação do problema**: consiste em dividir o problema *multi-label* em tarefas menores do tipo *single-label*. Esta abordagem é intuitiva e mais utilizada, porém não leva em consideração as correlações entre as diferentes *labels* presentes em cada instância e pode apresentar resultados fracos [Zhang and Zhou 2007]

2. **Adaptação de algoritmos**: consiste em estender o funcionamento de um algoritmo específico para que ele consiga lidar com dados *multi-label* diretamente.

Com relação ao último, uma implementação derivada do KNN tradicional capaz de lidar com classificação *multi-label* nativamente, é proposta em [Zhang and Zhou 2007] com o nome de ML-KNN i.e *Multi-Label k-Nearest Neighbor*. Em resumo, o algoritmo adaptado funciona da seguinte forma: primeiramente, para cada instância de teste, seus k vizinhos mais próximos no conjunto de treinamento são identificados. Em seguida, de acordo com informações estatísticas obtidas com os conjuntos de *labels* dessas instâncias vizinhas, ou seja, o número de instâncias vizinhas pertencentes a cada classe possível, máximo a posteriori (MAP) é utilizado para determinar o conjunto de rótulos para a instância de teste [Zhang and Zhou 2007]. Então, a partir de testes em três diferentes problemas de classificação, o trabalho concluiu que a performance do algoritmo é superior a técnicas tradicionais usadas em problemas *multi-label*.

2.6. Representação de Texto

A representação de texto é um aspecto importante na Classificação e consiste em reduzir a complexidade dos documentos, assim fazendo com que sejam mais fáceis de lidar, logo o documento é transformado de sua versão inicial completamente textual para um vetor de documento [Khan et al. 2010]. Esta etapa é primordial para a aplicação dos algoritmos de classificação e está diretamente relacionada aos resultados alcançados [Rossi 2016].

O modelo espaço-vetorial é geralmente utilizado quando trabalhamos com coleções de textos. Nele, os documentos são representados por vetores e as dimensões correspondem à termos ou atributos da coleção de textos [Rossi 2016]. Esta representação proposta por [Salton et al. 1975], é relativamente simples e permite o uso de algoritmos de AM tradicionais.

2.7. Processamento de Linguagem Natural (PLN)

[Liddy 2001] define PLN como uma série de técnicas computacionais para analisar e representar textos que ocorrem naturalmente em um ou mais níveis de análise linguística com o objetivo de alcançar o processamento de linguagem para uma gama de tarefas ou aplicações. Assim, focaremos aqui no uso do PLN como meio para o pré-processamento em análises de texto.

De acordo com [Alahmadi 2016], a fase de pré-processamento consiste na tarefa de converter o texto em um conjunto de termos bem definidos. Esses termos podem ser palavras, conceitos ou uma combinação dos dois. Nesta etapa também é levada em consideração a remoção de ruídos e irregularidades que afetam negativamente a performance da classificação. Isso inclui a remoção de dígitos, pontuações, pronomes, preposições, etc. Podemos citar como técnicas comuns de pré-processamento: [Pota et al. 2015]

1. **Tokenização**: é o processo de dividir o texto em fragmentos denominados *tokens*.

2. **Eliminação de *stop Words***: Descarte de palavras comuns como pronomes, preposições, artigos, caracteres numéricos e de pontuação. Em outras palavras, a remoção do que não tem relevância para determinada classificação.

3. ***Tagging***: associação de classes a cada *token*, como artigo, verbo, advérbio, etc.

4. ***Stemming***: processo de reduzir os *tokens* para suas palavras de origem, por exemplo, as palavras "gato", "gata", "gatos" e "gatas" reduzidas a "gato" e as palavras "tiver", "tenho", "tinha", "tem" reduzidas a "ter". Portanto, a vantagem de aplicar o *stemming* é a redução de vocabulário e abstração de significado.

2.8. Extração de termos

A fase de pré-processamento pode resultar na extração de diversos tipos de termos e influenciar diretamente na performance do processo de classificação. Esses termos são geralmente classificados em: [Pota et al. 2015]:

1. **Léxicos**: são as palavras no contexto de uma questão. Exemplo de técnica, *Bag-of-Words* que corresponde a casa par (t, f), onde t representa a palavra da questão e f a quantidade de vezes em que essa palavra aparece.

2. **Sintáticos**: termos com origem da estrutura sintática da palavra. Por exemplo, *Part-of-speech (POS)-tags* que consiste em converter as palavras para a forma *(word, tag)* onde tag é a classe sintática da palavra como advérbio, pronome, verbo, etc.

3. **Semânticos**: termos associados a uma determinada classificação da questão. Podemos citar como exemplo os *Hypernyms/Hiponyms* que são palavras que representam o conceito semântico de generalização/especialização. Por exemplo: Cor/(Azul, Amarelo, Vermelho) ou Flor/(Rosa, Jasmim, Orquídea).

2.9. *Term-Weighting*

Term-Weighting, como o nome sugere, é o processo de atribuir pesos aos termos extraídos. Os pesos são usados para criar um vetor com esse termo para questão (Modelo Espaço - Vetorial) e a partir dai, aplicar técnicas para selecionar os termos mais relevantes. Podemos citar as seguintes técnicas como as mais utilizadas nessa tarefa: [Emmanuel et al. 2013]

1. **Binária**: consiste basicamente em atribuir 1 caso o *feature* aparecer em uma determinada questão e 0 caso contrário;

2. ***Term Frequency (TF)***: baseia-se no total de ocorrências de um *feature* em uma mesma questão;

3. ***Inverse Document Frequency (TF*IDF)***: o objetivo desta técnica é ajustar o valor obtido por TF, uma vez que os termos comuns em muitas questões normalmente não contribuem para a classificação e possuem alta frequência. A fórmula para o cálculo do TF*IDF para cada *feature* se da por:

$$TF * IDF = TF * \log(\frac{N}{n})$$

   Onde **N** representa o total de questões e **n** o número de questões em que o *feature* está presente.

2.10. Trabalhos Correlatos

É possível encontrar na literatura diversos trabalhos que focam na análise dos dados resultantes do ENADE. O próprio INEP divulga, a cada aplicação, o Relatório de Curso analisando de forma geral os resultados para cada área em relação as questões, questionários e demografia. A maioria dos trabalhos encontrados se baseiam nos microdados disponibilizados pelo INEP, assim focando na análise do desempenho do ensino superior de maneira geral ou um curso específico daqueles avaliados nos anos em questão.

[Cretton and Gomes 2016] objetivam a extração de conhecimento e compreensão do nível de dificuldade do componente específico da prova utilizando mineração de dados e KDD, por meio da plataforma WEKA utilizando os microdados disponibilizados, com enfoque no curso de medicina. A metodologia do trabalho foi baseada nas etapas que compõem o processo de KDD. Essas etapas consistiram na seleção dos dados, onde os atributos (colunas) pertinentes foram selecionados e as linhas referente ao curso em questão separadas do todo. No pré-processamento, etapa responsável pelo tratamento e normalização dos dados, os quais foram selecionados anteriormente, viabiliza a aplicação dos algoritmos de mineração de dados. A transformação antecede a mineração em si, aqui o foco é a formatação dos dados de forma com que facilite a mineração, sendo assim, os autores descrevem a tarefa de substituir os valores em código de certos atributos, com suas respectivas descrições como pertencente a esta etapa. Por fim, temos a mineração de dados, como dito anteriormente, a mineração de dados é uma tarefa que engloba diversas técnicas utilizadas para explorar grandes volume de dados, portanto, a utilizada nesse trabalho em questão foi a classificação utilizando árvore de decisão como o algoritmo J48. Dessa forma, este algoritmo toma como ponto inicial o atributo de maior significância o qual aparece como a raiz da árvore, a partir desta raiz são geradas ramificações, que

representam a relevância desta ligação, as quais podem também gerar outras ramificações que funcionariam da mesma forma. Para os autores, tal estrutura teria então a capacidade de representar, de forma intuitiva, padrões simples e complexos, de onde as informações poderiam ser extraídas. Como conclusão do trabalho, determinaram que os resultados obtidos foram relevantes a ponto de poderem auxiliar tanto os estudantes de medicina como candidatos a vestibulares desta área, quanto para as próprias instituições, facilitando nas tomadas de decisões e aprimoramento do curso.

[Lima et al. 2018] propõe uma metodologia para análise de provas por conhecimento de conteúdo a dois exames nacionais brasileiros: ENEM e ENADE. A metodologia se baseia em três etapas principais: Identificação/Catalogação, Agrupamento dos dados por tema, Análises. Cada uma contendo três atividades. A primeira etapa consiste na identificação dos temas, no caso do ENADE para as provas de Ciências da Computação, onze temas foram selecionados partindo de suas definições publicadas em cada ano de aplicação. No download das provas e a extração das questões, não apenas seu enunciado como também as alternativas. E por último a classificação das questões nos temas selecionados, esse processo é apontado como o que mais demanda trabalho manual, já que foi realizado a partir da leitura de cada questão e atribuição delas em um dos temas. Assim, a segunda etapa é responsável basicamente pelo cálculo de acertos por tema e a extração das respostas dos alunos ao questionário da prova, tudo isso feito a partir do download dos microdados disponibilizados pelo INEP. A terceira e última etapa é onde a análise dos dados extraídos e normalizados acontece, visando obter informações importantes, melhoria na visualização de dados e a produção de relatórios mais detalhados. A pesquisa propõe algumas técnicas que podem ser aplicadas como a Estatística Descritiva, utilizada para analisar a estrutura do exame e o resultado do aluno; Estatística Inferencial, para verificar a correlação entre variáveis, diferenças entre grupos ou diferenças em momentos distintos no tempo, e por fim, Data Mining, usado principalmente para prever resultados dos alunos nos temas definidos. Sendo assim, um protótipo foi criado visando automatizar as etapas 2 e 3, requirindo o download das fontes e a definição dos temas e dicionários de palavras para eles como trabalho manual, logo foram analisadas informações sobre estudantes de Ciência da Computação em 4 edições da prova. A conclusão tomada foi de que a metodologia apresentou flexibilidade e adaptabilidade, mostrando potencial para seu uso em outros exames, além disso apontou que a implementação de uma abordagem automática para a fase 1 precisaria do uso de outras tecnologias, inclusive a análise em textos. Por mais que exploratória, os resultados da pesquisa motivam uma aplicação mais extensiva tanto nos aspectos de implementação quanto nos possíveis resultados.

[Silva 2020] a partir do estudo de caso em provas do ENEM, objetiva identificar a arquitetura de um classificador ou um conjunto de classificadores de forma maximizar o de desempenho do processo de classificação de questões no contexto educacional. O estudo teve como base 25 mil questões pré-processadas retiradas das avaliações do ENEM até 2017, classificadas por especialistas dentro das disciplinas, competências e habilidades. Com isso foram realizados experimentos com diversos classificadores resultados da união de diferentes tipos de reapresentação de textos, cálculos de peso de termos e algoritmos de Aprendizado de Máquina Supervisionados que ao final possibilitou um comparativo com indicadores de desempenho. Assim, a ideia inicial era utilizar as habilidades do ENEM como critério para a classificação, porém a grande quantidade de classes possíveis e a distribuição irregular das questões entre as habilidades dentro

das mesmas competência impossibilitaram essa configuração. Logo, foi decidido por utilizar as 30 competências definidas para as 4 áreas do conhecimento como critério final de classificação. O modelo geral de classificação obtido se baseou na técnica Cross Industry Standard Process for Data Mining (CRISP-DM) e envolve uma série de requisitos que englobam os dados que serão classificados critérios de classificação, representação computacional dos dados, seleção das informações mais relevantes, aplicação de algoritmos classificadores, entre outros. Partindo de um baseline baseado no estado-da-arte da classificação, o trabalho concluiu que o modelo espaço-vetorial, juntamente com o algoritmo KNN e a técnica IQF * QF * ICF demostraram ser a melhor combinação de componentes para ser utilizada na classificação de questões cuja acurácia final variou de 74% a 89%.

O trabalho feito por [Araujo 2021] é o que mais se aproxima do proposto aqui, devido a base de dados utilizada. Nele são utilizadas as provas do ENADE para o curso de Ciências da Computação aplicadas entres 2005 e 2017, extraindo o texto das questões objetivas e discursivas de conteúdo específico para o curso em questão. O objetivo foi avaliar o desempenho de quatro algoritmos de classificação: Rede Neural, Naive Bayes (NB), Support Vector Machine (SVM) e Random Forest (RF). A tarefa de classificação foi feita através do Aprendizado de Máquina Supervisionado utilizando como base a classificação manual das questões em 17 temas definidos, detalhada no trabalho de [Charao et al. 2020]. Dessa forma, as etapas envolveram a extração dos dados, pré-processamento, transformação do texto e a aplicação dos algoritmos e sua avaliação. O autor seleciona três subcasos para a aplicação, no primeiro apenas os texto das questões são considerados; já no segundo o texto das 150 questões selecionadas foi utilizado para formação dos *embeddings*, junto ao CorpusTCC [Pardo and Nunes 2003] e por fim, no subcaso três a configuração é semelhante à anterior, porém foi utilizado um *corpus* personalizado, composto por 23 livros de computação, para o *embeddings*. O intuito das duas configurações extras foi aumentar a quantidade de palavras disponíveis no vocabulário e, consequentemente, melhorar os resultados obtidos com os algoritmos, no segundo subcaso o vocabulário mais que dobrou de tamanho com 2696 palavras e no terceiro mais ainda, com 18527 palavras. A conclusão foi de que o modelo utilizando Redes Neurais se mostrou o mais ineficiente nos três subcaso. Por outro lado, técnicas clássicas como SVM, NB e RF obtiveram resultados intermediários de acurácia e com aplicação simples, entre eles o modelo utilizando SVM se mostrou o de melhor desempenho com 48%, 50% e 60% de acurácia nos três subcasos, respectivamente.

Com os trabalhos citados anteriormente, é claro concluir que os dados resultantes das provas do ENADE possibilitam uma melhor compreensão dos alunos, do curso e das próprias instituições, servindo para possíveis melhorias na forma de aprender e ensinar, e até mesmo revisões mais drásticas na estrutura do curso.

3. Metodologia

A respeito da metodologia empregada, o trabalho teve início com a revisão da literatura especifica a respeito do tema da pesquisa. Sendo o foco da revisão abranger os conceitos fundamentais para o desenvolvimento do trabalho nas três áreas de conhecimentos principais: Mineração de Dados, Aprendizado de Máquina (AM) e Processamento de Linguagem Natural (PLN). Além disso, levantar os trabalhos correlatos a fim de identificar as técnicas comuns utilizadas, dificuldades encontradas e pontos de melhoria. Com relação

os dados utilizados, foram coletadas amostras de cinco anos de aplicação do ENADE para o curso de Ciências da Computação (CC). Contemplando não somente os microdados disponibilizados pelo INEP como também os Relatórios de Curso e o conteúdo íntegro da prova, questões e enunciados. Nesse sentido, a extração e tratamento dos dados foram a etapa seguinte, todos eles se encontravam na forma de documentos PDF, logo foi necessário a extração do texto contido neles. Já as bibliotecas em Python foram utilizadas para essa tarefa quando possível, porém alguns destes documentos precisaram passar por técnicas de extração de texto mais elaboradas, como OCR em softwares especializados. O tratamento do texto se deu utilizando também a linguagem de programação Python, fazendo uso de bibliotecas especializadas em NPL. Por fim, foi contemplado a escolha da implementação específica do KNN, seu uso na base de dados extraída e a avaliação dos resultados. As etapas foram divididas como ilustra a Figura 3 abaixo.

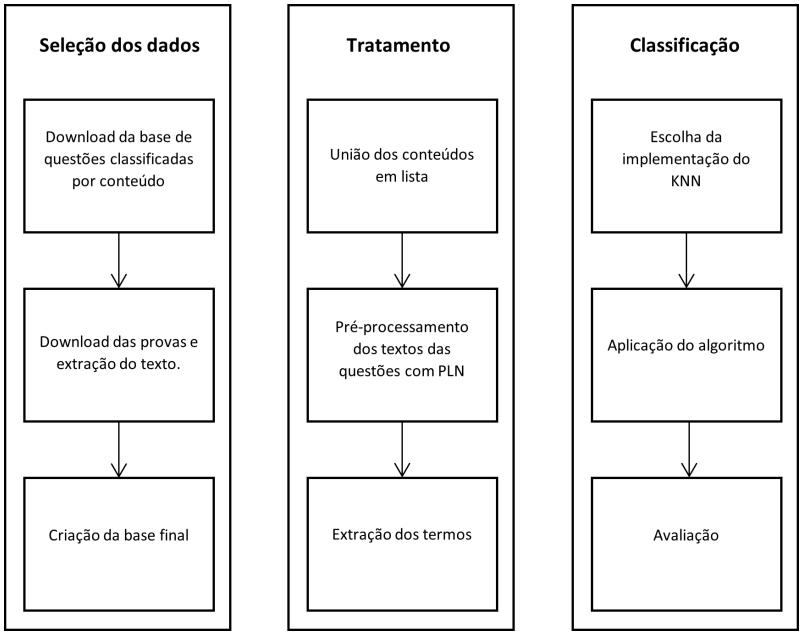

Figura 3. Etapas da metodologia aplicada (Autor)

4. Desenvolvimento

4.1. Seleção dos dados

Como comentado anteriormente, a classificação por meio do aprendizado de máquina supervisionado depende de uma base, na qual já existam objetos e suas classificações corretas para que o modelo possa usar como treinamento em suas predições. Logo, precisamos de uma base onde as questões já estejam classificadas em determinados conteúdos. Assim, com relação as questões foram utilizadas as provas de 2005 a 2017, extraindo as objetivas e discursivas do conteúdo específico para o curso de Ciências da Computação, sendo em cada uma das aplicações e no caso das questões objetivas considerados não só seus enunciados como também o texto de suas alternativas. Dessa forma, ao fim das extrações foram totalizados 150 enunciados de questões, sendo 30 de cada ano. Para a definição dos conteúdos a tarefa foi mais elaborada, embora o conteúdo programático fosse sempre divulgado nos editais, o atrelamento entre questões e conteúdos só começou

a ser feito oficialmente pelo INEP a partir da prova de 2014, sendo incluído nos Relatórios Síntese de Área, visto que seria necessária uma classificação comum para todas as provas. Foi utilizada então, a classificação feita por Charão em [Charao et al. 2020], onde os 17 conteúdos presentes nas provas de 2014 e 2017 foram usados como base para a classificação manual, feita por um especialista para o restante das questões nos anos 2005, 2008 e 2011. A Tabela 1 mostra os 17 conteúdos definidos, utilizando a sigla OC (Objeto de conhecimento), a mesma usada pelo INEP para representa-los. Portanto, a partir da relação entre questão e conteúdo disponibilizada foi possível junta-la com a base de questões, assim formando a base necessária para dar início às próximas etapas. Na relação feita por Charão cada questão pode ser atribuída a até três conteúdos. Logo, para que não fosse necessário descartar nenhum conteúdo e poder tirar o maior proveito do algoritmo, todos os conteúdos atrelados a uma determinada questão foram considerados. Nas Figuras 4 e 5 temos a relação de quantidade de questões por conteúdos, na primeira de forma individual para cada conteúdo e na segunda olhando para conjuntos de conteúdos, ou seja, por quantidade de questões com mais de um conteúdo atrelado.

OC	**Conteúdo**
OC_01	Algoritmos e Estruturas de Dados
OC_02	Arquitetura de Computadores e Sistemas Operacionais
OC_03	Banco de Dados
OC_04	Compiladores
OC_05	Computação Gráfica e Processamento de Imagem
OC_06	Engenharia de Software e Interação Homem-Computador
OC_07	Fundamentos e Técnicas de Programação
OC_08	Inteligência Artificial e Computacional
OC_09	Lógica e Matemática Discreta
OC_10	Paradigmas de Linguagens de Programação
OC_11	Probabilidade e Estatística
OC_12	Redes de computadores
OC_13	Sistemas Digitais
OC_14	Sistemas Distribuídos
OC_15	Teoria da Computação
OC_16	Teoria dos Grafos
OC_17	Ética, computador e sociedade

Tabela 1. Conteúdos utilizados para a classificação

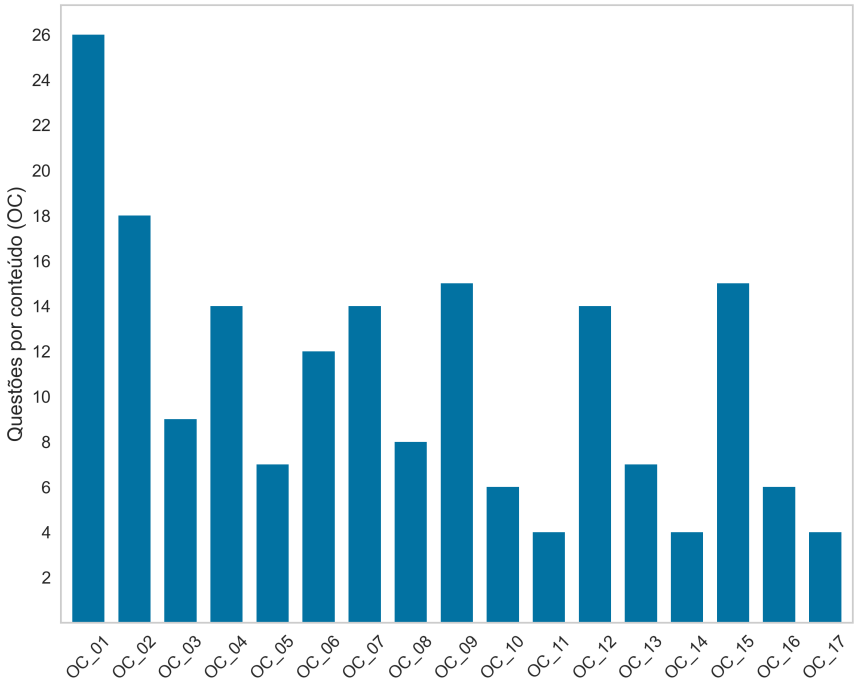

Figura 4. Distribuição de questões por conteúdo (Autor)

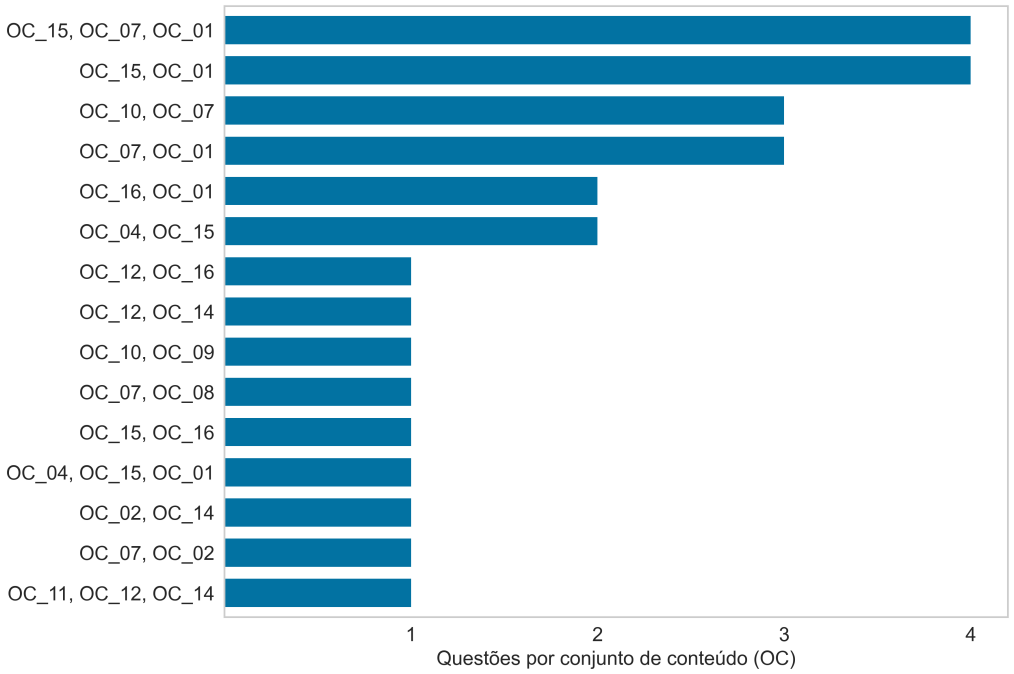

Figura 5. Distribuição de questões por conjunto conteúdo (Autor)

4.2. Tratamento

Uma vez com os arquivos brutos das provas em formato PDF, foi feita a extração de todo o texto contido nos arquivos utilizando a biblioteca PyPDF2, e por final, os enunciados das questões foram unidos à base final. É justamente no campo referente aos enunciados que o processo de pré-processamento teve foco, objetivando tanto a remoção de ruídos, provenientes da etapa de extração do texto, quanto a otimização do seu uso na etapa de classificação. Nesse sentido, o processo se iniciou com o tratamento de ruídos contemplando por exemplo, a remoção de quebras de linhas, caracteres especiais e termos muito pequenos ou muito grandes, então seguiu com as tarefas referentes ao processamento de linguagem natural utilizando a biblioteca NLTK, envolvendo a tokenização, o steamming e remoção de stopwords e pontuações. Na Figura 6 é apresentado um comparativo de quantidade de palavras por enunciado de cada questão, antes e depois do tratamento. É possível notar que após o tratamento ocorre uma diminuição de enunciado com mais de 100 palavras, justamente devido a exclusão dos ruídos mencionados anteriormente, assim fazendo com que a classificação pudesse ter foco nos termos que realmente agregam sentido ao texto. Para a transformação do texto e extração dos termos foi utilizado algoritmo TF-IDF, a implementação adotada foi a TfidfVectorizer da biblioteca Scikit-learn. Quanto aos três campos referentes aos conteúdos, esses foram unidos em forma de lista e armazenados em um só.

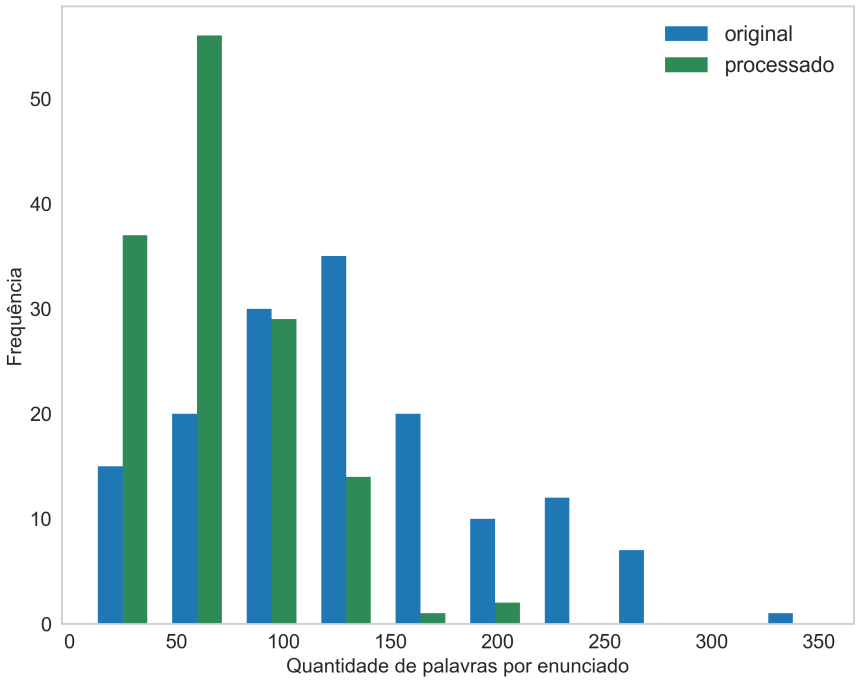

Figura 6. Quantidade de palavras por enunciado (Autor)

4.3. Classificação

Como mencionado, foi utilizado o algoritmo KNN para a tarefa de classificação, a escolha foi devida a sua fácil utilização, bom desempenho com volumes baixos de dados e a escassez de trabalhos encontrados, que exploravam seu uso junto aos dados obtidos a partir das provas do ENADE. Em relação a sua implementação, uma opção seria a própria disponibilizada pela biblioteca Scikit-learn, porém para que fosse possível trabalhar com o problema de forma *mult-label*, ou seja, utilizando o conjunto máximo de até três conteúdos por questão, seria necessário uma transformação do problema, como comentado na seção 2.5, caso contrário apenas um conteúdo teria que ser considerado da base.

Como alternativa, evitando o trabalho extra proveniente da transformação do problema, além também da escolha de apenas um conteúdo por questão, foi ,então, selecionado como implementação a adaptação MLKNN (Multilabel KNN) apresentada por Zhang e Zhou em [Zhang and Zhou 2007] , disponibilizada através da biblioteca skmultilearn. A escolha foi baseada no bom desempenho do algoritmo em lidar nativamente com problemas *multi-label*.

4.4. Resultados

Com a base de dados pronta e o algoritmo escolhido, foi feita a separação do dataset entre treino e teste, utilizando a proporção 80/20, respectivamente. Portanto, para o cálculo da acurácia foi utilizada a função accuracy_score presente no módulo de métricas da biblioteca Scikit-learn e foi observada a maior acurácia de 56% para k=3.

A escolha do k foi dada a partir da premissa de que questões atreladas a um mesmo conteúdo possuem aspectos em comum, então é seguro dizer que questões com o mesmo conjunto de conteúdo possuem ainda mais aspectos em comum. Assim, uma vez que o máximo de conteúdo para cada questão é 3, entende-se que esse seria o limite de vizinhos para cada questão no plano de decisão do algoritmo, o gráfico da Figura 8 ilustra os resultados de acurácia obtidos em relação ao k escolhido, é possivel notar que o resultado obtido com k=3 fica consideravelmente distante dos outros valores testados.

A nível de comparação foram utilizados os resultados dos algoritmos clássicos: Random Forest (RF), Naive Bayes (NB) e Support Vector Machine (SVM), atingidos por Araújo em [Araujo 2021], para o cenário contemplando apenas aos textos das 150 questões sem nenhum tipo de *embedding*. Na Figura 7, é possível notar que o MLKNN supera consideravelmente todos os três outros algoritmos.

Uma vez que o conjunto de dados utilizado foi o mesmo, assim como a técnica para transformação do texto, podemos, então, atribuir a acurácia superior a dois fatores principais. Sendo esses o bom desempenho do algoritmo KNN, lidando com conjuntos de dados consideravelmente pequenos e a característica *multi-label* da implementação MLKNN, que possibilita a descoberta de correlações entre rótulos, já que quando permitimos que um objeto se atrele a mais de um rótulo, estamos de certa forma aumentado as amostras disponíveis para esses rótulos, por exemplo o classificador pode não estar classificando corretamente entradas para os rótulos atrelados somente a A e nem aqueles atrelados somente a B, porém possa estar acertando para entradas, onde a classificação seja a combinação $\{A, B\}$.

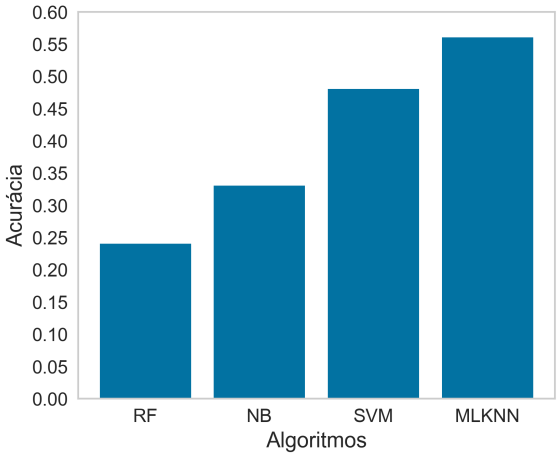

Figura 7. Comparação de acurácia (Autor)

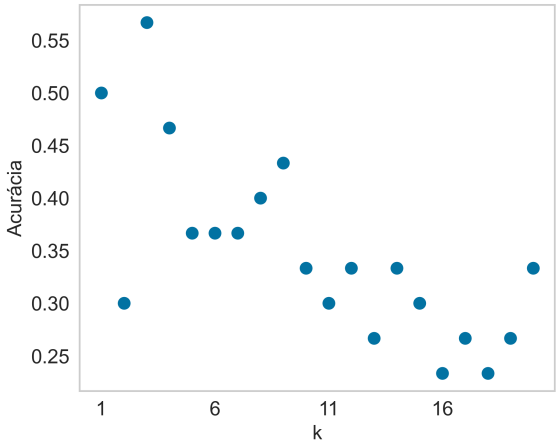

Figura 8. Acurácia por k escolhido (Autor)

5. Conclusão

O trabalho apresentou os resultados obtidos com a aplicação do classificador KNN, em uma base de questões das provas do ENADE para o curso de Ciências da Computação em 5 anos de aplicação. Dessa forma, foram apresentadas técnicas de tratamento e transformação essenciais para qualquer tarefa semelhante que envolva aprendizado de máquina e dados textuais não estruturados. Considerando a quantidade de dados disponíveis relativamente pequena no geral - apenas 150 questões - e também no escopo de cada questão, uma vez que a maioria delas não tinham um vocabulário grande mesmo antes do tratamento, os resultados do classificador foram satisfatórios, alcançando 56% de acurácia, superando algoritmos clássicos de classificação aplicados à mesma base de dados. Assim, junto a extração de termos através do algoritmo TF-IDF, a escolha do MLKNN, implementação do KNN adaptada para lidar com *multi-label*, foi o diferencial quando comparado com outros trabalhos, sendo possível considerar todos os conteúdos atrelados a cada questão, fator que contribuiu para que uma acurácia maior fosse alcançada. Implementações como o MLKNN, que exploram a característica *multi-label* do problema nativamente, são fundamentais para o avanço das técnicas de classificação de texto, especialmente quando lidamos com objetos de avaliações (provas, questionários, comentários, etc.), no qual o escopo para classificação tende a ser menor do que o usual em tarefas do gênero, logo a possibilidade de aproveitar todo o conteúdo em mãos sem a necessidade de adaptação do problema para técnicas mais tradicionais pode significar, como visto aqui, em resultados expressivamente melhores.

Um ponto de melhora está na avaliação do algoritmo, já que o método utilizado considera como uma classificação correta no caso de um problema *multi-label*, a combinação exata de *labels* presentes, ou seja, caso o modelo classificasse uma questão que tivesse três conteúdos atrelados com apenas dois deles, o retorno seria considerado errado e, consequentemente, prejudicaria a acurácia mesmo que a classificação não tenha sido totalmente errada. Portanto, a utilização de métodos de avaliação capazes de considerar tais cenários poderiam aumentar consideravelmente os resultados obtidos.

Por fim, o trabalho teve caráter exploratório e espera-se que sirva como ponto de partida ou auxílio para outras análises com dados semelhantes, visando a melhora dos resultados ou a utilização de outras técnicas e algoritmos.

Referências

Alahmadi, A. (2016). Automatic text classification using bag of words and bag of concepts based representations.

Araujo, L. R. d. (2021). Classificação automática de questões de provas: análise comparativa de algoritmos e aplicação ao enade.

Charao, A. S., Wiechork, K., Rodrigues, M. L., and Barbosa, F. P. (2020). Explorando resultados por questão no enade em ciência da computação para subsidiar revisão de projeto pedagógico de curso. In *Anais do XXVIII Workshop sobre Educação em Computação*, pages 16–20. SBC.

Cretton, N. N. and Gomes, G. R. (2016). Aplicação de técnicas de mineração de dados na base de dados do enade com enfoque nos cursos de medicina. *Acta Biomedica Brasiliensia*, 7(1):74–89.

e José Augusto Baranauskas, M. C. M. (2003). Conceitos sobre aprendizado de máquina. In *Sistemas Inteligentes Fundamentos e Aplicações*, pages 89–114. Manole Ltda, Barueri-SP, 1 edition.

Emmanuel, M., Khatri, S. M., and Babu, D. R. R. (2013). A novel scheme for term weighting in text categorization: Positive impact factor. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2292–2297.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37.

Ferreira, M. F. (2014). O curso de pedagogia: perfil de ingresso, inserção profissional e promoção social.

Katakis, I., Tsoumakas, G., and Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, page 5. Citeseer.

Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24.

Liddy, E. D. (2001). Natural language processing.

Lima, P. D. S., Ambrósio, A. P., Brancher, J. D., and Felix, I. (2018). Sysenade-análise das questões de provas do enade organizadas pelos temas abordados. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 7, page 419.

Pardo, T. A. S. and Nunes, M. d. G. V. (2003). A construção de um corpus de textos científicos em português do brasil e sua marcação retórica. Technical report, Technical Report.

Patra, A. and Singh, D. (2013). A survey report on text classification with different term weighing methods and comparison between classification algorithms. *International Journal of Computer Applications*, 75(7).

Pota, M., Fuggi, A., Esposito, M., and De Pietro, G. (2015). Extracting compact sets of features for question classification in cognitive systems: A comparative study. In *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 551–556.

Rossi, R. G. (2016). *Classificaçao automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Universidade de São Paulo.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Silva, V. d. A. (2020). *Classificação automática de questões baseada em competências: ENEM-Estudo de caso*. PhD thesis, Universidade de São Paulo.

Tan, A.-H. et al. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, volume 8, pages 65–70. Citeseer.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90.

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.