**CEFET/RJ - CENTRO FEDERAL DE EDUCACAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA**

# Avaliação de técnicas de processamento de linguagem natural e aprendizado de máquina em sistemas de avaliação automática de respostas a itens discursivos

Ramon Grande da Luz Bouças

Prof. Orientador:

Eduardo Bezerra, D.Sc.

**Rio de Janeiro,**

**Março de 2023**

**CEFET/RJ - CENTRO FEDERAL DE EDUCACAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA**

# Avaliação de técnicas de processamento de linguagem natural e aprendizado de máquina em sistemas de avaliação automática de respostas a itens discursivos

Ramon Grande da Luz Bouças

Projeto final apresentado em cumprimento às normas do Departamento de Educação Superior do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação

Prof. Orientador:
Eduardo Bezerra, D.Sc.

**Rio de Janeiro,**
**Março de 2023**

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

B753 Bouças, Ramon Grande da Luz
Avaliação de técnicas de processamento de linguagem natural e aprendizado de máquina em sistemas de avaliação automática de respostas a itens discursivos / Ramon Grande da Luz Bouças – 2023.
xvi, 70f : il. (algumas color.) + apêndices , enc.

Projeto Final (Graduação). Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2023.
Bibliografia : f. 67-70.
Orientador: Eduardo Bezerra da Silva

1. Computação 2. Aprendizado de máquina. 3. Processamento de linguagem natural. I. Silva, Eduardo Bezerra da (Orient.). II. Título.

CDD 004

Elaborada pelo bibliotecário Leandro Mota de Menezes – CRB/7 nº 5281

# DEDICATÓRIA

Dedico esse trabalho aos meus professores do Colégio Pedro II, pessoas que foram fundamentais na minha formação intelectual e acadêmica.

# AGRADECIMENTOS

Agradeço primeiramente à Deus por ter me dado a sabedoria para fazer as escolhas que fiz. Agradeço aos meus pais Cássio e Lucia Helena, por terem me guiado no caminho da verdade e da justiça. Agradeço à minha irmã Mirella, pelo amor, companheirismo e por fazer minha vida mais engraçada e alegre. Agradeço ao professor Eduardo Bezerra pela paciência e dedicação durante o processo e pelos ensinamentos e orientações, sem os quais seria impossível concluir esse trabalho. Agradeço, de forma geral, aos professores do CEFET/RJ pelos ensinamentos que me ajudaram a passar pelo curso e seguir até esse ponto. Agradeço finalmente aos meus colegas de empresa, pessoas com as quais em uma conversa aqui e outra ali me ensinaram valiosas lições sobre Ciência de Dados. Dentre esses não poderia deixar de agradecer ao Gabriel Daiha Alves, grande colega, amigo e meu primeiro mestre e mentor em minha trajetória profissional em ciência de dados e também ao grande mestre Vinícius André Velozo Lopes, que tanto vem contribuindo no meu aprimoramento técnico-científico e profissional.

"Tenho a impressão de que você acha que nossa principal missão é inventar palavras novas. Nada disso! Estamos destruindo palavras — dezenas de palavras, centenas de palavras todos os dias. Estamos reduzindo a língua ao osso. A Décima Primeira Edição não conterá uma única palavra que venha a se tornar obsoleta antes de 2050." Deu uma dentada faminta no pão e engoliu duas colheradas de ensopado, depois continuou falando, com uma espécie de paixão pedante. Seu rosto escuro e afilado se animara, seus olhos haviam perdido a expressão zombeteira e adquirido um ar quase sonhador. "Que coisa bonita, a destruição de palavras! Claro que a grande concentração de palavras inúteis está nos verbos e adjetivos, mas há centenas de substantivos que também podem ser descartados. Não só os sinônimos; os antônimos também. Afinal de contas, o que justifica a existência de uma palavra que seja simplesmente o oposto de outra? Uma palavra já contém em si mesma o seu oposto. Pense em "bom", por exemplo. Se você tem uma palavra como "bom", qual é a necessidade de uma palavra como "ruim"? "Desbom" dá conta perfeitamente do recado. É até melhor, porque é um antônimo perfeito, coisa que a outra palavra não é. Ou então, se você quiser uma versão mais intensa de "bom", qual é o sentido de dispor de uma verdadeira série de palavras imprecisas e inúteis como "excelente", "esplêndido" e todas as demais? "Maisbom" resolve o problema; ou "duplimaisbom", se quiser algo ainda mais intenso. Claro que já usamos essas formas, mas na versão final da Novafala tudo o mais desaparecerá. No fim o conceito inteiro de bondade e ruindade será coberto por apenas seis palavras —na realidade por uma palavra apenas. Você consegue ver a beleza da coisa, Winston? Claro que a ideia partiu do "g. i.", acrescentou, como alguém que se lembra de um detalhe que não havia mencionado. Uma espécie de ansiedade inconsistente perpassou o rosto de Winston ao ouvir falar no Grande Irmão. Mesmo assim, Syme detectou instantaneamente uma certa falta de entusiasmo. "Você não sente muita admiração pela Novafala, Winston", disse ele, quase triste. "Até mesmo quando escreve, continua pensando em Velhafala. Li alguns daqueles artigos que você publica no Times de vez em quando. São muito bons, mas são traduções. No fundo você preferiria continuar usando a Velhafala, com todas as suas inexatidões e nuances inúteis de significado. Não compreende a beleza da destruição de palavras. Você sabia que a Novafala é a única língua do mundo cujo vocabulário encolhe a cada ano?" Winston sabia, claro. Sorriu com simpatia — esperava —, sentindo-se inseguro quanto ao que diria, se abrisse a boca para falar. Syme arrancou com os dentes outro fragmento de pão escuro,

mastigou-o depressa e continuou: “Você não vê que a verdadeira finalidade da Novafala é estreitar o âmbito do pensamento? No fim teremos tornado o pensamento-crime literalmente impossível, já que não haverá palavras para expressá-lo. Todo conceito de que pudermos necessitar será expresso por apenas uma palavra, com significado rigidamente definido, e todos os seus significados subsidiários serão eliminados e esquecidos. Na Décima Primeira Edição já estamos quase atingindo esse objetivo. Só que o processo continuará avançando até muito depois que você e eu estivermos mortos. Menos e menos palavras a cada ano que passa, e a consciência com um alcance cada vez menor.

–(1984, George Orwell)

# RESUMO

Atualmente, exames avaliativos que envolvem questões discursivas e redações fazem parte da realidade de milhões de estudantes ao redor do mundo. O trabalho de correção de itens discursivas é algo que requer muitas horas de trabalho de mão de obra altamente qualificada. Em resposta a esse problema, existem sistemas computadorizados de avaliação automática de redações e de respostas a questões discursivas. Muitos desses sistemas utilizam abordagens estatísticas, como regressões, ou abordagens baseadas em aprendizado de máquina. O presente trabalho visa avaliar diferentes abordagens de Processamento de Linguagem Natural e de Aprendizado de Máquina para a avaliação automática de itens discursivos. Para isso, elaboramos um *pipeline* de previsão composto por um módulo de correção de erros ortográficos, um módulo de extração de *features*, e módulos de aplicação de algoritmo preditivo e avaliação. Fazemos uma avaliação comparativa de diferentes versões do *pipeline*, as quais vão se diferenciar pela técnica de representação vetorial empregada (Latent Semantic Indexing, Universal Sentence Encoder, TF-IDF, ou doc-to-vec) e também pela abordagem de previsão empregada. (classificação, regressão ou classificação ordinal)

**Palavras-chaves**: Avaliação automática de redações, Avaliação automática de respostas discursivas, Processamento de Linguagem Natural.

# ABSTRACT

Currently, exams that involve discursive questions and essays are part of the reality of millions of students around the world. The work of correcting discursive items is something that requires many hours of work by highly qualified labor. In response to this problem, there are computerized systems for automatic evaluation of essays and answers to discursive questions. Many of these systems use statistical approaches such as regressions or machine learning based approaches. The present work aims to compare natural language processing and machine learning approaches for automatic evaluation of essays. For this, we developed a evaluation pipeline composed of a spelling error correction module, a feature extraction module and modules for the application of predictive and evaluation algorithms. We make a comparative evaluation of different versions of the pipeline, which vary alongside the vector representation technique applied (Latent Semantic Indexing, Universal Sentence Encoder, TF-IDF, or doc-to-vec) and also vary by the prediction approach used (classification, regression or ordinal classification)

**Keywords**: Automatic assessment of essays, Automatic assessment of discursive responses, Natural Language Processing.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Lista de Abreviações

| | | |
|---|---|---|
| ASAP-SAS | Automated Student Assessment Prize - Short Answer Scoring | 5, 8, 40 |
| ASAP-AES | Automated Student Assessment Prize - Automated Essay Scoring | 5, 8, 22, 40, 63 |
| CBOW | Continous Bag Of Words | 14, 16, 17 |
| ENEM | *Exame Nacional do Ensino Médio* | 2, 4, 41 |
| IDF | Inverse Document Frequency | 21 |
| IMDB | Internet Movie Database | 66 |
| LSI | Latent Semantic Indexing | xv, xvi, 5, 7, 20, 22, 33, 34, 40, 48, 53, 56, 58, 59, 60, 61, 63 |
| NAEP | National Assessment Of Educational Progress | 2 |
| NLTK | *Natural Language Toolkit* | 30, 32 |
| PEG | Project Essay Grader | 4 |
| POS | Part-of-Speech | 22, 24 |
| PV-DBOW | Paragraph Vector - Distributed Bag Of Words | xiii, 17, 18, 34 |
| PV-DM | Paragraph Vector - Distributed Memory | xiii, 17, 18, 34 |
| QWK | Quadratic Weighted Cohen's Kappa | 6, 8, 9, 10, 26, 27, 38, 39, 44, 47, 56, 58, 59, 60, 63 |
| SVD | Singular Value Decomposition | 20 |
| SVR | Support Vector Regression | 22, 25 |
| TF | Term Frequency | 21 |
| TF-IDF | *Term frequency-inverse document frequency* | 5, 7, 21, 22, 33, 34, 35, 40, 52, 56, 58, 59, 60, 62, 64, 65, 66 |
| USE | Universal Sentence Encoder | 5, 12, 13, 14, 19, 33, 34, 40, 57, 58, 60, 61, 62, 63, 64, 65, 66 |

# Capítulo 1

# Introdução

## 1.1 Contextualização

Atualmente, educadores utilizam diversas técnicas para a avaliação do aprendizado de estudantes. Entre elas, encontram-se o uso de itens de múltipla escolha e itens discursivos. Dentre os itens discursivos, encontram-se os *Essays*, forma de avaliação que mostra certa proximidade ao que conhecemos no Brasil como redação e as *Short Answer*, que são equivalentes ao que conhecemos como respostas discursivas. A temática dos *Essays* pode ou não ser baseada em conteúdo de alguma disciplina. Independentemente disso, é costumeiro levar em conta, além do conteúdo do texto, aspectos relacionados à capacidade de comunicação e de raciocínio demonstrados pelos estudantes. Inclusive, há quem considere que, em *Essays* que são avaliadas sobre ambos os critérios, o estilo, ou seja as características gerais da escrita, acabam interferindo até mesmo na avaliação dos aspectos de conteúdo. Por exemplo, Shermis and Burstein [2003] apontam que na avaliação de *Essays*, avaliadores humanos são suscetíveis ao *Halo Effect*, tendência de avaliar um texto favoravelmente em todas as características, quando se tem uma boa impressão acerca do texto, e levantam a suspeita que a correlação entre as notas em estilo e conteúdo talvez sejam mais altas do que deveriam ser.

Há dois tipos de *Essay* que vale a pena destacar: o *Examination Essay* e o *Composition Essay*. O *Examination Essay* é muito presente em sistemas educacionais inspirados pelo estilo britânico e exige que os estudantes mostrem conhecimento de conteúdo acadêmico ensinado em um curso e sintetizem esse conhecimento em um espaço de tempo relativamente curto [Shermis and Burstein, 2003] apud [Brown, 2010]. O *Composition Essay*, comum no ensino americano, embora ainda avalie de alguma forma o conteúdo do texto, é mais focado no estilo e nas habilidades de escrita [Shermis and Burstein, 2003] apud [Brown, 2010]. Podemos definir esse formato de avaliação em uma máxima "learn to write and write to learn" [Brown, 2010]. Podemos então perceber pelas descrições dadas anteriormente, que os *Composition Essay* mostram bastante convergência com as redações trabalhadas na educação básica brasileira. Por outro lado, não foi possível observar equivalência entre os *Examination Essay* e alguma forma de avaliação

presente na educação básica brasileira. Portanto no escopo de nosso trabalho, não abarcaremos os *Examination Essay*, focando apenas nos *Composition Essay*. Dessa forma, a partir daqui, quando nos referirmos a *Essays* estaremos nos referindo exclusivamente aos *Composition Essays*, e o mesmo vale para quando mencionarmos o termo redação. As *Short answers*, por outro lado, têm como objetivo principal avaliar se o candidato possui conhecimento acerca de um tópico. É comum que sejam questões mais específicas e que esperam respostas curtas, por isso o nome *(Short Answers)*. Comumente, a correção utiliza um conjunto de respostas pensado no momento da elaboração do exame, prática conhecida como "gabarito". Podemos então perceber que as *Short answer* estão alinhadas com o que conhecemos no Brasil como questões discursivas.

Muitas vezes, exames de larga escala aplicam itens discursivos. No Brasil, temos o exemplo do *Exame Nacional do Ensino Médio* (ENEM), que contou com cerca de 3 milhões de inscritos em 2021[1] e que, embora seja baseado na maior parte em itens de múltipla escolha, cuja correção já é automatizada, possui um componente discursivo que é a redação. Nos Estados Unidos, podemos destacar o exemplo do National Assessment of Educational Progress (NAEP), avaliação desenvolvida em 1969 e que é a única avaliação nacionalmente representativa e continuamente aplicada dos conhecimentos de estudantes americanos. Entre as disciplinas avaliadas, encontram-se inglês, matemática, história, geografia, entre outras que costumam compor o currículo da educação básica. O público alvo é o de estudantes de 9, 13 e 17 anos. O NAEP cobre tanto redação quanto respostas discursivas[2].

A avaliação automática de itens discursivos, categoria que abarca redações e respostas discursivas, consiste no uso de programas de computador especializados com o objetivo de atribuir um conceito (nota, avaliação) a um texto escrito em um contexto educacional [Ramesh and Sanampudi, 2021]. A ideia central é criar um padrão de correção para um exame que seja facilmente escalável para avaliar um grande número de respostas. Outra função importante da avaliação automática de respostas discursivas é o papel formativo, por meio do fornecimento de conselhos sobre os itens que o estudante precisa modificar de modo a melhorar a qualidade de seu texto escrito [Shermis and Burstein, 2003].

O principal desafio na construção de modelos de avaliação automática é prever, dados um enunciado de item discursivo, uma resposta e um formato de avaliação (se a avaliação consiste

[1]Essa informação foi obtida no endereço eletrônico https://www.gov.br/inep/pt-br/assuntos/noticias/enem/74-do-inscritos-participam-do-primeiro-dia-de-enem

[2]As informações sobre o NAEP foram obtidas no endereço eletrônico https://www.nagb.gov/naep/about-naep.html

de uma ou mais notas e quais são os conceitos possíveis), qual seria a avaliação atribuída por um avaliador humano à resposta.

Existem abordagens de avaliação automática, tanto de redações, como de respostas discursivas que são baseadas em técnicas de aprendizado de máquina [Ramesh and Sanampudi, 2021]. Cabe destaque às abordagens baseadas em aprendizado supervisionado, nas quais são usados algoritmos que treinam com um conjunto de textos pré-avaliado por avaliadores humanos e buscam entender como as características do texto influenciam a nota atribuída.

Como exemplos de conjuntos de dados que podem ser usados para treinamento de modelos de avaliação de itens discursivos, temos os conjuntos fornecidos na competição *Automated Student Assessment Prize*, promovida no ano de 2012 pela Fundação Hewlett[3] por meio do portal de competições de ciência de dados Kaggle[4].

O conjunto de textos de redações é composto por vários subconjuntos, os quais estão associados a diferentes enunciados, possuem diferentes escalas de avaliação e possuem também diferentes regras para a formação da nota final. Nesse conjunto existem, por exemplo, redações com tema e redações baseadas em textos de apoio. Na maioria dos casos, o objetivo é prever o *Resolved Score*, que é a nota final da redação (As regras utilizadas em cada conjunto para o cálculo da nota final a partir das notas dos avaliadores são apresentados no Apêndice F). Trata-se, portanto, de uma tarefa de regressão. Contudo, existe um subconjunto de textos (subconjunto 2) em que a nota é dividida em dois domínios, os quais são chamados *Writing application*, que pode ser resumido como a propriedade de a redação do aluno ser um texto coerente, coeso e aderente ao tema proposto e *Language Conventions*, que pode ser entendido como a propriedade relacionada ao conhecimento e emprego adequado da norma culta. Portanto, para esse conjunto, há dois objetivos de previsão: a nota obtida no domínio *Writing application* e a nota obtida no domínio *Language Conventions*. Os dados disponíveis para a realização desse desafio são dados tabulares em que cada entrada do conjunto de textos descreve uma redação e as informações fornecidas são o número que os organizadores da competição atribuíram à proposta da redação à qual o texto está associado, as notas dadas pelo primeiro e segundo avaliador em cada um dos dois domínios, o texto da redação e a composição da nota de acordo com características chamadas de *trait score*. Um exemplo representativo desse conjunto de textos é apresentado no Apêndice B.

[3]https://hewlett.org

[4]O conjunto de textos para redações pode ser encontrado em https://www.kaggle.com/competitions/asap-aes/data, enquanto o conjunto de textos para respostas discursivas pode ser encontrado em https://www.kaggle.com/competitions/asap-sas/data

O conjunto de textos de respostas discursivas também é composto por vários subconjuntos de textos com diferentes enunciados, temas e escalas de avaliação. Diferentemente do conjunto de redações, nesse conjunto, a nota final é sempre a nota do primeiro avaliador. Nesse conjunto existem itens cuja resposta depende de um texto (Formato que mostra semelhança ao modelo utilizado no ENEM) e itens cuja resposta independe de algum texto base. Os dados disponíveis para a realização desse desafio são dados tabulares em que cada entrada do conjunto de textos descreve uma resposta discursiva e as informações fornecidas são o número associado ao enunciado de item ao qual o texto está associado, e as notas atribuídas pelo primeiro e segundo avaliador. O objetivo de previsão é determinar a nota atribuída pelo primeiro avaliador. Um dos elementos desse conjunto de textos é apresentado no Apêndice C.

## 1.2 Motivação

Um problema da aplicação de exames discursivos é que sua correção, comparada à de exames de múltipla escolha, apresenta um custo bastante elevado, dado que a avaliação por humanos é uma tarefa que demanda grande quantidade de horas de trabalho de mão de obra altamente qualificada. Nesse sentido, desde 1966 com o Project Essay Grader (PEG) [Ajay HB, 1973], têm surgido diversas abordagens de avaliação e correção automática de itens discursivos [Ramesh and Sanampudi, 2021]. Essas iniciativas visam diminuir tal custo e minimizar potenciais vieses associados aos avaliadores humanos.

A avaliação automática de itens discursivos é uma tarefa que não pode ser executada com o simples uso de linguagens de programação ou ainda com técnicas como *pattern matching* ou ainda *language processing* [Ramesh and Sanampudi, 2021], pois tais técnicas não endereçam o problema de capturar um mesmo conceito sendo expresso por meio de diferentes escolhas de palavras. Sendo assim, existe o desafio de avaliar todas as possíveis respostas para um item, independente da forma (escolha de palavras) com que essas repostas foram construídas.

## 1.3 Objetivos

O presente trabalho tem como objetivo geral investigar o uso de diferentes técnicas de processamento de linguagem natural e de aprendizado de máquina para realizar a avaliação automática de redações e respostas discursivas. Como objetivos específicos, temos os seguintes:

- Avaliar e comparar a aplicação de diferentes técnicas de representação vetorial de texto

(LSI, Universal Sentence Encoder (USE), *Doc2vec* e *term frequency-inverse document frequency* (TF-IDF)) em um *pipeline* de avaliação automática de redações.

- Avaliar e comparar a aplicação de diferentes técnicas de representação vetorial de texto (LSI, USE, *Doc2vec* e TF-IDF) em um *pipeline* de avaliação automática de respostas discursivas.

- Avaliar e comparar a aplicação de diferentes abordagens de previsão (Regressão, Classificação e Classificação ordinal) em um *pipeline* de avaliação automática de redações.

- Avaliar e comparar a aplicação de diferentes abordagens de previsão (Regressão, Classificação e Classificação ordinal) em um *pipeline* de avaliação automática de respostas discursivas.

## 1.4 Metodologia

As competições Automated Student Assessment Prize - Automated Essay Scoring (ASAP-AES) e Automated Student Assessment Prize - Short Answer Scoring (ASAP-SAS) fornecem conjuntos de dados, para que os competidores treinem modelos de aprendizado de máquina. Nesse trabalho, a etapa de coleta dos dados foi realizada manualmente. Fizemos o *download* dos conjuntos de dados fornecidos, os quais vêm estruturados com uma coluna com o texto do item discursivo (redação ou resposta discursiva, dependendo da competição) e colunas informando as notas atribuídas. Escolhemos utilizar os conjuntos de dados presentes nessas competições, para experimentar as abordagens que comparamos, pois são conjuntos que contêm textos escritos em resposta a enunciados bastante diversos. Além disso, por serem conjuntos usados em competições, existem outros trabalhos que podem ser usados para comparação. Em nosso sistema, a etapa de preprocessamento é composta por um módulo que realiza a correção de palavras escritas erroneamente e por um módulo que gera algumas *features* mais básicas como contagem de sentenças, contagem de palavras e de palavras únicas, além de realizar outros preprocessamentos. Há também um módulo que gera características a partir do texto das respostas discursivas, em que testamos diferentes técnicas de representação vetorial de texto como *Doc2vec*, *Universal Sentence Encoder*, LSI e TF-IDF.

Para cada técnica de representação vetorial aplicada, utilizamos o resultado para a aplicação de diferentes abordagens de previsão (regressão, classificação, e classificação ordinal), em todas essas abordagens o algoritmo de aprendizado de máquina subjacente usado é o algoritmo de

árvores de decisão aleatórias *(Random Forest)*, em sua versão de classificação ou de regressão, conforme o caso. Para as versões do pipeline utilizando diferentes técnicas de representação vetorial e diferentes abordagens de previsão, dividimos os conjuntos de textos em treino e teste. Usando apenas o conjunto de treino, otimizamos os hiperparâmetros dos algoritmos por meio da abordagem de *Grid Search* com validação cruzada sobre a métrica de erro apropriada. Selecionado o conjunto de hiperparâmetros que apresentou melhor desempenho preditivo no conjunto de treino, o algoritmo realiza uma previsão com o conjunto de teste e a avaliação dessa previsão é feita utilizando o Quadratic weighted Cohen's Kappa (QWK), métrica utilizada pelas competições mencionadas.

## 1.5 Organização dos capítulos

Além desta introdução, o trabalho se divide em mais quatro outros capítulos. O Capítulo 2 apresenta a fundamentação teórica e os trabalhos relacionados. Detalhamos a metodologia proposta para realizar comparações no Capítulo 3. O Capítulo 4 exibe os resultados alcançados na Avaliação Experimental. O Capítulo 5 apresenta as considerações finais e as oportunidades de trabalhos futuros. O Apêndice A fornece uma descrição de características das respostas discursivas. O Apêndice B fornece um exemplo representativo de redação. O Apêndice C fornece um exemplo representativo de resposta discursiva. O Apêndice D traz uma visão acerca da relação entre quantidade de palavras, quantidade de sentenças e nota no conjunto de redações. O Apêndice E traz uma visão acerca da distribuição das palavras entre os diferentes conceitos no conjunto de respostas discursivas. Finalmente, o Apêndice F mostra as regras de formação das notas nos diferentes conjuntos.

# Capítulo 2

# Fundamentação Teórica

Esse capítulo se destina à apresentação da fundamentação teórica e dos trabalhos relacionados. Na Seção 2.1 é apresentado o conceito de avaliação automática de itens discursivos. Na Seção 2.2 tratamos sobre a abordagem de previsão classificação ordinal. Na Seção 2.3 tratamos sobre o conceito de aprendizado por transferência, e na Seção 2.4 tratamos sobre o conceito de aprendizado multitarefa. Na Seção 2.5 tratamos sobre *text embedding*. Na Seção 2.6 descrevemos a técnica LSI. Na Seção 2.7 descrevemos a técnica TF-IDF. Finalmente, na Seção 2.8, apresentamos os trabalhos relacionados.

## 2.1 Avaliação automática de itens discursivos

A Avaliação automática de itens discursivos consiste na tarefa de criar sistemas de avaliação computadorizada que atribuem notas ou conceitos baseados em características de um texto dado como resposta para um item avaliativo [Ramesh and Sanampudi, 2021]. O principal contexto de aplicação é o educacional, seja para avaliar de forma eficiente grandes quantidades de estudantes, ou ainda como ferramenta didática voltada a fornecer conselhos aos estudantes sobre como melhorar a escrita [Shermis and Burstein, 2003]. Uma característica da avaliação automática de itens discursivos é automatizar os processos e critérios já praticados por avaliadores humanos. A premissa, portanto é que os critérios utilizados por avaliadores humanos são adequados. Nesse sentido, a estratégia utilizada é formular o problema da automatização da avaliação como um problema de previsão em que a variável alvo é a nota atribuída a um texto. De forma geral, sistemas de avaliação automática, possuem um primeiro módulo que realiza alguns pré-processamentos básicos no texto, entre os quais estão, a correção de erros ortográficos, a remoção de *stopwords* (palavras que atuam como conectivos, não fornecendo sentido ao texto) e o *stemming*, processo de remoção de sufixos, que visa manter somente a palavra raiz, na qual reside a maior parte do sentido da palavra original. Além disso, esses sistemas costumam possuir um módulo que extrai características do texto, e um módulo de treino, que vai treinar modelos de aprendizado de máquina com os textos e a nota atribuída à eles. Finalmente, esses

sistemas possuem um módulo de previsão, o qual utiliza o modelo treinado para atribuir, de fato a nota à redação. As duas vertentes do campo de avaliação de itens discursivos abordadas em nosso trabalho são a avaliação automática de redações e a avaliação de respostas discursivas. A avaliação de redações é baseada em itens como coesão, coerência e uso da norma culta, características que representam a qualidade do texto. Na avaliação das respostas discursivas, não há tanta preocupação com a qualidade do texto. Espera-se que sejam relativamente sucintas e evoquem respostas específicas, muitas vezes restritas a termos ou conceitos [Ramachandran et al., 2015].

### 2.1.1 QWK - Quadratic weighted cohen's kappa

A métrica utilizada nas competições ASAP-AES e ASAP-SAS do *Kaggle*, das quais obtivemos nossos dados, é a métrica QWK. O QWK é uma métrica que julga a concordância entre dois avaliadores. Para calcular essa métrica, precisamos de dois vetores com o mesmo tamanho (correspondentes a dois avaliadores dando nota aos mesmos elementos). Primeiramente, construímos uma matriz histograma denotada $O$, de modo que cada valor $O_{ij}$ corresponda à quantidade de elementos que receberam o valor $i$ em uma das avaliações e o valor $j$ na outra avaliação.

Construímos também uma matriz $w$ de pesos, que recebe um valor baseado na diferença entre as duas avaliações. Nessa matriz, cada valor $w_{ij}$ é calculado de acordo com a Equação 2.1, em que $N$ representa o total de elementos submetidos à avaliação por ambos avaliadores.

$$w_{ij} = (i-j)^2/(N-1)^2 \tag{2.1}$$

Construímos ainda uma terceira matriz $E$, que é uma matriz calculada sob a premissa de que não há correlação entre os dois avaliadores. O cálculo dessa matriz é feito por meio do produto vetorial entre os dois vetores-histograma correspondentes a cada um dos avaliadores, como mostrado na Equação 2.2, em que $a_1$ corresponde ao vetor-histograma das notas atribuídas pelo primeiro avaliador e $a_2$ corresponde ao vetor-histograma das notas atribuídas pelo segundo avaliador.

$$E = \vec{a_1} \times \vec{a_2} \tag{2.2}$$

Considerando essas três matrizes e realizando uma normalização na matriz $E$, para que esta tenha soma de valores igual à da matriz $O$, podemos obter o QWK por meio da aplicação da

Equação 2.3:

$$\kappa = 1 - \frac{\sum_{ij} w_{ij} O_{ij}}{\sum_{ij} w_{ij} E_{ij}} \tag{2.3}$$

Para facilitar o entendimento, na Tabela 2.1 apresentamos um exemplo fictício de avaliações para um conjunto de redações e utilizamos esse exemplo para explicar o cálculo da métrica QWK.

**Tabela 2.1:** A Tabela apresenta exemplos fictícios de avaliações para ilustrar o cálculo do QWK.

| Redação | Avaliador 1 | Avaliador 2 |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 1 | 1 |
| 5 | 2 | 2 |
| 6 | 3 | 3 |
| 7 | 3 | 1 |
| 8 | 1 | 3 |
| 9 | 1 | 2 |

Na Equação 2.4, montamos a matriz $O$ para nosso exemplo.

$$O = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix} \tag{2.4}$$

Na Equação 2.5, usamos a Equação 2.1 para montar a matriz de pesos $w$ para nosso exemplo.

$$W = \begin{bmatrix} 0 & 1/64 & 1/16 \\ 1/64 & 0 & 1/64 \\ 1/16 & 1/64 & 0 \end{bmatrix} \tag{2.5}$$

Para montar a matriz $E$ para nosso exemplo, precisamos dos vetores histograma $a_1$ (Equação 2.6) e $a_2$ (Equação 2.7).

$$a_1 = \begin{bmatrix} 4 & 2 & 3 \end{bmatrix} \tag{2.6}$$

$$a_2 = \begin{bmatrix} 3 & 3 & 3 \end{bmatrix} \tag{2.7}$$

Na Equação 2.8 usamos a Equação 2.2 para calcular a matriz $E$ a partir de $a_1$ (Equação 2.6) e $a_2$(Equação 2.7).

$$E = \begin{bmatrix} 4 & 2 & 3 \end{bmatrix} \times \begin{bmatrix} 3 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 12 & 12 & 12 \\ 6 & 6 & 6 \\ 9 & 9 & 9 \end{bmatrix} \tag{2.8}$$

É necessário fazer ainda, uma normalização, de modo que a soma dos elementos de $E$ (Equação 2.8) seja igual à soma dos elementos de $o$ (Equação 2.4). A Equação 2.9 mostra a matriz $E$ após a normalização:

$$E = \begin{bmatrix} 4/3 & 4/3 & 4/3 \\ 2/3 & 2/3 & 2/3 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.9}$$

Considerando as matrizes $O$, (Equação 2.4), $w$ (Equação 2.5) e $E$, (Equação 2.9) e aplicando a Equação 2.3, obtemos o valor do QWK[1]: 0,307.

## 2.2 Classificação ordinal

No presente trabalho, além das abordagens já mencionadas de regressão e classificação, uma das abordagens para enquadrar o problema que será avaliada é a abordagem de classificação ordinal. Algoritmos de classificação funcionam mapeando um conjunto de atributos para uma variável alvo categórica. Aplicações práticas de aprendizado de máquina frequentemente envolvem situações em que se apresenta uma ordem entre as diferentes categorias. Apesar disso, algoritmos de classificação padrão não conseguem se beneficiar da informação de ordem entre as diferentes categorias [Frank and Hall, 2001].

Estatísticos costumam diferenciar variáveis categóricas em 2 tipos básicos: variáveis nomi-

[1] A explicação sobre o funcionamento da métrica QWK foi adaptada da que se encontra no site da própria competição `https://www.kaggle.com/competitions/asap-aes/overview/evaluation`

nais e variáveis ordinais, sendo as variáveis ordinais aquelas em que há uma relação de ordem aparente dentro do contexto. Um exemplo de variável ordinal é, por exemplo, uma avaliação de qualidade, em que fica claro que 'bom' é superior a 'médio' que é superior a 'ruim'. Outro tipo de variável que vale a pena destacar são as variáveis intervalares, aquelas que são espaçadas em intervalos iguais, também conhecidas como variáveis discretas, podemos dar como exemplo a escala de dor de 1 a 10, espaçada em intervalos de tamanho 1, usada em triagens médicas.

Os algoritmos de classificação comuns não são capazes de considerar essa informação e consideram que errar prevendo bom como ruim tem impacto igual a errar prevendo bom como médio. Situação equivalente acontece nas variáveis intervalares, em que os algoritmos de classificação comuns consideram que errar prevendo 2 como 8 tem impacto igual a errar prevendo 2 como 3. Uma abordagem alternativa ao uso de algoritmos de classificação é o uso de algoritmos de regressão, situação que é relativamente simples para o caso das variáveis intervalares. Por outro lado, para problemas com variáveis puramente ordinais aplicar a regressão exige uma interpretação *ad-hoc*, normalmente envolvendo uma conversão do valor contínuo previsto pela regressão para a variável ordinal do problema de interesse, conversão que costuma ser baseada em extenso conhecimento acerca do domínio.

Mesmo em problemas envolvendo variáveis intervalares, o mapeamento dos valores contínuos previstos pela regressão para valores discretos, ainda depende de decisão do pesquisador. No presente trabalho, uma das abordagens testadas é prever a nota usando um algoritmo de regressão e tratar valores não inteiros aproximando-os para o inteiro mais próximo. Não há motivo em particular para procedermos dessa forma, poderíamos igualmente ter tomado a decisão de sempre aproximar para o inteiro inferior (truncar) ou para o inteiro superior (arredondar).

Frank and Hall [2001] apresentam uma técnica simples que permite ao algoritmo de aprendizagem usado levar em conta a ordem dos valores da classe. Primeiramente os dados são transformados de um problema de classificação ordinal com $K$ classes para $K-1$ problemas de classificação binária, A ideia é converter um atributo ordinal $A*$ com valores ordenados $V_1, V_2, \ldots, V_k$ para $k-1$ atributos binários. Um atributo binário para cada um dos primeiros $k-1$ atributos originais, em que o $i$-ésimo atributo binário representa o teste $A* > V_i$

Em tempo de treinamento, o conjunto de dados original é transformado em $k-1$ conjuntos de dados, que irão possuir as mesmas variáveis preditoras, mas possuirão variáveis alvo diferentes, da mesma forma, serão treinados $k-1$ modelos, um para cada tarefa. Em tempo de inferência, para cada instância, a probabilidade de pertencimento a cada classe é calculada da

seguinte forma mostrada nas Equações 2.10, 2.11 e 2.12:

$$\Pr(V_1) = 1 - \Pr(Y > V1) \tag{2.10}$$

$$\Pr(V_i) = \Pr(Y > V_{i-1}) - Pr(Y > V_i), 1 < i < k \tag{2.11}$$

$$\Pr(V_k) = \Pr(Y > V_{k-1}) \tag{2.12}$$

Calculadas as probabilidades de pertencimento da instância a cada classe, podemos atribuir o exemplo à classe com maior probabilidade.

## 2.3 Aprendizado por transferência

Técnicas de *Text-Embedding*, como o *Doc2vec* e o USE, utilizadas nesse trabalho para fornecer representação vetorial aos textos utilizam o conceito de aprendizado por transferência, Inspirado no funcionamento do cérebro humano de transferir o conhecimento entre domínios o aprendizado por transferência busca usar conhecimento de um domínio-fonte *(source domain)* para melhorar o desempenho ou diminuir o número de exemplos necessários para o treinamento na tarefa alvo *(target domain)* [Zhuang et al., 2019]. É importante destacar que nem sempre a transferência do aprendizado será bem-sucedida, por exemplo, aprender a andar de bicicleta não ajuda a aprender a tocar piano mais rápido. Para continuar discorrendo sobre aprendizado por transferência, apresentamos as definições *domínio*, *aprendizado por transferência* e *tarefa*. As definições a seguir são adaptadas de Zhuang et al. [2019].

**Definição 1.** *Domínio: Um domínio D é composto por duas partes: Um espaço de variáveis* $\chi$ *e uma distribuição marginal P(X), em outras palavras* $D = (X, P(X))$. *O simbolo X denota um conjunto de instâncias, que é definido como* $X = (x \mid x_i \in \chi, i = 1, 2, \ldots, n)$.

**Definição 2.** *Tarefa: Uma tarefa T consiste em um espaço de rótulos (labels) Y e uma função de decisão* $f$, *isto é* $T = (Y, f)$ *A função de decisão* $f$ *é uma função implícita, que deve ser aprendida a partir do conjunto de dados disponível.*

**Definição 3.** *Aprendizado por transferência: Dadas algumas observações correspondentes a uma quantidade* $m_s$ *de domínios fonte e tarefas fonte e uma quantidade* $m_t$ *de tarefas alvo, o aprendizado por transferência utiliza o conhecimento implícito nos domínios fonte, para melhorar o desempenho das funções de decisão aprendidas* $f_j$ *aprendidas em que* $(j = 1, 2, 3, \ldots, m_t)$.

Quando $m_s = 1$ temos o aprendizado de transferência de fonte única *(single source)*, além disso é possível ter $m_t = 1$, ou seja o aprendizado com uma única tarefa alvo, o qual é o foco da maioria dos estudos, foco maior ainda tem o cenário em que $m_s = m_t = 1$, ou seja há uma única tarefa fonte e uma única tarefa alvo. É possível ainda, dividir o aprendizado por transferência entre aprendizado por transferência homogêneo e heterogêneo. No homogêneo, o espaço de variáveis da tarefa fonte é o mesmo da tarefa alvo, os domínios diferem apenas em distribuições marginais, enquanto no heterogêneo, os domínios diferem também no espaço de *features*.

## 2.4 Aprendizado Multitarefa

Além do aprendizado por transferência, o USE usa também o conceito de aprendizado multitarefa. A ideia do aprendizado multitarefa *(multitask learning)* é de aprender em conjunto um grupo de tarefas relacionadas. De forma mais específica, o aprendizado multitarefa reforça cada tarefa tomando vantagem das interconexões entre as tarefas, ou seja considerando tanto as semelhanças, quanto as diferenças entre as tarefas, e dessa forma a generalização de cada tarefa é aumentada [Zhuang et al., 2019].

A principal diferença entre o aprendizado por transferência e o aprendizado multitarefa é que o primeiro transfere o conhecimento de um domínio fonte para um domínio alvo, enquanto o aprendizado multitarefa transfere o conhecimento ao aprender de forma simultânea algumas tarefas relacionadas. Podemos sumarizar dizendo que enquanto o aprendizado por transferência foca na tarefa alvo, o aprendizado multitarefa trata de forma equivalente cada uma das tarefas que estão sendo otimizadas em conjunto [Zhuang et al., 2019]. É possível utilizar conjuntamente o aprendizado por transferência e o aprendizado multitarefa, e traremos um exemplo disso na Seção 2.5.3.

## 2.5 Text Embeddings

Muitos sistemas de processamento de linguagem natural trabalham com o simples mapeamento de palavras para índices em um vocabulário e, por isso, tratam palavras como unidades atômicas, sem levar em conta a similaridade entre elas [Mikolov et al., 2013]. Nesse sentido, existe a necessidade de técnicas que consigam mapear elementos mais complexos como sentenças, ou até mesmo textos, para uma representação vetorial que leve em conta a similaridade entre as palavras. As técnicas de *Text embedding* descritas nesse trabalho podem ser descritas

como a aplicação de uma rede neural artificial que é otimizada para uma tarefa (ou mais de uma tarefa ao mesmo tempo), visando gerar uma representação vetorial dos textos, que possa ser usada para aumentar o desempenho preditivo de tarefas diversas daquelas foram utilizadas para gerar a representação vetorial. Essas técnicas de *Text embedding* estão inseridas em um contexto mais geral de aprendizado por transferência *(transfer learning)*, pois utilizam tarefas para aprender representações vetoriais de textos que são utilizadas para apoiar outras tarefas, realizando assim, a transferência do aprendizado. Nesta Seção, apresentamos o *Doc2vec* e o USE, técnicas de *text-embedding* utilizadas nesse trabalho. Antes de apresentar as técnicas utilzadas, apresentaremos o *Word2vec*, técnica cuja compreensão auxiliará muito no entendimento do *Doc2vec*.

### 2.5.1 Word2vec

A metodologia proposta em Mikolov et al. [2013] apresenta-se em duas arquiteturas chamadas Continous Bag of Words (CBOW) e *Skip-gram*, cuja comparação podemos ver na Figura 2.1. Ambas as arquiteturas constroem um vocabulário $V$ de palavras (em que podem ser utilizadas todas as palavras presentes no conjunto de textos utilizados para o treino, ou apenas palavras que tenham frequência acima de um determinado limiar). O mapeamento de cada palavra para um índice no vocabulário é feito por meio de uma técnica assemelhada ao *one-hot–encoding*. Por exemplo, se 'Elephant' é a 4560 palavra em um vocabulário de 10000 palavras 'Elephant' será representada por um vetor que terá 0 em todas as posições e 1 na 4560 posição.

#### Continuous bag of words (CBOW)

Nessa arquitetura, a tarefa utilizada para otimizar o *word embedding* é a seguinte. Considere um trecho de texto como uma sequência de palavras:

$$(y_1, y_2, \ldots, y_i, y_{i+1}, \ldots, y_n)$$

Para essa sequência, o modelo tentará prever a palavra $y_i$ baseado em $n$ palavras anteriores e $n$ Palavras posteriores ($n$ é um hiperparâmetro que pode ser tunado).

**Skip-Gram**

Na arquitetura continuous skip-gram model, a tarefa consiste em: considere novamente um trecho de texto como uma sequência de palavras:

$$(y_1, y_2, \ldots, y_i, y_{i+1}, \ldots, y_n)$$

O algoritmo considera uma distância máxima $C$ ( $C$ é um hiperparâmetro), passando a olhar apenas as palavras distantes no máximo $C$ palavras de $y_i$, ou seja:

$$(y_{i-c}, y_{i-2}, y_{i-1}, \ldots, y_i, y_{i+1}, \ldots, y_{i+c})$$

Para cada palavra do conjunto de treino, é selecionado aleatoriamente um valor R, compreendido entre 1 e C, e são selecionadas R palavras no histórico (palavras anteriores a $y_i$) e R palavras no futuro (palavras posteriores a $y_i$), vale esclarecer que as R palavras no histórico e no futuro não são necessariamente as R palavras imediatamente anteriores ou posteriores a $y_i$, elas são selecionadas aleatoriamente entre as palavras definidas pela vizinhança $C$, embora a amostragem do algoritmo seja propositalmente enviesada para selecionar com menos frequência palavras mais distantes de $y_i$ [Mikolov et al., 2013]. Considerando tudo isso, o modelo tenta prever as 2R palavras ($R$ do histórico e $R$ do futuro) a partir da palavra $y_i$.

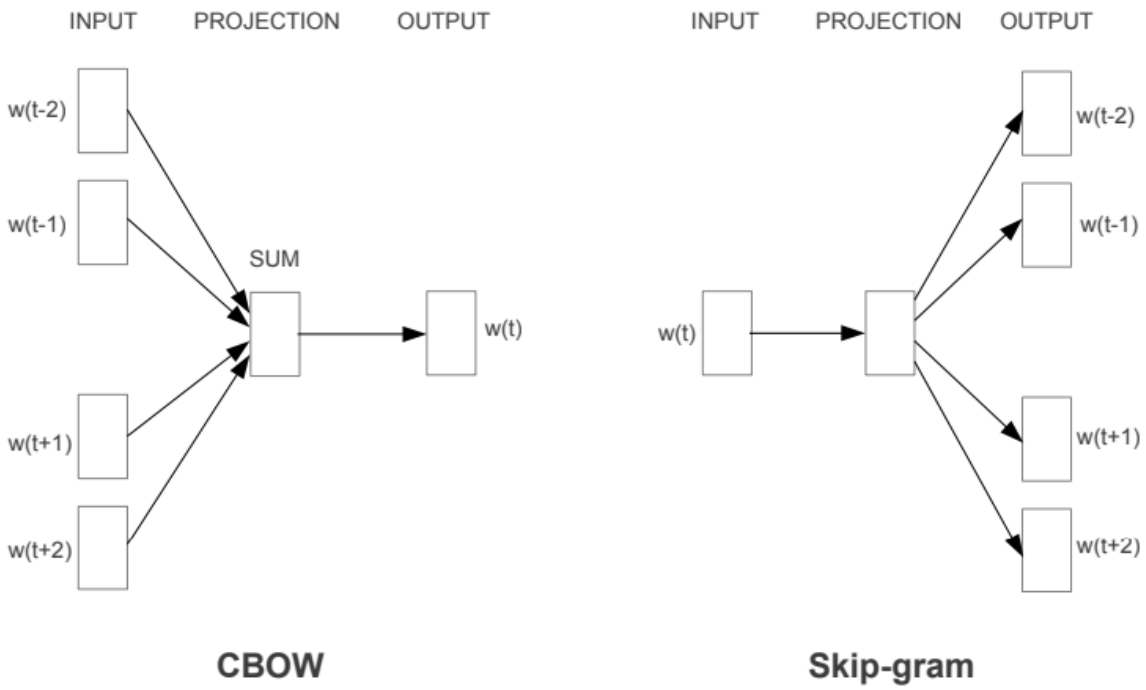



**Figura 2.1:** Comparação entre CBOW e *Skip-Gram*, Figura retirada de Mikolov et al. [2013].

### 2.5.2 Doc2vec

A ideia do *Doc2vec* é criar representações vetoriais para sequências de texto de qualquer tamanho, desde frases, até documentos inteiros. Cabe pontuar que a técnica trata da mesma forma textos independente do tamanho, desde simples frases até livros de centenas de páginas. Assim como o *Word2vec*, o *Doc2vec* se apresenta em duas arquiteturas: PV-DM e PV-DBOW. Semelhantemente ao *Word2vec*, o *Doc2vec* constrói um vocabulário $V$ de palavras mapeando cada palavra para um índice no vocabulário, criando o que vamos chamar de *word vector*. A inovação do *Doc2vec* é trabalhar com um conjunto de documentos e mapear cada documento para um vetor, chamado *document vector* usando técnica análoga à usada no *Word2vec*. Por exemplo, se a frase *'There are many beautiful elephants in India'* é a 2370 frase em um vocabulário de 5000 documentos, *'There are many beatiful elephants in india'* será representada por um vetor que terá 0 em todas as posições e 1 na 2370 posição.

A Figura 2.2 mostra a arquitetura PV-DM, a qual é análoga à arquitetura CBOW. Na arquitetura PV-DM, para uma sequência de palavras:

$$(y_1, y_2, \ldots, y_i, y_{i+1}, \ldots, y_{2n})$$

O algoritmo tentará prever a palavra $y_i$ baseado em $n$ palavras anteriores e $n$ Palavras posteriores ($n$ é um hiperparâmetro que pode ser tunado). O que diferencia o PV-DM do CBOW é o fato de que no PV-DM, além dos *word-vectors* o *document vector* também é usado na camada de entrada.

A Figura 2.3 apresenta o PV-DBOW, arquitetura que é análoga à arquitetura Skip-gram, mas que ao invés do *word-vector*, utiliza o *paragraph vector*. Para cada documento do conjunto de treino, o algoritmo vai selecionar uma janela de texto e realizar uma tarefa de classificação, objetivando prever quais palavras estão contidas nessa janela de texto.

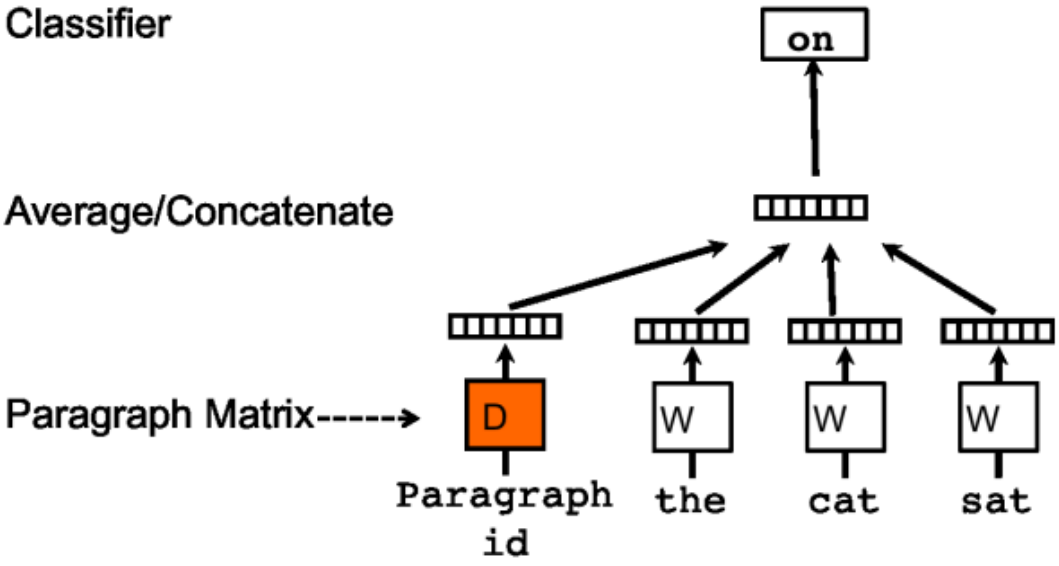



**Figura 2.2:** PV-DM, uma das arquiteturas possíveis para a implementação do *Doc2vec*. Figura retirada de Le and Mikolov [2014].

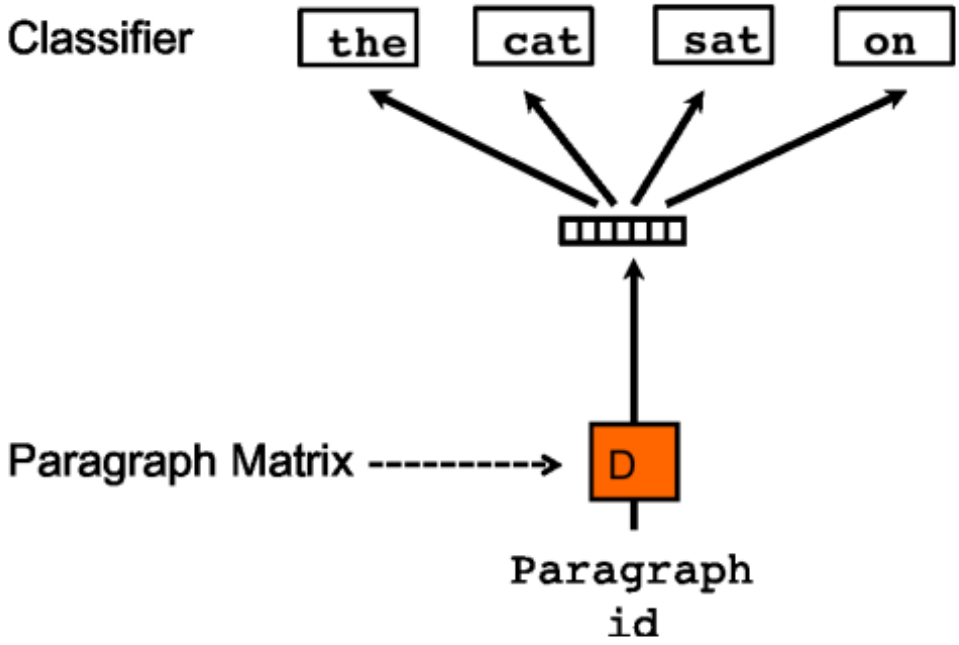



**Figura 2.3:** PV-DBOW, uma das arquiteturas possíveis para a implementação do *Doc2vec*. Figura retirada de Le and Mikolov [2014].

### 2.5.3 Universal Sentence Encoder

Os modelos descritos anteriormente otimizam o *text embedding* com base em uma única tarefa, porém o modelo de *text embedding* produzido poderá ser utilizado em aplicações diversas de processamento de linguagem natural. Dentro desse cenário, Cer et al. [2018] apresentam abordagens que além de utilizar o conceito de aprendizado por transferência, utilizam o conceito de aprendizado multitarefa (*multi-task learning*). A ideia por trás do uso desse conceito é produzir *text embeddings* que possuam melhor generalização em diferentes tarefas para as quais sejam realizadas a transferência do aprendizado. O USE se apresenta em duas arquiteturas: uma baseada em *Transformer* [Vaswani et al., 2017]) e outra baseada em *Deep Averaging network* [Iyyer et al., 2015]. Em ambas as arquiteturas, o USE é otimizado em uma seleção de problemas diversos dentro do universo de processamento de linguagem natural. Um dos problemas que é utilizado na otimização do USE é a tarefa de *skip-thought* [Kiros et al., 2015]. Basicamente a tarefa consiste em, dada uma sentença, prever a sentença anterior e a próxima sentença. Outra tarefa utilizada é a tarefa de *Conversational Input-Response Prediction*, tarefa que consiste em, dada uma frase de uma conversa, escolher a resposta correta, dentre uma lista de respostas. A tarefa é inspirada em Henderson et al. [2017], artigo em que os autores propõem uma arquitetura escalável para um sistema de previsão de resposta a *emails*. Na Figura 2.4, podemos ver um exemplo dessa tarefa.

A Terceira tarefa utilizada para otimizar o USE é a tarefa de inferência em linguagem natural (*Natural Languagem Inference*). Nessa tarefa, são dados pares de hipótese-premissa e o objetivo de previsão é avaliar se a hipótese contradiz a premissa, se a hipótese decorre da premissa, ou ainda se a hipótese é neutra à premissa. Na Tabela 2.2, vemos alguns exemplos dessa tarefa[2].

**Tabela 2.2:** Exemplos da tarefa de inferência em linguagem natural.

| Hipótese | Premissa | Julgamento |
|---|---|---|
| Há um jogo de futebol com alguns homens jogando | Alguns homens estão praticando um esporte | Implicação |
| Eu amo filmes da Marvel | Eu odeio filmes da marvel | Contradição |
| Eu amo filmes da Marvel | Um navio chegou | Neutra |

[2]A Tabela apresentada foi adaptada para o português, a versão original, pode ser encontrada em https://amitness.com/2020/06/universal-sentence-encoder/

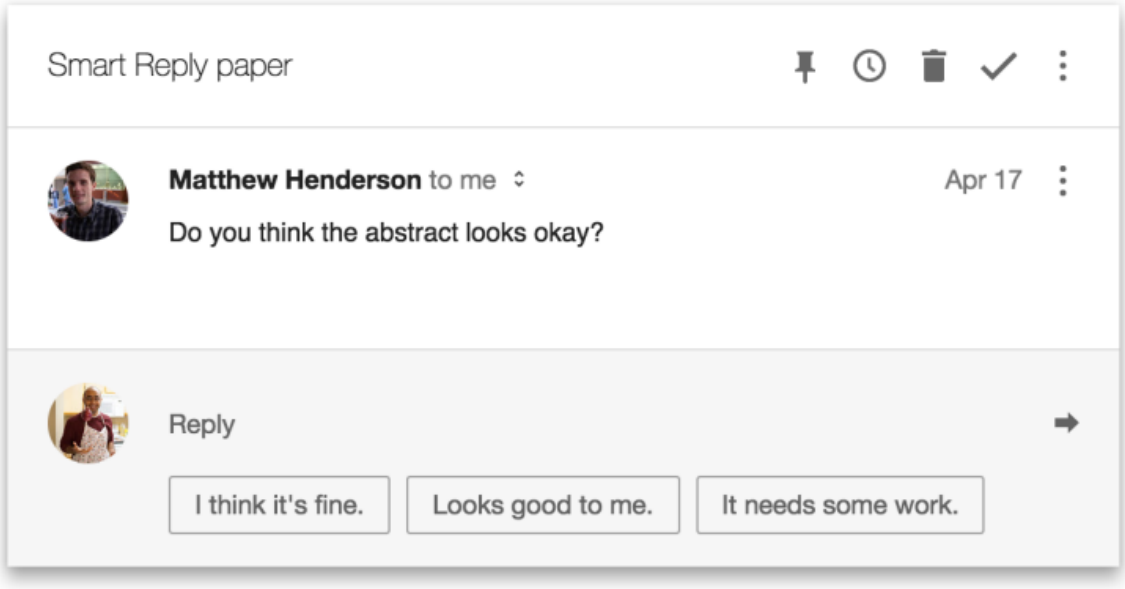
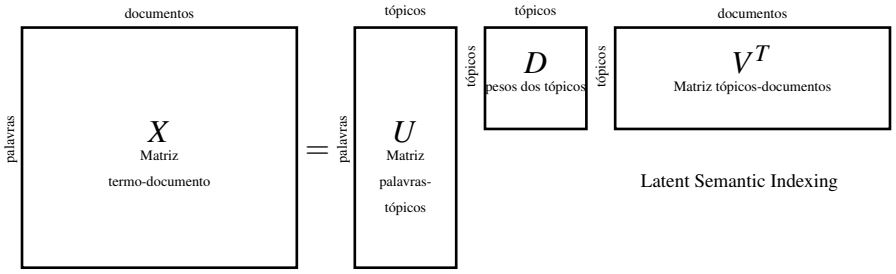



**Figura 2.4:** Exemplo de Conversational Input-Response Prediction. Figura retirada de Henderson et al. [2017].

## 2.6 Latent Semantic Indexing

O LSI é um método que busca mapear um conjunto de documentos para uma série de tópicos, indicando o quanto cada documento está associado à cada tópico. Cabe pontuar, que quando usamos o termo tópico, estamos na realidade discorrendo sobre conceitos, sobre conjunto de palavras que aparecem juntas. Dentro de um corpus com textos versando, por exemplo sobre futebol, política e ciência da computação, o algoritmo não informará que os textos tratam sobre futebol, política ou ciência da computação, mas provavelmente conseguirá identificar que existem 3 tópicos nos quais é possível separar grande parte dos documentos, e conseguirá indicar pra cada documento e cada tópico, o nível de pertencimento do documento ao tópico.

O funcionamento do LSI começa com uma matriz termo-documento, matriz que possui uma linha para cada documento e em que cada coluna indica a quantidade de vezes que cada palavra do vocabulário ocorreu no documento. A matriz termo-documento segue uma ideia bastante semelhante a uma abordagem *Bag-of-words*. O algoritmo utiliza a Singular Value Decomposition (SVD) para decompor essa matriz em um produto de três matrizes: A matriz $U$, matriz ortogonal que representa o nível de associação das palavras aos tópicos, a matriz $D$, matriz diagonal que representa a importância dos tópicos, e a matriz $V$ transposta, matriz ortogonal que representa o nível de associação dos documentos aos tópicos.




No contexto de nosso trabalho, a matriz mais importante é a matriz $V$ transposta, justamente porque essa matriz indica para cada documento, o nível de associação do documento com cada tópico e portanto é a matriz que fornece a representação vetorial que estamos buscando para o documento.

## 2.7 TF-IDF

O TF-IDF é uma estatística que representa o quão relevante um termo é para um documento, a partir da frequência desse termo no documento e em outros documentos do *corpus*. É possível dizer que o TF-IDF aponta as palavras com as quais podemos sumarizar um documento [Qaiser and Ali, 2018].

A sigla surge a partir da combinação de dois termos em inglês: Term Frequency (TF) e Inverse Document Frequency (IDF). A métrica é calculada conforme mostrado na Equação 2.13:

$$\text{TF-IDF} = TF * IDF \tag{2.13}$$

Para entender o significado da métrica, é interessante analisar o significado de cada termo em separado. O TF representa uma métrica do quão frequente é um termo em um documento. Um mesmo *corpus* pode ter documentos de tamanhos bastante variados, portanto, para evitar que o tamanho do documento influencie na métrica, analisamos a frequência relativa e não a frequência absoluta das palavras. O cálculo, portanto, segue conforme apresentado na Equação 2.14 em que $Q_p$ representa a quantidade de vezes que o termo aparece no documento e $T_d$ representa o tamanho do documento:

$$TF = Q_p / T_d \tag{2.14}$$

O TF traz informação sobre a frequência do termo no documento, mas essa informação sozinha não garante que um termo seja relevante ao documento, pois podemos estar tratando de uma

*stopword*, e nesse caso, teremos uma palavra que embora seja bastante frequente no documento, não estará contribuindo de forma relevante para o significado. Para endereçar esse problema, é calculada também uma outra métrica o *IDF*, conforme será mostrado na Equação 2.15 em que $D_t$ representa a quantidade de documentos total e $D_w$ a quantidade de documentos em que o termo aparece.

$$IDF = \ln(D_T / D_W) \tag{2.15}$$

A explicação sobre o funcionamento do TF-IDF feita acima foi adaptada de Qaiser and Ali [2018].

## 2.8 Trabalhos Relacionados

Há diversas propostas para realizar a tarefa de avaliação automática de itens discursivos. Elas diferem em vários eixos, entre eles destacamos os conjuntos de dados disponíveis para treino, as características extraídas, as formas como elas foram extraídas, as métricas de avaliação usadas, e as técnicas de aprendizado de máquina usadas em cada sistema proposto.

Adamson et al. [2014] propõem um sistema baseado em um modelo Support Vector Regression (SVR) que opera principalmente com n-gramas, marcação de partes do discurso *(Part-of-Speech (POS) - tagging)*, contagens de caracteres, palavras, sentenças e contagem de erros ortográficos, experimentando tal sistema no conjunto de dados fornecido pelo *Kaggle* na competição ASAP-AES. Os autores experimentaram o uso de LSI, tanto como geradora de características para o SVR, quanto como modelo preditivo independente. O uso do LSI como modelo preditivo independente produziu resultados visivelmente inferiores aos produzidos pelo SVR, enquanto o uso do LSI como gerador de características para o SVR não trouxe ganho relevante, quando comparado ao uso do SVR sem essas características, sendo inclusive visivelmente inferior em vários conjuntos de dados, conforme mostrado na Figura 2.5. Esse trabalho apresenta uma observação interessante: a quantidade de palavras únicas está relacionada a nota obtida pelo aluno, mesmo em um cenário no qual a contagem de palavras não está relacionada a nota obtida. Os autores sugerem que a explicação de tal fenômeno possa estar relacionada ao vocabulário e sofisticação de escrita demonstrados pelo aluno.

Sultan et al. [2016] apresentam um modelo de regressão *Ridge* baseado em características que medem a similaridade semântica entre a resposta do estudante e respostas de referência. As

variáveis utilizadas são a similaridade semântica, calculada por meio de uma soma ponderada entre as similaridades léxica e contextual, além da similaridade de vetor semântico, em que tanto a resposta de referência quanto a resposta do estudante são submetidas a um processo de *text embedding*, de modo a obter sua representação vetorial, e é calculada a similaridade de cosseno entre ambos os vetores. Tais características são recomputadas usando uma técnica conhecida como *question demoting* que consiste em remover, tanto da resposta de referência, quanto da resposta do estudante, as palavras que aparecem no enunciado. O objetivo da técnica é evitar recompensar a simples repetição de palavras presentes no enunciado. Os autores avaliaram o sistema em desafios de avaliação formatados dentro das duas grandes tarefas de aprendizado supervisionado: uma tarefa de regressão proposta por Mohler et al. [2011] e uma tarefa de classificação proposta em Agirre et al. [2012].

Shehab et al. [2016] propõem um sistema contendo duas componentes complementares: um motor de avaliação baseado em rede neural e um segundo componente de análise de características de escrita, o qual é composto por uma série de programas que avaliam a gramática, o estilo de escrita e a estrutura discursiva do texto e proveem *feedback*. No motor de avaliação, acontecem algumas etapas antes de se passar pela rede neural. Primeiramente, acontece uma checagem gramatical, após a qual, o texto é *tokenizado* palavra por palavra e o algoritmo remove as *stopwords* (palavras como 'a', 'the' , 'at', as quais atuam majoritariamente como conectores e não acrescentam informação ao texto). Após isso, o sistema realiza o processo de *stemming*, que modifica o final da palavra de modo a mapear palavras com mesma raiz, ou inflexões da mesma palavra para um ponto comum. A rede neural que os autores utilizam é o algoritmo *learning vector quantization*, algoritmo que busca realizar uma classificação, e que é baseado em prototipação. Neste algoritmo, os neurônios aprendem um vetor de protótipo que é utilizado para classificar cada observação e a previsão é dada de acordo com o vetor de protótipos, o qual é atualizado levando em conta se o exemplo está mais próximo ao protótipo de uma ou de outra classe. O módulo de avaliação da escrita possui um modo de gramática e uso da língua, ou seja, ele avalia se as construções empregadas fazem ou não sentido na língua inglesa, para isso os autores utilizam um corpus de 30 milhões de palavras oriundas de jornais. Além das combinações de bigramas e trigramas presentes e é comparada a frequência dos bigramas e trigramas do texto apresentado pelo aluno com a frequência apresentada no corpus. Existe um segundo módulo, que se atenta para a possível confusão entre palavras homófonas. Há também um terceiro módulo, de detecção de estilo indesejado, o qual se preocupa com sentenças muito

longas ou muito curtas, ou repetições excessivas de palavras. Finalmente, há um modulo de elementos do discurso, que identifica se o autor estruturou seu texto em introdução desenvolvimento, argumentação, e conclusão.

Bachman et al. [2002] propõem um sistema chamado *WebLas*, que consiste em uma abordagem para avaliação de respostas curtas baseada em um sistema web. O sistema conta com apoio de duas ferramentas externas: o *LinkGrammar* [Grinberg et al., 1995] e o *WordNet* [Oram, 2001]. No sistema proposto, o primeiro módulo é o modulo descrito como criador de exame, que na realidade é um módulo por meio do qual o elaborador insere no sistema os elementos do exame entre os quais estão os textos de apoio, o enunciado do item e as respostas modelo. O *WebLas* então envia para o *LinkGrammar* a resposta modelo inserida e este por sua vez realiza dois processos: o tageamento POS e o *parsing*. Basicamente, o sistema realiza esses processos no sentido de "interpretar" a resposta. O sistema submete sua interpretação à validação do elaborador, o qual pode confirmar, ou não, a interpretação do sistema. Após a confirmação, o *WebLas* pesquisa na *Wordnet* por sinônimos e submete esses resultados, novamente a uma validação pelo elaborador, o qual pode ainda, adicionar novos sinônimos. Após esse incremento no conjunto de respostas, o avaliador atribui pontuações às respostas, ou seja, o sistema fornece a funcionalidade de atribuir crédito parcial, não indicando as respostas simplesmente como corretas ou erradas. Considerando os conjuntos de respostas que receberam cada avaliação pelo elaborador, o sistema elabora expressões regulares para cada conceito. O módulo final de cálculo das notas é relativamente simples, nele o sistema avalia se a resposta é compatível com alguma das expressões regulares. Para evitar que pequenos desvios ortográficos atrapalhem a avaliação, o sistema aplica diferentes técnicas como *stemming* e *edit distance*. A partir do momento em que a resposta se encaixa na expressão regular para um conceito, o sistema atribui o conceito à resposta.

Ramachandran et al. [2015] não apresentam um sistema para avaliar respostas discursivas, mas desenvolve uma abordagem que pode ser usada como parte de um sistema maior. A abordagem proposta consiste em um método de geração automática de expressões regulares para avaliação de respostas discursivas. O método usa o texto de rubrica dada aos examinadores e as respostas que obtiveram melhor nota para gerar os padrões. A abordagem consiste em, para cada sentença no texto de rubrica: gerar os *word-order-graphs* e extrair as arestas do grafo, substituir as *stopwords* e *function words* (palavras que tem função gramatical, mas não adicionam conteúdo ao texto) pelo padrão w0,4 (ou seja, qualquer conjunto de até 4 palavras ou ainda

nenhuma palavra), sem alterar as palavras de conteúdo. Ou seja, gerar variações da resposta original que sejam robustas a simples alterações em artigos e preposições, por exemplo. Ordenar os *tokens* que aparecem mais frequentemente nas respostas mais bem avaliadas por avaliadores humanos, no enunciado do item, ou nos textos de apoio e selecionar na rubrica os *tokens* mais frequentes nessas três fontes mencionadas, Dentre esses tokens mais frequentes os autores utilizam a *WordNet*, para gerar *tokens* sinônimos. Posteriormente o sistema adiciona os sinônimos na classe de alternativas e realiza o processo de *stemming*. Finalmente o sistema combina todas as classes de palavras e obtém os conjuntos de expressões regulares. A ideia geral da proposta é que as presenças de termos descritos por essas expressões regulares possam ser usadas como características para um modelo preditivo que de outra forma teria de contar com expressões regulares geradas manualmente, processo que demandaria muito esforço.

Nesse sentido, podemos observar que os trabalhos que propõem sistemas automáticos para avaliação de redações tendem a seguir uma formulação geral de definir aspectos da qualidade textual que serão avaliados, desenvolver algum algoritmo que a partir do texto busca extrair esses aspectos como características e treinar um modelo preditivo com essas características e a nota dada por avaliadores humanos. Nessa linha podemos observar Adamson et al. [2014] e Shehab et al. [2016]. Por outro lado, na avaliação de respostas discursivas, os sistemas tendem a se basear em uma lista de respostas sugeridas para a pergunta, e fazer uma análise comparativa entre a resposta do estudante e a resposta sugerida, usando abordagens como expressões regulares [Bachman et al., 2002], [Ramachandran et al., 2015], ou utilizando conceitos de similaridade semântica [Sultan et al., 2016]. A Tabela 2.3 faz uma síntese dessa análise comparativa, informando se o artigo aborda redações ou respostas discursivas, informando os autores do artigo e sumarizando a abordagem empregada.

**Tabela 2.3:** Apresentação dos trabalhos relacionados.

| Tipo | Artigo | Abordagem |
|---|---|---|
| Redação | Shehab et al. [2016] | Rede neural |
| Redação | Adamson et al. [2014] | SVR |
| Resposta Discursiva | Ramachandran et al. [2015] | Expressão regular, visando alimentar modelo preditivo |
| Resposta Discursiva | Bachman et al. [2002] | Expressao regular,sem usar modelo preditivo |
| Resposta Discursiva | Sultan et al. [2016] | Similaridade semantica e Regressao ridge |

A Tabela 2.4 mostra uma comparação entre a abordagem proposta por Ramachandran et al.

[2015] e duas abordagens mostradas em Tandalla [2012] uma envolvendo as expressões regulares montadas por Tandalla [2012] e outra abordagem sem estas expressões regulares

**Tabela 2.4:** Comparação entre os resultados obtidos por Ramachandran et al. [2015] e Tandalla [2012]. Tabela adaptada de Ramachandran et al. [2015]. Os resultados são medidos em termos de QWK

| Conjunto | Auto P | Tandalla | Baseline |
|---|---|---|---|
| 1 | 0.86 | 0.85 | 0.82 |
| 2 | 0.78 | 0.77 | 0.76 |
| 3 | 0.66 | 0.64 | 0.64 |
| 4 | 0.70 | 0.65 | 0.66 |
| 5 | 0.84 | 0.85 | 0.80 |
| 6 | 0.88 | 0.88 | 0.86 |
| 7 | 0.66 | 0.69 | 0.63 |
| 8 | 0.64 | 0.62 | 0.59 |
| 9 | 0.84 | 0.84 | 0.82 |
| 10 | 0.79 | 0.78 | 0.76 |
| Média | 0.78 | 0.77 | 0.75 |

A Figura 2.5 mostra uma análise comparativa medida em termos de QWK entre as abordagens testadas pelo autor.

No presente trabalho, avaliamos diferentes abordagens, algumas das quais convergem com as abordagens apresentadas acima, enquanto outras divergem das abordagem propostas nos trabalhos elencados acima, pois dentre estes, somente Sultan et al. [2016] emprega a ideia de representação vetorial de texto usando abordagens de *text embedding*, mesmo assim a representação textual de uma sentença é feita, usando o modelo para representar palavra por palavra e ponderando as representações das palavras de modo a representar a sentença, enquanto em nossa abordagem, usamos técnicas que possibilitam representar um trecho de texto. Cabe pontuar que nosso trabalho difere dos elencados acima pela avaliação de diferentes abordagens de previsão, em especial da classificação ordinal.

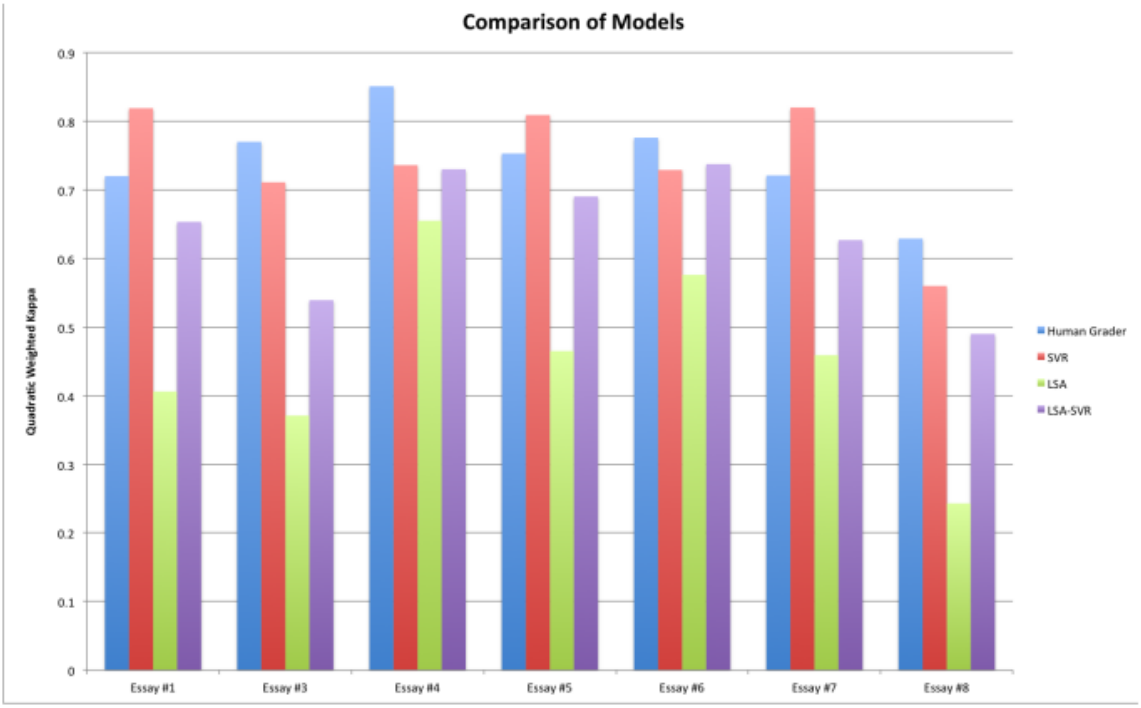



Figure 1: Model Results

**Figura 2.5:** Para cada abordagem, é mostrado o desempenho preditivo medido em termos de QWK entre as notas previstas pela abordagem e a nota final atribuída por avaliadores humanos. A título de comparação é trazido o QWK entre as notas parciais dadas pelos avaliadores humanos. Figura retirada de Adamson et al. [2014].

# Capítulo 3

# Avaliação automática de respostas a itens discursivos

Esse capítulo se destina à apresentação do *pipeline* construído para a avaliação de itens discursivos. A Figura 3.1 apresenta um fluxograma com os diferentes módulos que compõem o *pipeline* de avaliação automática de itens discursivos. Nas próximas seções, descrevemos cada um desses módulos. A Seção 3.1 apresenta o módulo inicial de pré-processamento, o qual realiza a correção de erros ortográficos e uniformiza os esquemas de colunas dos conjuntos de redações e respostas discursivas. A Seção 3.2 apresenta as técnicas de representação vetorial utilizadas, além de descrever a construção de algumas *features* comuns a todas as técnicas e apresentar a motivação para a escolha das técnicas utilizadas. A Seção 3.3 explica o treinamento dos modelos dentro das abordagens de previsão utilizadas. Finalmente a Seção 3.4 explica o processo de previsão empregado em cada abordagem e a avaliação dos modelos.

## 3.1 Pré-processamento

Para corrigir o texto de cada item discursivo, primeiramente aplicamos uma função que criamos para corrigir erros simples, como espaçamento indevido antes ou após a virgula, repetição da palavra *the*, entre outros.[1]

De modo a ilustrar o procedimento descrito, recorremos a duas respostas retiradas do conjunto de dados de respostas discursivas. (Grifamos algumas partes das respostas para auxiliar na percepção das modificações realizadas).

- To replicate this experiment you would also need to know vinergar they used, the size of the cup and **the the** size of the samples.[2]
- The word “invasive” is significant throughout the article because the whole article is about how the python is rapidly invading the tropical climates of the **U.S.** One reason is because

[1] Para essa correção inicial, usamos a biblioteca *re*. Para mais informações consulte https://docs.python.org/3/library/re.html

[2] A resposta pode ser encontrada no conjunto de treino de respostas discursivas fornecido, ela possui TextId 144 e pertence ao primeiro conjunto de respostas

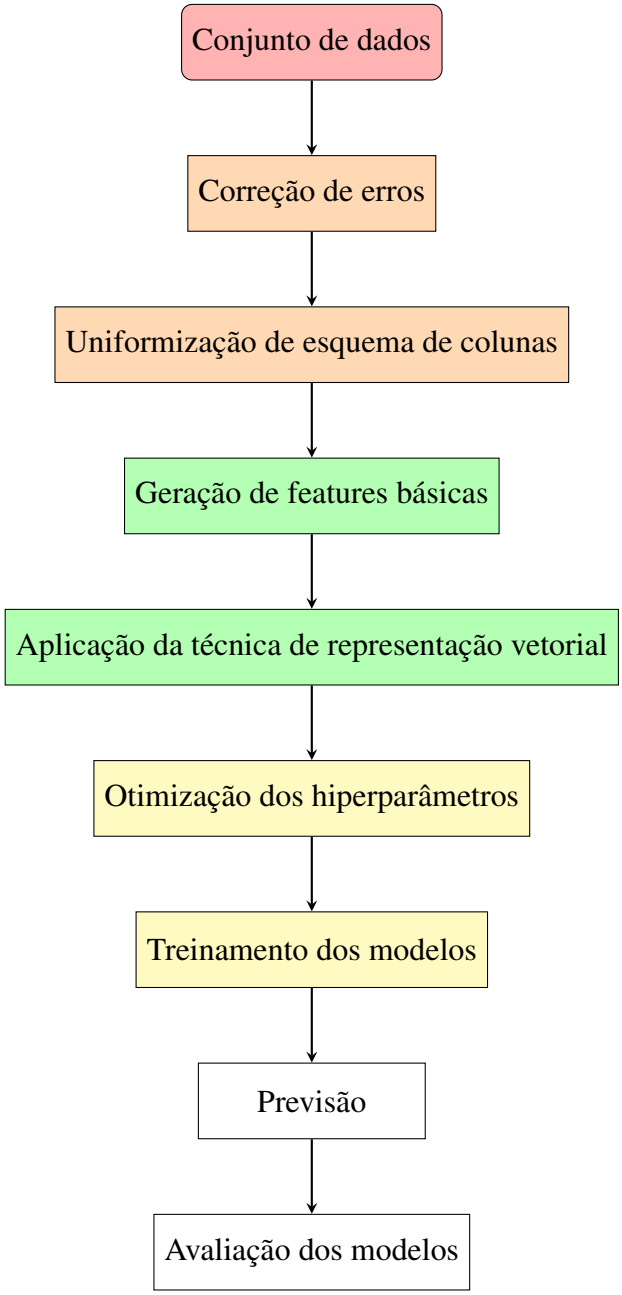



**Figura 3.1:** Fluxograma do pipeline de avaliação automática de itens discursivos. Os processos mostrados em laranja representam processos componentes da etapa de pré-processamento, os processos mostrados em verde representam processos componentes da etapa de extração de *features*. Os processos descrito em amarelo compõem a etapa de treinamento dos modelos. Finalmente, os processos descritos em branco compõem a etapa de previsão e avaliação.

of Horncane Andrelilyns stated in **paragraph 12,that really** picked off the invasive capect of the large python.[3]

Após o procedimento descrito anteriormente, temos como resultado, respectivamente:

- To replicate this experiment you would also need to know vinergar they used, the size of the cup and **the** size of the samples.

- The word “invasive” is significant throughout the article because the whole article is about how the python is rapidly invading the tropical climates of **the U. S. One** reason is because of Horncane Andrelilyns stated in **paragraph 12, that really** picked off the invasive capect of the large python.

Cabe pontuar que o procedimento adotado é um procedimento realmente simplista adotado para a correção de erros ortográficos, e que, faz substituições adequadas em alguns casos como substituir ’the the’ por ’the’, e colocar o espaço após a vírgula antes de ’that’,mas realiza substituições errôneas como colocar espaço após o ponto em ’U.S’, resultando em ’U. S.’.

Após a aplicação dessa correção inicial, realizamos a separação do texto em sentenças[4], corrigimos a sentença e concatenamos a sentença corrigida em uma cadeia de texto (*string*), de modo a gerar o novo texto.

Explicaremos agora, o processo de correção utilizado sentença a sentença. Primeiramente, a sentença é submetida a um processo chamado tokenização, o qual consiste em fazer a separação da sentença em palavras (que na terminologia de processamento de linguagem natural, são chamadas, muitas vezes, de *tokens*).

A etapa descrita acima gera as seguintes listas de *tokens*: (Cada *token* é separado por vírgula e os *tokens* grifados são palavras identificadas como soletradas erroneamente):

- ’To’, ’replicate’, ’this’, ’experiment’, ’you’, ’would’, ’also’, ’need’, ’to’, ’know’, **’vinergar’**, ’they’, ’used’, ’,’, ’the’, ’size’, ’of’, ’the’, ’cup’, ’and’, ’the’, ’size’, ’of’, ’the’, ’samples’

- ’The’, ’word’, ’“’, ’invasive’, “”, ’is’, ’significant’, ’throughout’, ’the’, ’article’, ’because’, ’the’, ’whole’, ’article’, ’is’, ’about’, ’how’, ’the’, ’python’, ’is’, ’rapidly’, ’invading’, ’the’, ’tropical’, ’climates’, ’of’, ’the’, ’U.’, ’S.’, ’One’, ’reason’, ’is’, ’because’,

[3] A resposta pode ser encontrada no conjunto de treino de respostas discursivas fornecido, ela possui TextId 9013 e pertence ao quarto conjunto de respostas

[4] Para essa separação, utilizamos a biblioteca de processamento de linguagem natural *Natural Language Toolkit* (NLTK). Para mais informações, consulte `https://www.nltk.org`

'of', **'Horncane'**, **'Andrelilyns'**, 'stated', 'in', 'paragraph', '12', ',', 'that', 'really', 'picked', 'off', 'the', 'invasive', **'capect'**, 'of', 'the', 'large', 'python', '.'

Nos dois exemplos mostrados, o processo de separação de sentenças identificou que o texto é composto somente por uma única sentença. No primeiro exemplo, em nossa interpretação, tal identificação foi correta, porém, no segundo exemplo, tal identificação foi errônea. Acreditamos que o motivo da identificação errônea no segundo caso é a ambiguidade entre o segundo ponto de U.S. e um possível ponto final. Cabe pontuar, que como o procedimento consiste em separar sentenças, para separar palavras e corrigir palavras individualmente, um eventual erro na separação em sentenças não importa prejuízo ao funcionamento pretendido do corretor.

A lista de palavras que corresponde a uma sentença passa por uma verificação. Nessa verificação, identificamos as palavras que provavelmente estão ortograficamente incorretas por meio de um dicionário de frequência montado com base em um corpus de textos da língua inglesa. Para cada palavra de uma sentença de entrada, obtemos (por meio da biblioteca usada) a correção mais provável para a palavra. De posse da correção sugerida, substituímos as ocorrências da palavra identificada como errada pela correção sugerida.[5]

Após última etapa, temos como resultado final da correção:

- To replicate this experiment you would also need to know vinegar they used, the size of the cup and the size of the samples.

- The word "invasive" is significant throughout the article because the whole article is about how the python is rapidly invading the tropical climates of the U. S. One reason is because of Horncane Andrelilyns stated in paragraph 12, that really picked off the invasive expect of the large python.

Após a correção de erros ortográficos, o *pipeline* realiza outros preprocessamentos, de modo a realizar a padronização do esquema de colunas. No conjunto de redações, é feito um processo de *Data augmentation* ao transformar as duas informações de nota de domínio em dois exemplos de texto, um dos exemplos terá como nota a nota do primeiro domínio e outro dos exemplos, terá como nota a nota do segundo domínio, além disso, é acrescida uma coluna *Domain* de modo a identificar se a nota se refere a uma nota do primeiro ou do segundo domínio. No conjunto de respostas discursivas, é acrescida uma coluna *Domain* com valores 1 para todos os

[5]Para realizar os procedimentos descritos, usamos a biblioteca *pyspellchecker*. Para mais informações sobre a biblioteca, acesse:https://pypi.org/project/pyspellchecker/

exemplos, para uniformizar os esquemas de colunas entre os conjuntos de redações e respostas discursivas. Além disso, cabe pontuar que as colunas são renomeadas de modo a uniformizar os esquemas entre os conjuntos. Finalmente cabe pontuar que, como o segundo conjunto de redações é o único a ter dois domínios, e, portanto, dois objetivos de previsão, para não aumentar o escopo do trabalho, decidimos trabalhar somente com os exemplos associados ao primeiro domínio. Os exemplos relativos a notas do segundo domínio, serão porém mantidos, pois pretendemos analisar a relação entre as notas dos dois domínios, conforme apontaremos na Seção 5.2.

## 3.2 Módulo de extração de *Features*

A Seção 3.2.1 apresenta a etapa inicial que é executada para todas as técnicas de representação vetorial utilizadas e que gera algumas *Features* básicas, além de separar para cada enunciado um conjunto de treino e um conjunto de teste. A Seção 3.2.2 explica o motivo de escolha das técnicas de representação vetorial que analisamos nesse trabalho. Finalmente, a Seção 3.2.3 explica como aplicamos cada técnica de representação vetorial no contexto de nosso pipeline.

### 3.2.1 Etapa Inicial Básica

Há uma etapa geral no processo de extração de *features*, que é realizada junto à todas as técnicas de representação vetorial utilizadas. Nessa etapa, geramos algumas *features* mais básicas, e que estarão presentes junto à representação vetorial do texto, como a quantidade de sentenças, a quantidade de palavras e a quantidade de palavras únicas. Calculamos a quantidade de sentenças[6], calculamos também as quantidades de palavras e palavras únicas[7]. Nessa etapa, realizamos a separação, para os itens discursivos de cada enunciado, em treino e teste, fazendo a separação de forma estratificada pela nota (variável-alvo), além disso trabalhamos apenas com textos em que mais de 10 textos do conjunto tenham recebido aquela nota. Isso é realizado, pois entendemos que 10 exemplos é o mínimo necessário para poder realizar uma boa separação de forma estratificada (sendo uma separação 80/20, com 10 exemplos conseguimos garantir 8 exemplos no treino e 2 no teste). Finalmente, cabe pontuar que fazemos um embaralhamento (*shuffling*) dos exemplos antes de separar em treino e teste, para prevenir os efeitos de

[6]Para essa contagem, usamos o módulo *sent tokenizer* da biblioteca NLTK
[7]Para essa contagem, usamos o módulo *word tokenizer* da biblioteca NLTK

uma eventual ordem não aleatória dos registros no conjunto de dados. Todos esse procedimento é realizado de modo a garantir que nas diferentes abordagens avaliadas estejamos levando em consideração os mesmos conjuntos de dados, garantindo assim uma avaliação comparativa mais robusta entre as diferentes técnicas.

### 3.2.2 Escolha das técnicas de representação vetorial

Escolhemos trabalhar com o TF-IDF por ser um *baseline* importante entre as técnicas de representação vetorial de texto. Em relação às outras técnicas, vale lembrar que as competições cujos conjuntos utilizamos nesse trabalho ocorreram em 2012 e, desde então, novas técnicas de representação vetorial têm surgido como o *Doc2vec* [Le and Mikolov, 2014] e o USE [Cer et al., 2018]. Escolhemos essas técnicas por serem posteriores às competições e estarmos interessados em como *pipelines* envolvendo tais técnicas desempenham nos conjuntos de dados dessas competições. Adicionalmente, decidimos trazer uma técnica disponível há mais tempo, o LSI, por ser uma técnica mais antiga e por, apesar de já ter sido empregada na tarefa de avaliação de redações [Adamson et al., 2014], tal emprego ocorrer de forma diferente da que propomos nesse trabalho.

### 3.2.3 Técnicas de representação vetorial empregadas

#### Representação vetorial usando USE

Elaboramos uma versão do conjunto em que extraímos representações vetoriais dos textos utilizando um modelo que foi pré-treinado usando o algoritmo USE, que é disponibilizado pelo *Google*[8], a dimensionalidade utilizada no *embedding* é a de 512 que é a dimensionalidade oferecida, ou seja, cada texto é mapeado para um vetor 512-dimensional. Infelizmente, apenas os modelos pré-treinados são disponibilizados, e portanto, não é possível treinar uma versão do USE usando os conjuntos de texto avaliados nesse trabalho. Vale lembrar que como o modelo utilizado na representação vetorial não é treinado com os conjuntos de redações ou de respostas discursivas utilizadas nesse trabalho, o mapeamento do texto de um elemento $t_i$ do conjunto independe do mapeamento de qualquer outro elemento $t_j$do conjunto, por exemplo eliminar $t_j$ do conjunto ou realizar um processo de *Data Augmentation* inserindo $n$ cópias de $t_j$ no conjunto,

[8]O modelo pré-treinado pode ser baixado a partir da seguinte URL:“https://tfhub.dev/google/universal-sentence-encoder/4”

não deveria alterar a representação de $t_i$.

**Representação vetorial usando *Doc2vec***

Elaboramos algumas versões do conjunto em que extraímos variáveis dos textos utilizando o modelo *Doc2vec*. Diferentemente da versão que utiliza USE, nós não utilizamos um modelo pré-treinado, pois é possível treinar versões do *Doc2vec*. Dentre as duas arquiteturas possíveis, PV-DM e PV-DBOW, utilizamos a arquitetura PV-DBOW. Para treinar os nossos modelos *Doc2vec*, concatenamos o conjunto de redações e o conjunto de respostas discursivas, portanto nossos modelos *Doc2vec* são treinados com um corpus de aproximadamente 30000 textos, compreendendo os 8 conjuntos de redações e 10 conjuntos de respostas discursivas. Vale lembrar que no caso do *Doc2vec*, para cada dimensionalidade avaliada, é treinado um único modelo com todo o corpus, ou seja, o modelo usado para a representação vetorial de cada texto conhece todo o conjunto de textos. Tomamos a decisão de concatenar todos os conjuntos, tanto de redações, quanto de respostas discursivas, para poder treinar o *Doc2vec* em uma massa de dados maior, por entender que ao ter o *Doc2vec* treinado em uma massa de dados maior, teríamos ganho de desempenho preditivo. As dimensionalidades do *Doc2vec* que utilizamos foram 32, 64, 128, 256 e 512.

**Representação vetorial usando LSI**

Elaboramos também algumas versões do conjunto de dados, em que fazemos a extração de variáveis usando a técnica LSI. O primeiro passo é utilizar os textos para construir um vocabulário, que é uma coleção de todas as palavras que aparecem no corpus. Dado esse vocabulário, é construída a matriz termo-documento, matriz em que cada linha representará um documento e cada coluna indicará a quantidade de vezes em que o termo do vocabulário aparece no documento, Cabe pontuar que embora seja possível no LSI utilizar a métrica TF-IDF para ponderar a matriz termo-documento, decidimos não fazê-lo, pois uma das técnicas de representação vetorial usadas nesse trabalho é o TF-IDF e, portanto não queremos criar uma redundância entre as diferentes técnicas utilizadas. Após construir a matriz termo documento, o algoritmo utiliza essa matriz para construir um modelo LSI com uma determinada quantidade de tópicos, e esse modelo transforma a matriz termo documento em uma matriz que indica para cada documento o nível de pertencimento a cada tópico, e portanto, provê a representação vetorial desejada. Os valores de tópicos trabalhados são 10, 20, 30, 40, 50 e 100 tópicos.

**Representação vetorial usando TF-IDF**

Para cada conjunto de textos, temos a parte separada para teste e a parte separada para treino. Treinamos um modelo que chamamos de vetorizador TF-IDF ou somente vetorizador, usando os documentos separados para treino. Esse vetorizador fornece a representação vetorial de um documento usando a técnica TF-IDF, ou seja levando em conta não somente a frequência da palavra no documento, mas a quantidade de documentos no texto em que ela aparece, tratando-se, portanto, de uma representação vetorial que leva em conta não apenas o documento a ser representado, mas todo o conjunto em que o modelo é treinado. O vetorizador treinado nos documentos do conjunto de treino é usado também para dar a representação vetorial dos documentos do conjunto de teste, fazemos isso no sentido de evitar o fenômeno de vazamento de dados. A representação vetorial dada pelo TF-IDF aos textos com que trabalhamos possui uma dimensionalidade bem alta, por isso adotamos uma técnica de seleção de *features*. A técnica consiste em, para cada *feature* obtida através do TF-IDF, ou seja, contagem de palavras ponderada, calcular sua variância e após isso, selecionar as *N features* com maior variância. Para fazer isso, usamos o comando `.std()` da linguagem *Python* no *DataFrame*, gerando assim uma lista com a todas as colunas obtidas via TF-IDF e seus valores de desvio padrão. Feito isso, usamos o comando `.sort()` para ordenar as colunas em ordem decrescente de desvio padrão, e separamos em uma lista, as *n* primeiras colunas, ou seja, as *n* colunas com maior desvio padrão, obtida a lista, usamos a lista para filtrar um *Dataframe*, e portanto gerar a representação vetorial via TF-IDF com dimensionalidade *n*. A ideia subjacente é que *features* que tenham variância muito baixa têm pouco poder para diferenciar os exemplos e como consequência apresentarão menor poder preditivo, as quantidades de *features* que selecionamos são 32, 64, 128, 256 e 512. Cabe pontuar um detalhe interessante que é o fato de que, graças ao método de seleção que usamos, os conjuntos de menor dimensionalidade estão contidos nos de maior dimensionalidade (isso é, as 512 *features* de maior variância e, portanto, maior desvio padrão, incluem as 256 *features* de maior variância, que por sua vez incluem as 128 *features* de maior variância e por aí em diante).

## 3.3 Módulo de treinamento dos modelos

Na Seção 3.3.1 explicamos o racional envolvido na escolha do algoritmo de aprendizado de máquina aplicado nesse trabalho, enquanto na Seção 3.3.2 detalhamos, para cada uma das três abordagens: Regressão, classificação e classificação ordinal, como a abordagem foi implemen-

tada no *Pipeline* que construímos.

### 3.3.1 Escolha do algoritmo de Aprendizado de Máquina

Ao pensar o problema das diferentes formas em que pode ser abordado (Regressão, Classificação e Classificação Ordinal), levantamos as principais alternativas de algoritmos de aprendizado de máquina para essa tarefa e temos as seguintes possibilidades:

- Modelos de àrvore
  - Árvore de decisão
  - Florestas aleatórias *(Random Forest)*
  - Modelos baseados em *boosting* (*XGboost*, *Catboost*, entre outros)
- Modelos baseados em *Kernel* como o SVR
- Modelos baseados em Redes neurais

Não elencamos a possibilidade de uso de modelos lineares, pois desejamos ter um único algoritmo do qual possamos usar suas versões de regressão e classificação. O motivo de usar versões do mesmo algoritmo é tornar a comparação entre as abordagens testadas mais equânime. Além da necessidade de garantir igualdade na comparação, não testamos modelos lineares, pois acreditamos que a relação entre as variáveis preditoras e a variável-alvo provavelmente será não-linear. Dentre essas grandes famílias, decidimos não optar pelo SVR por não estarmos dispostos a dispender tempo na seleção do melhor *Kernel*. Decidimos não priorizar modelos baseados em redes neurais, pois queremos trabalhar com modelos em que a explicabilidade é mais simples, além de entendermos que cada conjunto de textos possui um número de exemplos que não é tão adequado ao trabalho com redes neurais (cerca de 2 mil exemplos). Isto posto, entendemos que a melhor família de modelos para se trabalhar é a família de modelos de árvore de decisão, e dentro dessa família, descartamos o algoritmo de árvore de decisão pois, conforme a profundidade e complexidade da árvore criada aumenta, o algoritmo se torna mais suscetível à ocorrência do fenômeno de sobre-ajuste [Ho, 1995], restando então o *random forest* e os algoritmos baseados em *boosting*, decidimos pelo *random forest*, por entendê-lo como algoritmo mais simples entre os elencados.

### 3.3.2 Abordagens para enquadrar o problema de avaliação automática

A Seção 3.3.2 explica como realizamos a otimização de hiperparâmetros e treinamos os modelos de regressão. A Seção 3.3.2 explica como realizamos a otimização de hiperparâmetros e treinamos os modelos de classificação. A Seção 3.3.2 explica como transformamos os problemas de classificação com $N$ classes em $N-1$ problemas de classificação binária, e como treinamos os modelos para esses problemas.

#### Abordagem baseada em Regressão

O módulo de treinamento de modelo sobre a abordagem regressão utiliza a versão de regressão do algoritmo de florestas aleatórias de árvores de decisão (*Random Forest Regressor*). Para cada enunciado e técnica de representação vetorial (levando em conta as diferentes dimensionalidades da mesma técnica), o módulo de treinamento do modelo realiza o processo de otimização de hiperparâmetros. Essa otimização é feita utilizando a técnica de *grid search*. No caso, fazemos o *grid search* para os parâmetros *min samples leaf*, o qual representa o número mínimo de amostras tolerado em um nó de qualquer das árvores construídas, *max depth*, que representa a altura máxima que uma árvore pode ter e definimos também o parâmetro *criterion*, que representa a métrica utilizada na hora de selecionar um atributo para dividir a árvore em nós, para o qual usamos o valor 'squared error', que indica que a métrica usada na hora de decidir como vai ser realizado o *split* do nó é a métrica de menor erro quadrático. Testamos algumas combinações de hiperparâmetros em que cada combinação de hiperparâmetros é avaliada de acordo com a métrica *Determination coefficient*, também conhecida como $R^2$, usando uma abordagem chamada validação cruzada (*cross-validation*), em que os dados são divididos em 5 fatias, de modo que em cada iteração o modelo é treinado com 80% dos dados de treino e 20% dos dados de treino são usados para a validação. Trabalhamos com a divisão em 5 fatias pois utilizar 10 fatias como é o padrão aumentaria o tempo necessário para a execução do pipeline e consideramos que nas configurações atuais o pipeline já possui um tempo de execução elevado. A validação cruzada realizada é chamada validação cruzada estratificada. Nesse tipo de validação cruzada a separação das fatias é feita tomando cuidado para ter distribuições da variável alvo o mais próximo possível entre teste e validação.

**Abordagem baseada em Classificação**

O módulo de treinamento de modelo sobre a abordagem classificação utiliza a versão de classificação do algoritmo de florestas aleatórias de árvores de decisão (*Random Forest Classifier*). Para cada enunciado e técnica de representação vetorial (levando em conta as diferentes dimensionalidades da mesma técnica), o módulo de treinamento do modelo realiza, assim como na abordagem de regressão, processo de otimização de hiperparâmetros, utilizando *grid search*, mas a validação cruzada é realizada sobre a métrica QWK. Para o *Grid Search*, consideramos novamente os hiperparâmetros *min samples leaf* e *max depth* com o mesmo *grid* utilizado na regressão, quanto ao hiperparâmetro *criterion*, usamos o valor *'gini'*. Assim como na regressão, também trabalhamos com a abordagem de validação cruzada estratificada, dividindo o conjunto de dados em 5 fatias.

**Abordagem baseada em Classificação Ordinal**

O módulo de treinamento de modelo sobre a abordagem classificação ordinal utiliza um estimador construído por nós, e que é baseado na abordagem de classificação ordinal descrita em Frank and Hall [2001]. Nesse estimador, a ideia é transformar um problema de classificação com $N$ classes em $N-1$ problemas de classificação binária e treinar modelos para esses $N-1$ problemas de classificação binária. A abordagem possibilita o uso de algoritmos tradicionais para o treinamento de modelos preditivos nos $N-1$ problemas, portanto selecionamos a versão de classificação do algoritmo de florestas aleatórias de árvore de decisão (*RandomForestClassifier*), para evitar que a diferença percebida entre os desempenhos preditivos das abordagens de classificação e classificação ordinal seja atribuída à escolha do estimador. Na Seção 3.4, apresentamos como os $N-1$ modelos treinados são usados para constituir as previsões das $N$ classes do problema original. Finalmente, cabe salientar que decidimos não usar a técnica *grid search* na abordagem de classificação ordinal, pois usar essa abordagem demandaria um tempo de execução bastante elevado. A motivação para o tempo elevado é que, ao utilizar essa abordagem, treinamos um número de modelos significativamente maior que nas outras abordagens, dado que, para cada técnica de representação vetorial, dimensionalidade e texto, ao invés de treinarmos um modelo, como nas outras abordagens, treinamos $N-1$ modelos.

A Tabela 3.1 apresenta uma sumarização sobre as abordagens de previsão explicadas anteriormente, informando a versão do *Random Forest* utilizada e a métrica empregada na validação cruzada. Vale reforçar que não é apontada nenhuma métrica para a validação cruzada na abor-

dagem de classificação ordinal, pois decidimos não usar validação cruzada nessa abordagem.

**Tabela 3.1:** Visão geral acerca das abordagens de previsão empregadas. Nessa tabela apresentamos, para cada abordagem de previsão avaliada, o algoritmo de aprendizado de máquina usado e a métrica usada no processo de validação cruzada. Cabe pontuar que como não realizamos validação cruzada na abordagem de previsão classificação ordinal, nenhuma métrica está indicada.

| Abordagem | Algoritmo Subjacente | Métrica |
|---|---|---|
| Regressão | Regressor *Random Forest* | *Coeficiente* ($R^2$) |
| Classificação | Classificador *Random Forest* | QWK |
| Classificação Ordinal | Classificador *Random Forest* | - |

## 3.4 Módulo de previsão e avaliação

Para as abordagens de regressão e classificação, trata-se de um módulo relativamente pouco complexo que, para cada combinação de enunciado e técnica de representação vetorial (novamente levando em conta as diferentes dimensionalidades da mesma técnica), lê o conjunto de *features*, carrega o modelo treinado no módulo anterior, realiza a previsão utilizando o modelo carregado, aproxima a previsão para o inteiro mais próximo (pois na abordagem de regressão as previsões não são discretas e a métrica QWK precisa de valores inteiros), e finalmente calcula o QWK para o conjunto de dados.

Para a abordagem de classificação ordinal, o processo é bastante semelhante ao processo descrito anteriormente. Porém há uma diferença no sentido de que não é carregado 1 modelo, mas sim $N-1$ modelos. Esses $N-1$ modelos são usados para prever as probabilidades do alvo ser maior que cada uma das classes conforme explicado anteriormente na Seção 2.2, e a partir disso, é calculada a probabilidade de pertencimento a cada classe, conforme mostrado nas anteriormente nas Equações 2.10, 2.11 e 2.12. Dada a probabilidade de pertencimento à cada classe, a previsão pontual atribuindo finalmente uma classe ao exemplo é feita simplesmente elegendo a classe com maior probabilidade prevista.

# Capítulo 4

# Avaliação Experimental

Este capítulo apresenta os experimentos realizados nessa etapa do projeto. A Seção 4.1 apresenta os conjuntos de dados, tanto de redações, quanto de respostas discursivas que estamos utilizando neste trabalho. A Seção 4.2 traz algumas estatísticas sobre os conjuntos de dados, tanto de redações quanto de respostas discursivas. A Seção 4.3 apresenta os experimentos realizados, tanto para redação, quanto para respostas discursivas, envolvendo as quatro técnicas de representação vetorial usadas: LSI, *Doc2vec*, USE e TF-IDF, além das três abordagens de previsão trabalhadas: regressão, classificação e classificação ordinal.

## 4.1 Conjuntos de Dados

Utilizamos conjuntos de dados provenientes de duas competições: ASAP-AES de avaliação automática de redações e ASAP-SAS de avaliação automática de respostas discursivas. O pipeline proposto foi desenvolvido pensando na avaliação de redações, mas também será aplicado na avaliação de respostas discursivas. De forma geral, aplicamos o mesmo *pipeline* para redações e respostas discursivas, nos trechos em que é necessário necessidade de proceder de forma diferente, explicitamos as diferenças e apresentamos o motivo.

### 4.1.1 Redações

Dos 8 conjuntos de redações, 4 são de redações do tipo argumentativo e 4 são de redações baseadas em texto-fonte. Os conjuntos tratam de redações com diferentes temas e escalas de avaliação. A Tabela 4.1 apresenta informações sobre as redações argumentativas, como o número atribuído pelos criadores da competição ao conjunto de redações, a quantidade de redações presentes em cada conjunto, o tema das redações e o intervalo das notas que podem ser atribuídas às redações. Enquanto a Tabela 4.2 apresenta informações sobre as redações baseadas em texto fonte, como o número atribuído pelos criadores da competição ao conjunto de redações, a quantidade de textos presente em cada conjunto, o texto fonte que é fornecido como

apoio para a elaboração da redação, o autor desse texto fonte, e o intervalo das notas que podem ser atribuídas aos textos. Vale ressaltar que o fato de o número do conjunto ser atribuído pelos criadores da competição faz com que tenhamos os conjuntos 1, 2, 7 e 8 na Tabela 4.1 e os conjuntos 3,4,5 e 6 na Tabela 4.2, ao invés de termos uma ordenação sequencial, como por exemplo, 1,2,3 e 4 em uma tabela e 5, 6, 7 e 8 em outra.

**Tabela 4.1:** Apresentação dos conjuntos de dados que envolvem redação do tipo dissertativo-argumentativo. Nas colunas, QTD representa a quantidade de redações que compõem o conjunto. Nos valores, D1 e D2 significam domínio 1 e domínio 2.

| Conjunto | Tema | Intervalo da nota | QTD |
|---|---|---|---|
| 1 | Influência dos computadores na vida das pessoas | 2-12 | 1783 |
| 2 | Censura nas bibliotecas | 1-6(D1) e 1-4(D2) | 3600 |
| 7 | Paciência | 0-30 | 1569 |
| 8 | Benefícios da risada | 0-60 | 723 |

**Tabela 4.2:** Apresentação dos conjuntos de dados que envolvem redação baseada em texto fonte, Nos valores, QTD representa a quantidade de redações que compõem o conjunto.

| Conjunto | Texto fonte | Autor | Intervalo da nota | QTD |
|---|---|---|---|---|
| 3 | ROUGH ROAD AHEAD: Do Not Exceed Posted Speed Limit | Joe Kurmaskie | 0-3 | 1726 |
| 4 | Winter Hibiscus | Minfong Ho | 0-3 | 1771 |
| 5 | Home: The Blueprints of Our Lives | Narciso Rodriguez | 0-4 | 1805 |
| 6 | The Mooring Mast | Marcia Amidon Lüsted | 0-4 | 1800 |

### 4.1.2 Respostas Discursivas

Tratamos de 10 conjuntos de dados, os quais variam em diferentes aspectos. Quanto à disciplina cuja avaliação pertencem, há 5 conjuntos pertencentes à disciplina de ciências, 3 pertencentes a disciplina de língua inglesa e 2 pertencentes tanto a disciplina de língua inglesa quanto a disciplina de artes. Quanto ao tipo, há 8 casos de texto em que as respostas estão relacionadas de alguma forma à um texto fonte (vale pontuar a similaridade com itens do ENEM nesse sen-

tido) e 2 itens cuja resposta independe de texto de apoio. Cabe finalmente expor a diversidade de assuntos que permeia os itens, os quais perpassam temas como chuva ácida, elasticidade de polímeros a perguntas sobre personagens do texto fonte. A Tabela A.1 dá informações sobre cada conjunto de textos de respostas discursivas, a saber: o número atribuído pelos criadores da competição, a disciplina à que pertencem os itens, a quantidade de textos, o tamanho médio dos textos de cada conjunto, o intervalo de notas possível e o assunto da questão.

## 4.2 Análise exploratória

### 4.2.1 Redações

Na Tabela 4.3, fizemos um compilado de algumas características de cada conjunto de redações, como a média de palavras, o desvio padrão do número de palavras, a média do número de palavras únicas e a média do número de sentenças.

**Tabela 4.3:** A tabela traz breves estatísticas sobre os conjuntos de dados de redações (STD se refere ao desvio padrão da quantidade de palavras e UW, a quantidade de palavras únicas).

| Conjunto | Textos | Média | STD | Média - UW | Média sentenças |
|---|---|---|---|---|---|
| 1 | 1783 | 413 | 139 | 181 | 22 |
| 2 | 3600 | 424 | 175 | 174 | 20 |
| 3 | 1726 | 120 | 60 | 75 | 6 |
| 4 | 1771 | 103 | 57 | 62 | 4 |
| 5 | 1805 | 139 | 67 | 78 | 6 |
| 6 | 1800 | 170 | 63 | 99 | 7 |
| 7 | 1569 | 195 | 103 | 99 | 12 |
| 8 | 723 | 694 | 239 | 265 | 35 |

Na Tabela 4.4 detalhamos, para os conjuntos 2,3,4,5 e 6, quantas redações de cada conjunto obtiveram cada conceito.

**Tabela 4.4:** A tabela traz a quantidade de Redações que receberam cada conceito (conjuntos 2-6).

| Conjunto | 0.0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 |
|---|---|---|---|---|---|---|---|
| 2 | 0 | 53 | 344 | 1493 | 1628 | 75 | 7 |
| 3 | 39 | 607 | 657 | 423 | 0 | 0 | 0 |
| 4 | 312 | 636 | 570 | 253 | 0 | 0 | 0 |
| 5 | 24 | 302 | 649 | 572 | 258 | 0 | 0 |
| 6 | 44 | 167 | 405 | 817 | 367 | 0 | 0 |

A forma de apresentação em tabela não se mostrou adequada para os conjuntos 1,7 e 8, portanto, para esses conjuntos apresentaremos a distribuição das redações por nota em formato de histograma. As Figuras D.9, D.10 e D.11 mostram histogramas que representam a distribuição da quantidade de redações por conceitos para os conjuntos 1, 7 e 8, respectivamente.

Na Figura 4.1, trazemos uma visão sobre a correlação entre o número de palavras, o número de sentenças e a nota obtida em cada redação para o primeiro conjunto de redações. No eixo horizontal, está representada a quantidade de palavras da redação, no eixo vertical, está representada a quantidade de sentenças, enquanto a tonalidade de roxo usada representa a nota obtida, quanto mais escuro, maior a nota.

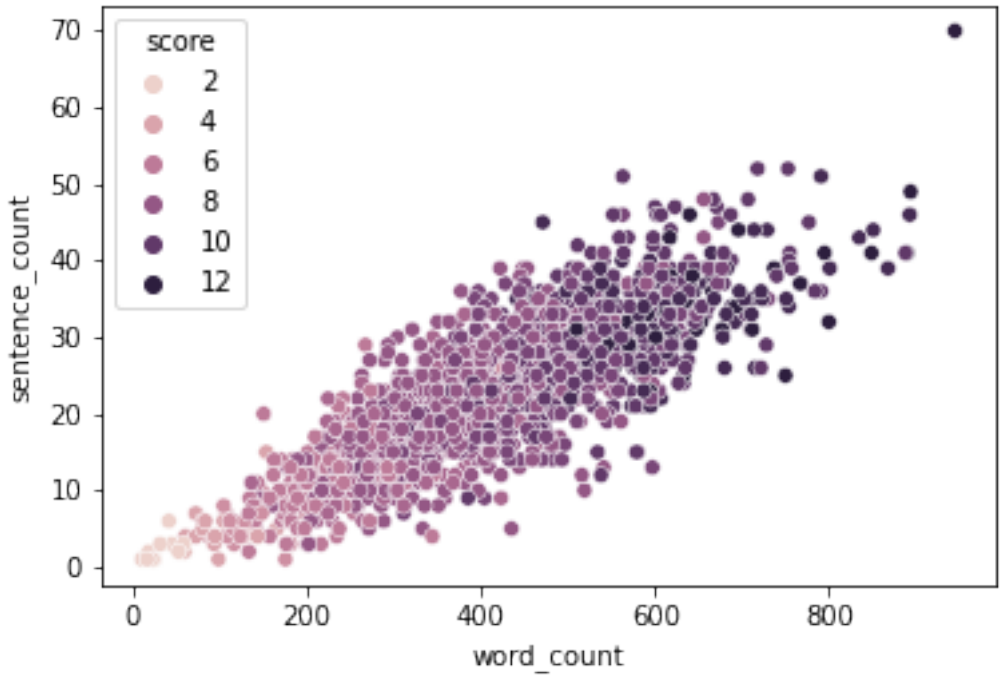



**Figura 4.1:** A título de exemplo, trouxemos uma dispersão mostrando a relação entre as quantidade de palavras, sentenças e as notas obtidas para o primeiro conjunto de redações. No Apêndice D trazemos essa mesma visão para os outros conjuntos.

(No Apêndice D, trazemos essa mesma visão para todos os conjuntos de redação). A conclusão que conseguimos tirar dessas visões é de que redações no quadrante superior direito, ou seja, aquelas com muitas palavras, e muitas sentenças tendem a ser melhor avaliadas. Tal conclusão é interessante, pois vai de encontro ao apontado em Adamson et al. [2014]. Nesse artigo, os autores apontam uma correlação entre o número de palavras únicas e a nota obtida pelo estudante e acreditam que tal efeito não seja causado pelo tamanho da redação, argumentando

que ao remover a *feature* contagem de palavras, não foi observada redução significativa do desempenho preditivo. Acreditamos que o autor não percebeu tal efeito, pois removeu a *feature* contagem de palavras, mas não removeu a *feature* contagem de sentenças. Nas visualizações de dispersão, poderemos perceber que há uma correlação linear notável entre as contagens de palavras e de sentenças, o que não é um resultado particularmente surpreendente.

**Análise da concordância entre as notas parciais dos dois avaliadores**

A Tabela 4.5 mostra um descritivo da quantidade de redações pela diferença entre as notas parciais dos dois avaliadores. Cabe notar que nos conjuntos 1 até 6, há uma quantidade muito baixa de casos nos quais a diferença entre as notas foi maior que 1. Nos conjuntos 7 e 8, observam-se diferenças maiores com maior frequência, mas é importante lembrar que o intervalo de notas possível é maior que nos outros conjuntos.

**Tabela 4.5:** Quantidade de redações por diferença absoluta entre as notas parciais dos dois avaliadores.

| Conjunto | 0 | 1 | 2 | 3 | 4 | >4 |
|---|---|---|---|---|---|---|
| 1 | 1165.0 | 596.0 | 22.0 | 0.0 | 0.0 | 0.0 |
| 2 | 1410.0 | 386.0 | 4.0 | 0.0 | 0.0 | 0.0 |
| 3 | 1292.0 | 428.0 | 3.0 | 3.0 | 0.0 | 0.0 |
| 4 | 1367.0 | 403.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 5 | 1046.0 | 722.0 | 36.0 | 1.0 | 0.0 | 0.0 |
| 6 | 1121.0 | 652.0 | 25.0 | 2.0 | 0.0 | 0.0 |
| 7 | 458.0 | 495.0 | 340.0 | 169.0 | 92.0 | 15.0 |
| 8 | 201.0 | 147.0 | 132.0 | 86.0 | 60.0 | 97.0 |

A Tabela 4.6 mostra a concordância entre os dois avaliadores medida em termos de QWK. É possível perceber que ela varia muito entre os diferentes conjuntos, indicando que possivelmente alguns conjuntos tenham critérios de correção menos sujeitos à ambiguidade, ou que o treinamento dos avaliadores tenha sido melhor.

**Tabela 4.6:** QWK entre as notas parciais dos dois julgadores para redações. Cabe pontuar que T representa conjunto de textos

| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|
| 0.721 | 0.814 | 0.769 | 0.851 | 0.753 | 0.776 | 0.721 | 0.624 |

### 4.2.2 Respostas Discursivas

Na Tabela 4.7, fizemos um compilado de algumas características de cada conjunto de respostas discursivas, como a média de palavras, o desvio padrão do número de palavras, a média do número de palavras únicas e a média do número de sentenças.

**Tabela 4.7:** A tabela traz breves estatísticas sobre os conjuntos de dados de respostas discursivas (STD se refere ao desvio padrão da quantidade de palavras e UW, a quantidade de palavras únicas).

| Conjunto | Textos | Média | STD | Média-UW | Média sentenças |
|---|---|---|---|---|---|
| 1 | 1672 | 51.6 | 24.2 | 35.3 | 2.6 |
| 2 | 1278 | 65.2 | 24.7 | 45.8 | 3.3 |
| 3 | 1891 | 52.7 | 16.5 | 38.4 | 2.8 |
| 4 | 1738 | 45.5 | 17.9 | 34.6 | 2.3 |
| 5 | 1795 | 28.5 | 24.6 | 19.5 | 2.3 |
| 6 | 1797 | 26.3 | 23.8 | 18.2 | 2.1 |
| 7 | 1799 | 45.3 | 27.1 | 33.7 | 2.5 |
| 8 | 1799 | 57.7 | 35.9 | 40.0 | 3.0 |
| 9 | 1798 | 54.8 | 40.1 | 38.3 | 3.0 |
| 10 | 1640 | 47.4 | 34.4 | 31.9 | 1.7 |

Na Tabela 4.8, detalhamos para cada conjunto, a quantidade de respostas discursivas que obtiveram cada conceito.

**Tabela 4.8:** Quantidade de respostas discursivas que receberam cada conceito. Cabe pontuar que C representa conceito e T representa conjunto de textos,

| C | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 380 | 168 | 451 | 669 | 1391 | 1515 | 932 | 549 | 434 | 290 |
| 1 | 429 | 326 | 999 | 937 | 328 | 160 | 448 | 473 | 742 | 770 |
| 2 | 524 | 467 | 441 | 132 | 42 | 71 | 419 | 777 | 622 | 580 |
| 3 | 339 | 317 | 0 | 0 | 34 | 51 | 0 | 0 | 0 | 0 |

Na Figura 4.2, trazemos uma visão, através de um gráfico de caixa, sobre a distribuição das quantidades de palavras em respostas discursivas que obtiveram cada conceito.

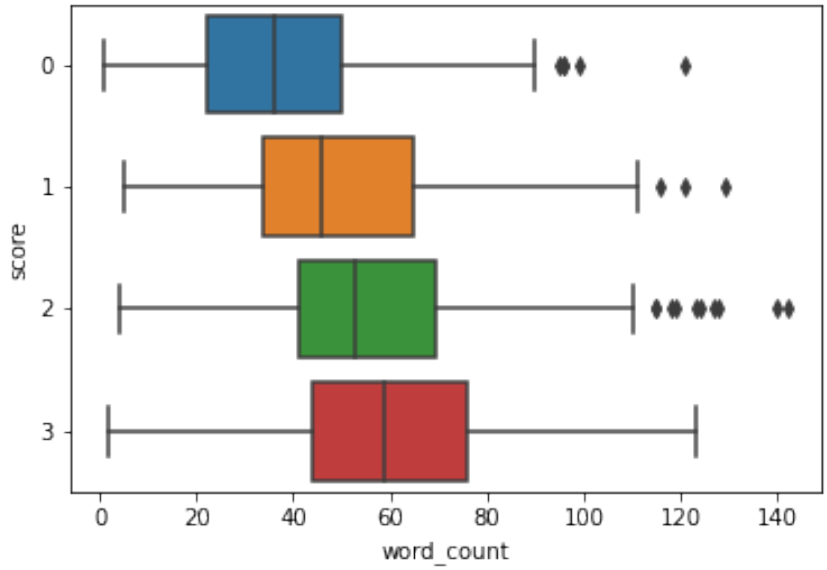



**Figura 4.2:** A título de exemplo, trouxemos um diagrama de caixa mostrando a distribuição de palavras por cada conceito para o primeiro conjunto de respostas discursivas, distribuições de palavras por conceito para outros conjuntos podem ser achadas no Apêndice E

.

(No Apêndice E trazemos essa mesma visão para todos os conjuntos de respostas discursivas). Essas visões nos mostram que as respostas discursivas avaliadas com conceitos maiores tendem a ser mais extensas em número de palavras.

**Análise da concordância entre as notas parciais dos dois avaliadores**

A Tabela 4.9 mostra um descritivo da quantidade de respostas discursivas pela diferença entre as notas parciais dos dois avaliadores, observamos que na grande maioria dos casos não há discordância, com uma minoria relevante de casos em que há discordância de 1 ponto.

**Tabela 4.9:** Quantidade de respostas discursivas por diferença absoluta entre as notas parciais dos dois avaliadores. Cabe pontuar que T representa conjunto de textos e D representa a diferença entre as notas atribuídas.

| D | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1495 | 1091 | 1439 | 1359 | 1733 | 1742 | 1726 | 1511 | 1454 | 1451 |
| 1 | 166 | 179 | 450 | 375 | 62 | 52 | 72 | 259 | 339 | 189 |
| 2 | 11 | 8 | 2 | 4 | 0 | 2 | 1 | 29 | 5 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | |

A Tabela 4.10 mostra a concordância entre as notas parciais medidas em termos de QWK. Cabe pontuar que os conjuntos 3 e 4 os quais têm, inclusive o mesmo texto base apresentam uma discordância entre os dois avaliadores visivelmente maior que os outros conjuntos.

**Tabela 4.10:** QWK entre as notas parciais dos dois julgadores para respostas discursivas.Cabe pontuar que T representa conjunto de textos.

| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.944 | 0.914 | 0.734 | 0.698 | 0.953 | 0.956 | 0.968 | 0.856 | 0.828 | 0.884 |

## 4.3 Resultados

Essa seção se destina à apresentação dos resultados obtidos em diferentes instâncias do *pipeline*. Como desejamos fazer uma análise comparativa das técnicas de representação vetorial e das abordagens de previsão, temos algumas instâncias de nosso *pipeline*, as quais possuem em comum apenas a etapa de pré-processamento. Há dois componentes que podem ser combinados de modo a variar as instâncias que são o módulo de extração de características, o qual é implementado em 4 versões diferentes e o módulo de treinamento dos modelos, o qual é implementado usando 3 abordagens diferentes a cada qual corresponderá uma abordagem de avaliação. Considerando as 4 abordagens de representação vetorial e 3 abordagens de treinamento e previsão empregadas, é possível ter 12 variações diferentes do *pipeline* proposto.

A Seção 4.3.1 apresenta considerações sobre os resultados obtidos através das diferentes abordagens de previsão avaliadas: Regressão, Classificação e Classificação ordinal. A Seção 4.3.2 apresenta, para cada uma das técnicas de representação vetorial avaliadas, uma visão sobre o desempenho através das diferentes dimensionalidades. Finalmente, a Seção 4.3.3 apresenta uma comparação das diferentes técnicas de representação vetorial empregadas nesse trabalho.

### 4.3.1 Comparação entre as abordagens de previsão

#### Resultados para Redações

Cada uma das Tabelas: 4.11, 4.12, 4.13 , 4.14 e 4.15 apresenta os resultados obtidos considerando uma dimensionalidade de representação vetorial: respectivamente, 32, 64, 128, 256 e 512. Como as dimensionalidades usadas para o LSI diferem das dimensionalidades usadas para as outras técnicas de representação vetorial mostraremos os resultados envolvendo essa abordagem na Tabela 4.16.

**Tabela 4.11:** A tabela apresenta os resultados em redações em que a representação vetorial empregada tem dimensionalidade 32,Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| C | DOC 2 VEC | 0.72 | 0.60 | 0.69 | 0.70 | 0.77 | 0.65 | 0.67 | 0.49 |
| CO | DOC 2 VEC | 0.00 | 0.04 | 0.42 | 0.19 | 0.25 | 0.20 | 0.01 | 0.00 |
| R | DOC 2 VEC | 0.84 | 0.64 | 0.69 | 0.72 | 0.80 | 0.74 | 0.75 | 0.63 |
| C | TF IDF | 0.76 | 0.72 | 0.73 | 0.76 | 0.83 | 0.69 | 0.70 | 0.40 |
| CO | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.00 |
| R | TF IDF | **0.88** | **0.78** | **0.76** | **0.80** | **0.86** | **0.80** | **0.80** | **0.72** |

**Tabela 4.12:** A tabela apresenta os resultados em redações em que a representação vetorial empregada tem dimensionalidade 64,Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| C | DOC 2 VEC | 0.67 | 0.58 | 0.70 | 0.69 | 0.77 | 0.66 | 0.66 | 0.42 |
| CO | DOC 2 VEC | 0.00 | 0.04 | 0.40 | 0.18 | 0.23 | 0.18 | 0.00 | 0.00 |
| R | DOC 2 VEC | 0.84 | 0.64 | 0.69 | 0.73 | 0.79 | 0.74 | 0.76 | 0.61 |
| C | TF IDF | 0.73 | 0.72 | 0.73 | 0.77 | 0.82 | 0.69 | 0.68 | 0.39 |
| CO | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.00 |
| R | TF IDF | **0.88** | **0.79** | **0.77** | **0.81** | **0.86** | **0.81** | **0.81** | **0.74** |

**Tabela 4.13:** A tabela apresenta os resultados em redações em que a representação vetorial empregada tem dimensionalidade 128,Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| C | DOC 2 VEC | 0.54 | 0.50 | 0.70 | 0.67 | 0.76 | 0.59 | 0.59 | 0.22 |
| CO | DOC 2 VEC | 0.00 | 0.00 | 0.39 | 0.18 | 0.24 | 0.14 | 0.00 | 0.00 |
| R | DOC 2 VEC | 0.83 | 0.62 | 0.69 | 0.70 | 0.80 | 0.73 | 0.75 | 0.57 |
| C | TF IDF | 0.63 | 0.69 | 0.74 | 0.75 | 0.82 | 0.73 | 0.62 | 0.34 |
| CO | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.00 |
| R | TF IDF | **0.88** | **0.79** | **0.77** | **0.82** | **0.86** | **0.82** | **0.81** | **0.76** |

**Tabela 4.14:** A tabela apresenta os resultados em redações em que a representação vetorial empregada tem dimensionalidade 256,Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| C | DOC 2 VEC | 0.50 | 0.53 | 0.68 | 0.66 | 0.74 | 0.55 | 0.52 | 0.33 |
| CO | DOC 2 VEC | 0.00 | 0.00 | 0.36 | 0.13 | 0.21 | 0.13 | 0.00 | 0.00 |
| R | DOC 2 VEC | 0.84 | 0.61 | 0.69 | 0.71 | 0.79 | 0.73 | 0.74 | 0.56 |
| C | TF IDF | 0.60 | 0.64 | 0.74 | 0.73 | 0.81 | 0.73 | 0.56 | 0.28 |
| CO | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.00 |
| R | TF IDF | **0.89** | **0.79** | **0.77** | **0.82** | **0.86** | **0.82** | **0.81** | **0.76** |

**Tabela 4.15:** A tabela apresenta os resultados em redações em que a representação vetorial empregada tem dimensionalidade 512,Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| C | DOC 2 VEC | 0.42 | 0.44 | 0.67 | 0.66 | 0.71 | 0.53 | 0.46 | 0.25 |
| CO | DOC 2 VEC | 0.00 | 0.00 | 0.35 | 0.08 | 0.17 | 0.11 | 0.00 | 0.00 |
| R | DOC 2 VEC | 0.83 | 0.61 | 0.70 | 0.71 | 0.81 | 0.73 | 0.74 | 0.56 |
| C | TF IDF | 0.53 | 0.61 | 0.73 | 0.68 | 0.78 | 0.64 | 0.46 | 0.23 |
| CO | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.00 |
| R | TF IDF | **0.88** | **0.78** | **0.76** | **0.82** | **0.86** | **0.82** | **0.82** | **0.77** |
| C | USE | 0.58 | 0.58 | 0.70 | 0.72 | 0.75 | 0.66 | 0.57 | 0.29 |
| CO | USE | 0.00 | 0.02 | 0.38 | 0.13 | 0.20 | 0.17 | 0.00 | 0.00 |
| R | USE | 0.83 | 0.62 | 0.70 | 0.75 | 0.79 | 0.75 | 0.76 | 0.50 |

**Tabela 4.16:** A tabela apresenta os resultados em redações em que o embedding empregado usa o LSI, Nas colunas T representa texto, D representa a dimensionalidade do LSI usada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | D |
|---|---|---|---|---|---|---|---|---|---|
| C | 0.78 | 0.58 | **0.69** | **0.70** | 0.76 | 0.62 | 0.65 | 0.40 | 10 |
| CO | 0.00 | 0.02 | 0.38 | 0.18 | 0.24 | 0.19 | 0.01 | 0.00 | 10 |
| R | 0.82 | 0.57 | 0.68 | 0.66 | 0.76 | 0.56 | **0.75** | 0.51 | 10 |
| C | 0.77 | 0.56 | **0.69** | **0.70** | 0.77 | 0.61 | 0.65 | 0.39 | 20 |
| CO | 0.00 | 0.02 | 0.38 | 0.19 | 0.23 | 0.16 | 0.01 | 0.00 | 20 |
| R | 0.82 | **0.60** | **0.69** | **0.70** | **0.78** | 0.60 | **0.75** | 0.44 | 20 |
| C | 0.76 | 0.55 | 0.68 | 0.67 | 0.77 | 0.65 | 0.67 | 0.42 | 30 |
| CO | 0.00 | 0.00 | 0.39 | 0.18 | 0.24 | 0.18 | 0.01 | 0.00 | 30 |
| R | **0.83** | 0.57 | 0.68 | 0.63 | **0.78** | 0.65 | **0.75** | 0.51 | 30 |
| C | 0.75 | 0.52 | 0.67 | 0.68 | 0.76 | 0.61 | 0.61 | 0.31 | 40 |
| CO | 0.00 | 0.00 | 0.38 | 0.18 | 0.23 | 0.19 | 0.01 | 0.00 | 40 |
| R | 0.82 | 0.57 | 0.68 | 0.66 | 0.76 | 0.62 | 0.66 | 0.45 | 40 |
| C | 0.75 | 0.57 | 0.67 | 0.68 | 0.77 | 0.59 | 0.61 | 0.29 | 50 |
| CO | 0.00 | 0.00 | 0.40 | 0.20 | 0.24 | 0.18 | 0.01 | 0.00 | 50 |
| R | 0.82 | **0.60** | 0.68 | 0.67 | 0.75 | **0.66** | 0.65 | **0.52** | 50 |
| C | 0.64 | 0.54 | 0.64 | 0.62 | 0.72 | 0.56 | 0.55 | 0.24 | 100 |
| CO | 0.00 | 0.00 | 0.35 | 0.12 | 0.14 | 0.07 | 0.00 | 0.00 | 100 |
| R | 0.82 | 0.50 | 0.68 | 0.61 | 0.77 | **0.66** | 0.37 | 0.48 | 100 |

Ao analisar os resultados apresentados nas Tabelas: 4.11, 4.12, 4.13 , 4.14, 4.15 e 4.16. é visível que os resultados obtidos nas instâncias do *Pipeline* que usam a abordagem de classificação ordinal obtém um desempenho baixo, de forma geral. Observamos que dentro dos mesmos métodos, não houve impacto relevante da dimensionalidade no desempenho preditivo e observamos também, que o TF-IDF obteve melhores resultados quando comparado à outros métodos. Cabe pontuar que é nítida a diferença de desempenho através dos diferentes conjuntos de texto, em particular, um padrão que se repete em todas as técnicas e dimensionalidades é a presença de um maior desempenho nos conjuntos 3 a 6, enquanto os conjuntos 1,2,7 e 8 apresentam desempenhos menos expressivos. Uma hipótese explicativa é a quantidade maior de classes nos conjuntos 1,7 e 8, mas ela não explica o desempenho no conjunto 2, então a hipótese que acreditamos mais aderente é a distribuição das redações pelas notas, em que nos conjuntos 1,2,7 e 8 há um numero importante de classes com poucos registros, situação que não se apresenta nos conjuntos de 3 a 6.

A Tabela G.5 apresenta os resultados obtidos nas instâncias do *Pipeline* que usam a abordagem de classificação na tarefa de avaliação de redações, enquanto a Tabela G.6 apresenta os resultados obtidos nas instâncias do *Pipeline* que usam a abordagem de regressão nessa mesma tarefa. Para comparar as abordagens acima mencionadas, utilizamos a seguinte lógica, pareamos os resultados em que no *pipeline* foi usada a mesma técnica de representação vetorial, com mesma dimensionalidade, para o mesmo texto e calculamos a diferença entre os resultados obtidos. Para fazer isso, primeiro montamos uma tabela com mesma organização das tabelas acima mencionadas por meio da subtração dos valores das tabelas G.5 e G.6. Não exibiremos essa tabela em nosso texto, para não poluir a explicação. A partir dessa tabela, para cada texto (coluna), calculamos a média das diferenças e obtivemos uma sumarização que é exibida na Tabela 4.17.

**Tabela 4.17:** A tabela mostra uma visão comparativa da diferença média de desempenho preditivo obtido entre as abordagens de classificação e regressão através dos diferentes conjuntos de redações. Cabe ressaltar que, nas colunas T significa texto, e no campo Medida M representa média e STD desvio padrão.

| Medida | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|
| M | -0.189 | -0.068 | -0.014 | -0.028 | -0.034 | -0.087 | -0.132 | -0.259 |
| STD | 0.117 | 0.059 | 0.018 | 0.046 | 0.029 | 0.070 | 0.118 | 0.138 |

É possível perceber que, em nenhum dos conjuntos de texto, a abordagem de classificação obteve melhor desempenho preditivo. Cabe pontuar também, que a diferença em favor da re-

gressão se mostra mais pronunciada nos conjuntos 1,7 e 8. Uma hipótese explicativa é de que essa diferença se explique pelo fato de tais conjuntos terem mais valores possíveis de nota, o que traz desvantagem à abordagem de classificação pela sua incapacidade de levar em conta a informação de ordem entre as classes.

### Resultados para Respostas discursivas

Cada uma das Tabelas: 4.18, 4.19 , 4.20 , 4.21 e 4.22 apresenta os resultados obtidos considerando uma dimensionalidade de representação vetorial: respectivamente, 32,64,128,256 e 512. Como as dimensionalidades usadas para o LSI diferem das dimensionalidades usadas para as outras técnicas de representação vetorial mostraremos os resultados envolvendo essa abordagem na Tabela 4.23.

**Tabela 4.18:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada tem dimensionalidade 32, Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | DOC2VEC | 0.54 | 0.25 | 0.01 | 0.47 | 0.31 | 0.20 | 0.27 | 0.42 | 0.62 | 0.49 |
| CO | DOC2VEC | 0.01 | 0.04 | 0.01 | 0.02 | 0.01 | 0.00 | 0.11 | 0.30 | 0.45 | 0.43 |
| R | DOC2VEC | 0.47 | 0.22 | 0.00 | 0.50 | 0.52 | 0.69 | 0.29 | 0.33 | 0.59 | 0.54 |
| C | TFIDF | **0.67** | **0.53** | 0.07 | 0.48 | 0.59 | 0.66 | 0.52 | 0.56 | **0.72** | 0.62 |
| CO | TFIDF | 0.34 | 0.45 | **0.66** | 0.34 | 0.03 | 0.04 | 0.54 | **0.71** | **0.72** | **0.77** |
| R | TFIDF | 0.63 | 0.49 | 0.03 | **0.56** | **0.76** | **0.84** | **0.58** | 0.58 | **0.72** | 0.69 |

**Tabela 4.19:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada tem dimensionalidade 64, Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | DOC2VEC | 0.58 | 0.24 | 0.00 | 0.46 | 0.38 | 0.09 | 0.23 | 0.42 | 0.57 | 0.50 |
| CO | DOC2VEC | -0.00 | 0.04 | 0.02 | 0.02 | 0.00 | 0.00 | 0.16 | 0.31 | 0.45 | 0.48 |
| R | DOC2VEC | 0.47 | 0.24 | 0.00 | 0.50 | 0.51 | 0.63 | 0.22 | 0.36 | 0.59 | 0.48 |
| C | TFIDF | **0.75** | **0.55** | 0.06 | 0.51 | 0.58 | 0.63 | 0.54 | 0.55 | 0.75 | 0.64 |
| CO | TFIDF | 0.34 | 0.45 | **0.66** | 0.34 | 0.03 | 0.04 | 0.54 | **0.71** | 0.72 | **0.77** |
| R | TFIDF | 0.74 | 0.48 | 0.02 | **0.57** | **0.77** | **0.85** | **0.61** | 0.60 | **0.76** | 0.71 |

**Tabela 4.20:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada tem dimensionalidade 128, Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação,CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | DOC2VEC | 0.61 | 0.18 | 0.00 | 0.42 | 0.22 | 0.11 | 0.16 | 0.40 | 0.59 | 0.52 |
| CO | DOC2VEC | 0.02 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.04 | 0.33 | 0.46 | 0.46 |
| R | DOC2VEC | 0.52 | 0.22 | 0.00 | 0.50 | 0.54 | 0.62 | 0.22 | 0.33 | 0.57 | 0.52 |
| C | TFIDF | **0.74** | **0.50** | 0.04 | 0.51 | 0.54 | 0.55 | 0.48 | 0.54 | **0.74** | 0.60 |
| CO | TFIDF | 0.34 | 0.45 | **0.66** | 0.34 | 0.03 | 0.04 | 0.54 | **0.71** | 0.72 | **0.77** |
| R | TFIDF | 0.73 | 0.46 | 0.01 | **0.59** | **0.77** | **0.86** | **0.63** | 0.61 | 0.76 | 0.70 |

**Tabela 4.21:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada tem dimensionalidade 256, Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação, CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | DOC2VEC | 0.59 | 0.30 | 0.01 | 0.48 | 0.36 | 0.21 | 0.20 | 0.42 | 0.58 | 0.47 |
| CO | DOC2VEC | 0.00 | 0.04 | 0.03 | 0.00 | 0.00 | 0.00 | 0.07 | 0.31 | 0.41 | 0.41 |
| R | DOC2VEC | 0.45 | 0.21 | 0.00 | 0.54 | 0.53 | 0.71 | 0.27 | 0.38 | 0.60 | 0.47 |
| C | TFIDF | **0.73** | 0.45 | 0.00 | 0.50 | 0.46 | 0.39 | 0.42 | 0.54 | 0.73 | 0.59 |
| CO | TFIDF | 0.34 | 0.44 | **0.66** | 0.34 | 0.03 | 0.04 | 0.54 | **0.71** | 0.72 | **0.77** |
| R | TFIDF | **0.73** | **0.46** | 0.01 | **0.57** | **0.78** | **0.86** | **0.64** | 0.61 | **0.77** | 0.69 |

**Tabela 4.22:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada tem dimensionalidade 512, Nas colunas T representa texto, M representa a técnica de representação vetorial empregada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação, CO classificação ordinal e R regressão.

| A | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | DOC2VEC | 0.59 | 0.23 | 0.00 | 0.48 | 0.32 | 0.21 | 0.21 | 0.44 | 0.58 | 0.52 |
| CO | DOC2VEC | 0.02 | 0.04 | 0.02 | 0.02 | 0.00 | 0.00 | 0.10 | 0.34 | 0.45 | 0.47 |
| R | DOC2VEC | 0.49 | 0.25 | 0.00 | 0.50 | 0.57 | 0.69 | 0.31 | 0.38 | 0.55 | 0.53 |
| C | TFIDF | **0.74** | 0.44 | 0.01 | 0.49 | 0.47 | 0.47 | 0.40 | 0.53 | 0.70 | 0.58 |
| CO | TFIDF | 0.34 | 0.45 | **0.66** | 0.34 | 0.03 | 0.04 | 0.54 | **0.71** | 0.72 | **0.77** |
| R | TFIDF | 0.73 | **0.49** | 0.01 | 0.57 | **0.78** | **0.77** | **0.65** | 0.60 | **0.77** | 0.70 |
| C | USE | 0.66 | 0.29 | 0.00 | **0.59** | 0.44 | 0.37 | 0.28 | 0.47 | 0.69 | 0.62 |
| CO | USE | 0.06 | 0.03 | 0.09 | 0.03 | 0.00 | 0.01 | 0.13 | 0.32 | 0.48 | 0.51 |
| R | USE | 0.65 | 0.28 | 0.00 | 0.57 | 0.68 | 0.73 | 0.35 | 0.43 | 0.67 | 0.61 |

**Tabela 4.23:** A tabela apresenta os resultados em respostas curtas em que a representação vetorial empregada usa o LSI, Nas colunas T representa texto, D representa a dimensionalidade do LSI usada e A representa a abordagem de previsão usada. Nos valores de A, C representa classificação, CO classificação ordinal e R regressão.

| A | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | D |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 0.36 | 0.21 | 0.00 | **0.42** | 0.16 | 0.11 | 0.16 | 0.34 | **0.54** | **0.48** | 10 |
| CO | 0.01 | 0.04 | 0.01 | 0.03 | 0.00 | 0.01 | 0.06 | 0.21 | 0.40 | 0.37 | 10 |
| R | 0.36 | 0.22 | 0.00 | **0.42** | 0.25 | 0.42 | 0.16 | 0.23 | 0.50 | 0.40 | 10 |
| C | 0.14 | 0.18 | -0.01 | 0.39 | 0.24 | 0.01 | 0.14 | 0.34 | **0.54** | 0.40 | 20 |
| CO | 0.01 | -0.02 | -0.01 | 0.04 | 0.00 | 0.00 | 0.09 | 0.26 | 0.41 | 0.33 | 20 |
| R | 0.07 | 0.21 | 0.00 | **0.42** | **0.39** | 0.30 | 0.11 | 0.27 | 0.51 | 0.39 | 20 |
| C | 0.38 | 0.19 | 0.00 | 0.40 | 0.09 | 0.00 | 0.12 | 0.34 | **0.54** | 0.37 | 30 |
| CO | 0.01 | 0.03 | -0.01 | 0.03 | 0.00 | -0.00 | 0.05 | 0.24 | 0.39 | 0.32 | 30 |
| R | 0.34 | 0.18 | 0.00 | 0.39 | 0.25 | 0.35 | 0.13 | 0.29 | 0.51 | 0.29 | 30 |
| C | 0.17 | 0.19 | 0.00 | 0.39 | 0.22 | 0.02 | 0.03 | 0.37 | 0.53 | 0.37 | 40 |
| CO | 0.01 | 0.02 | **0.03** | 0.02 | 0.00 | 0.01 | 0.02 | 0.29 | 0.38 | 0.35 | 40 |
| R | 0.12 | **0.23** | 0.00 | 0.41 | 0.31 | 0.26 | 0.10 | 0.29 | 0.48 | 0.40 | 40 |
| C | **0.39** | 0.12 | 0.00 | 0.36 | 0.20 | 0.02 | 0.04 | **0.39** | 0.50 | 0.34 | 50 |
| CO | 0.01 | 0.01 | -0.03 | 0.01 | 0.00 | 0.01 | 0.01 | 0.25 | 0.38 | 0.32 | 50 |
| R | 0.35 | 0.00 | 0.00 | 0.39 | 0.25 | **0.43** | 0.08 | 0.33 | 0.45 | 0.31 | 50 |
| C | 0.25 | 0.14 | 0.00 | 0.41 | -0.01 | 0.00 | -0.01 | 0.33 | 0.46 | 0.41 | 100 |
| CO | 0.00 | 0.01 | 0.02 | 0.00 | 0.00 | 0.00 | 0.01 | 0.17 | 0.32 | 0.35 | 100 |
| R | 0.14 | -0.01 | 0.00 | 0.37 | 0.27 | 0.41 | **0.21** | 0.13 | 0.23 | 0.42 | 100 |

Ao analisar as Tabelas 4.18, 4.19, 4.20, 4.21, 4.22 e 4.23, podemos notar que a abordagem de classificaçao ordinal obtém, na grande maioria dos casos, um desempenho preditivo bastante baixo, com valores de QWK geralmente menores que 0.1. Cabe notar, porém, dois cenários onde essa tendência não se mostra, a avaliação dos conjuntos 8, 9 e 10, em que os resultados são visivelmente melhores que na avaliação dos outros conjuntos e o cenário em que a técnica de representação vetorial empregada é a TF-IDF. É interessante pontuar que quando os dois cenários se intersetam, ou seja avalia-se o uso da técnica de representação vetorial TF-IDF para prever os resultados nos conjunto 8, 9 e 10, os resultados se mostram interessantemente positivos com valores acima de 0.6. Outro fato interessante é que ao usar a abordagem de classificação ordinal e usar a técnica de representação vetorial TF-IDF, o desempenho não varia conforme a dimensionalidade empregada, Há que se realizar estudos para entender o motivo desse fenômeno, mas uma possibilidade é que no treinamento dos modelos para os problemas de classificação binária o conjunto de features que realmente agrega valor preditivo esteja contido entre as 32 *features* do modelo de menor dimensionalidade, e, portanto as outras *features* não agreguem valor preditivo.

A Tabela G.2 apresenta os resultados obtidos nas instâncias do *Pipeline* que usam a abordagem de classificação na tarefa de avaliação de respostas discursivas, enquanto a Tabela G.3 apresenta os resultados obtidos nas instâncias do *Pipeline* que usam a abordagem de regressão nessa mesma tarefa. Para comparar as abordagens acima mencionadas, utilizamos a seguinte lógica, pareamos os resultados em que no *pipeline* foi usada a mesma técnica de representação vetorial, com mesma dimensionalidade, para o mesmo texto e calculamos a diferença entre os resultados obtidos. Para fazer isso, primeiro montamos uma tabela com mesma organização das tabelas acima mencionadas por meio da subtração dos valores das tabelas G.2 e G.3, não exibiremos essa tabela em nosso texto, para não poluir a explicação. A partir dessa tabela, para cada texto (coluna) calculamos a média das diferenças e obtivemos uma sumarização que é exibida na Tabela 4.24.

**Tabela 4.24:** A tabela mostra uma visão comparativa da diferença média de desempenho preditivo obtido entre as abordagens de classificação e regressão através dos diferentes conjuntos de respostas discursivas. Cabe ressaltar que, nas colunas T significa texto, e no campo Medida M representa média e STD desvio padrão.

| Medida | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| M | 0.056 | 0.017 | 0.009 | -0.059 | -0.276 | -0.302 | -0.105 | 0.009 | -0.002 | -0.008 |
| STD | 0.044 | 0.057 | 0.015 | 0.043 | 0.084 | 0.109 | 0.083 | 0.073 | 0.063 | -0.05 |

Diferentemente do observado na tarefa de avaliação de redações, na tarefa de avaliação de respostas discursivas, a vantagem pronunciada da regressão se restringe aos conjuntos 5,6 e 7. Analisando a distribuição das respostas discursivas por conceito mostrada na Tabela 4.8, podemos perceber que, em particular, os conjuntos 5 e 6 mostram uma grande concentração de respostas em um dos conceitos (superior a 75%). Uma possibilidade é que a dificuldade experimentada nos modelos seja causada pelo imbalanço entre as classes.

### 4.3.2 Análise do impacto da dimensionalidade da representação vetorial no desempenho preditivo

Para cada técnica de representação vetorial empregada nesse trabalho (à exceção do USE), usamos diferentes dimensionalidades, o objetivo é verificar a existência de tendência na relação entre a dimensionalidade e desempenho preditivo. Para essa análise, não consideramos a técnica USE, pois ela se apresenta em apenas uma dimensionalidade, e também não considera-

mos os resultados obtidos com a abordagem de classificação ordinal, pois, conforme mostrado anteriormente na Seção 4.3.1 os resultados obtidos com o uso dessa abordagem mostram um desempenho preditivo bastante reduzido.

### Redações

A Figura 4.3 apresenta uma visão sobre o impacto da dimensionalidade das técnicas de representação vetorial empregadas no desempenho preditivo na tarefa de avaliação de redações. Cabe lembrar que os resultados gerados em nosso trabalho contemplam técnicas de representação vetorial, dimensionalidades das técnicas de representação vetorial, abordagens de previsão e textos. Cada "caixa" nas figuras representa o conjunto de resultados obtidos com uma técnica de representação vetorial aplicada com mesma dimensionalidade. Dentro de cada conjunto, há 24 elementos, representando a aplicação de uma das 3 abordagens de previsão avaliadas a um dos 8 conjuntos de redação.

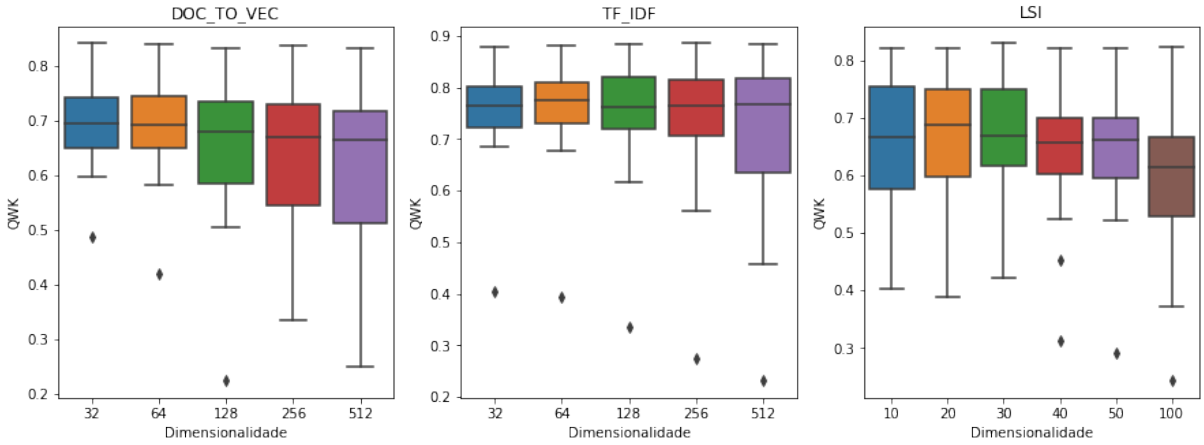



**Figura 4.3:** Diagramas de caixa mostrando, respectivamente para o *Doc2vec*, TF-IDF e LSI, a dispersão de valores de QWK obtidos em instancias do pipeline que utilizaram diferentes dimensionalidades da técnica de representação vetorial à qual se relaciona o diagrama. Cabe lembrar que a avaliação mostrada é referente à tarefa de avaliação de redações. Vale salientar que cada "caixa" nas figuras representa o conjunto de resultados obtidos com uma técnica de representação vetorial aplicada com mesma dimensionalidade. Dentro de cada conjunto, há 24 elementos, representando a aplicação de uma das 3 abordagens de previsão avaliadas a um dos 8 conjuntos de redação.

Ao analisar os resultados apresentados na Figura 4.3, podemos observar dois comportamentos acontecendo de acordo com a variação da dimensionalidade: na abordagem USE e na abordagem *Doc2vec* a distribuição dos resultados para valores acima da mediana não mostra alteração visível, porém a distribuição dos resultados para valores abaixo da mediana mostra um cenário em que valores mais distantes da mediana e portanto mais baixos, tornam-se mais

comuns, como isso não acontece para valores acima da mediana, uma consequência dedutível é que a distribuição torna-se mais assimétrica. O outro comportamento é o do LSI, em que as dimensionalidades mais altas têm suas caixas posicionadas abaixo das caixas das dimensionalidades mais baixas. Tal comportamento torna-se patente ao observar a versão do LSI com 100 tópicos, em que é possível ver a mediana visivelmente distante das medianas das outras dimensionalidades.

### Respostas discursivas

A Figura 4.4 apresenta uma visão sobre o impacto da dimensionalidade das técnicas de representação vetorial empregadas no desempenho preditivo na tarefa de avaliação de respostas discursivas.

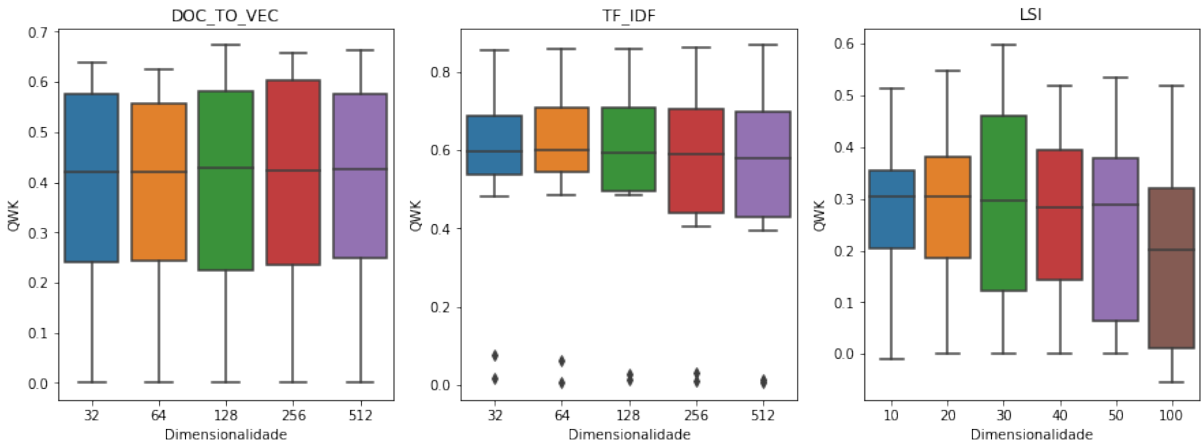



**Figura 4.4:** Diagramas de caixa mostrando, respectivamente para o *Doc2vec*, TF-IDF e LSI, a dispersão de valores de QWK obtidos em instancias do pipeline que utilizaram diferentes dimensionalidades da técnica de representação vetorial à qual se relaciona o diagrama. Cabe lembrar que a avaliação mostrada é referente à tarefa de avaliação de respostas discursivas. Cabe pontuar também, que cada 'caixa' do diagrama foi construída através de diferentes abordagens de modelagem e diferentes textos.

Ao analisar os resultados apresentados na Figura 4.4, observamos um panorama diferente do apresentado na tarefa de avaliação de redações. Podemos perceber, que nos experimentos realizados usando a técnica de representação vetorial *Doc2vec*, a amplitude do desempenho preditivo através dos conjuntos de texto e das abordagens de previsão avaliadas (regressão e classificação) é grande. Cabe pontuar que não houve diferença visível entre os desempenhos através das dimensionalidades. Nos experimentos envolvendo a técnica TF-IDF, podemos observar o mesmo comportamento de aumento da assimetria observado na tarefa de avaliação de redações. Finalmente, nos experimentos envolvendo a técnica LSI, a amplitude através das

abordagens de previsão e dos conjuntos de texto também se mostra bastante elevada. Não se nota diferença visível entre as medianas, com exceção do experimento com 100 tópicos.

### 4.3.3 Comparação das técnicas de representação vetorial

**Redações**

A Figura 4.5 apresenta os resultados obtidos para diferentes técnicas de representação vetorial na tarefa de avaliação de redações

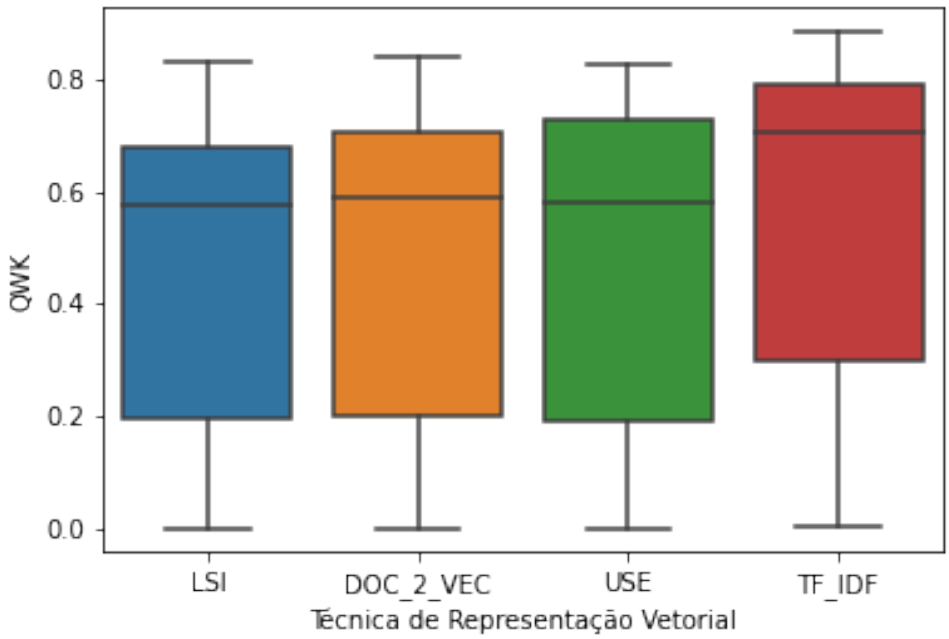



**Figura 4.5:** Diagrama de caixa mostrando a dispersão do desempenho preditivo na tarefa de avaliação de redações obtido com cada técnica de representação vetorial através das abordagens de previsão, dimensionalidades e diferentes conjuntos de texto.

Ao analisar os resultados mostrados na Figura 4.5, podemos perceber que entre as técnicas LSI, USE e *Doc2vec*, não houve diferença visível entre os desempenhos. Por outro lado, o TF-IDF mostra desempenho visivelmente superior às outras três técnicas apresentadas. Um fator que certamente contribui para isso é o fato de que o TF-IDF não apresentou os vários resultados de QWK bastante próximos a 0 que as outras técnicas apresentaram quando usadas em conjunto com a abordagem de classificação ordinal.

**Respostas discursivas**

A Figura 4.6 apresenta os resultados obtidos para diferentes técnicas de representação vetorial na tarefa de avaliação de respostas discursivas. Cabe lembrar que os resultados gerados em nosso trabalho contemplam técnicas de representação vetorial, dimensionalidades das técnicas de representação vetorial, abordagens de previsão e textos. Cada "caixa" nas figuras representa o conjunto de resultados obtidos com uma técnica de representação vetorial aplicada com mesma dimensionalidade. Dentro de cada conjunto, há 30 elementos, representando a aplicação de uma das 3 abordagens de previsão avaliadas a um dos 10 conjuntos de respostas discursivas.

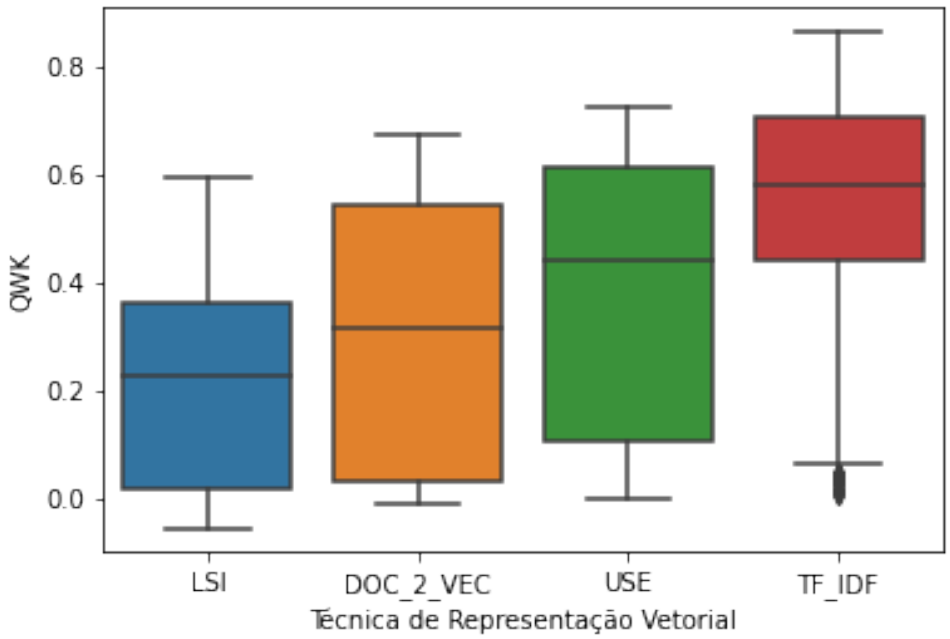



**Figura 4.6:** Diagrama de caixa mostrando a dispersão do desempenho preditivo na tarefa de avaliação de respostas discursivas obtido com cada técnica de representação vetorial. Cabe relembrar que cada "caixa" nas figuras representa o conjunto de resultados obtidos com uma técnica de representação vetorial aplicada com mesma dimensionalidade. Dentro de cada conjunto, há 30 elementos, representando a aplicação de uma das 3 abordagens de previsão avaliadas a um dos 10 conjuntos de respostas discursivas.

Ao analisar os resultados apresentados na Figura 4.6, é perceptível que há diferenças entre o desempenho das diferentes técnicas de representação vetorial. O LSI nitidamente obteve pior desempenho, quando comparado às outras técnicas, enquanto as técnicas USE e *Doc2vec* obtiveram desempenho intermediário. Cabe apontar a superioridade em desempenho preditivo mostrado pelo USE em relação ao *Doc2vec*, fato que contraria nossas expectativas prévias, dado que o *Doc2vec* é treinado com os conjuntos de texto das redações e respostas discursivas

avaliadas, enquanto o USE é treinado[1] com outros textos de domínios não necessariamente do domínio de redações ou exames avaliativos em geral. É possível que o desempenho preditivo inferior do *Doc2vec* seja explicado por não ser treinado no pipeline com uma quantidade de textos tão expressiva (treinamos com 30 mil textos, enquanto o USE é treinado com uma quantidade de exemplos na casa de milhões). Finalmente, cabe apontar o fato de que o TF-IDF mostrou desempenho superior às outras técnicas de representação vetorial avaliadas, fato bastante interessante, considerando que, entre as técnicas avaliadas, é a menor complexidade teórica e que menos exige poder computacional em nosso pipeline (principalmente porque ao avaliar diferentes dimensionalidades não precisamos fazer varias rodagens, simplesmente selecionamos as *features* com nosso método de seleção).

[1]Cabe lembrar que usamos uma versão pré-treinada do modelo, logo o treinamento não é feito por nós e sim pelos desenvolvedores

# Capítulo 5

# Conclusão

## 5.1 Análise Retrospectiva

Nesse trabalho, nos propusemos a construir um *pipeline* de avaliação automática de itens discursivos e avaliar o desempenho preditivo de acordo com a aplicação de diferentes técnicas de processamento de linguagem natural e aprendizado de máquina. Devido a questões de incompatibilidade de bibliotecas, tivemos dificuldades para reproduzir sistemas já desenvolvidos para essa tarefa, como o sistema desenvolvido por Luis Tandalla, o qual ganhou a competição ASAP-AES de 2012, evento do qual aproveitamos os textos (conjuntos de dados) usados neste trabalho. Apesar disso, conseguimos avaliar o uso de quatro técnicas de representação vetorial de texto e três abordagens de previsão.

Em nossos primeiros experimentos, construímos um *pipeline*, implementado por meio de *notebooks* Jupyter e que realizava o pré-processamento, a representação vetorial dos textos, treinava modelos e realizava as previsões. O escopo de textos analisado era somente redações e o problema era enquadrado como um problema de classificação, pois a métrica QWK não trabalha com valores não inteiros. Passamos a ter no escopo do trabalho a comparação de duas técnicas de representação vetorial: LSI e USE. Posteriormente, superamos o obstáculo para enquadrar o problema como regressão. Para fazer isso, tivemos de adicionar um passo de pós-processamento às previsões feitas, o qual aproxima a previsão para o inteiro mais próximo, também foi necessário passar a realizar validação cruzada com outra métrica que não o QWK, mas mantendo o QWK como métrica de avaliação final. O enquadramento do problema como regressão, provocou aumento no desempenho preditivo na tarefa de avaliação de redações. Em seguida, tendo visibilidade desse conjunto de resultados, passamos a analisar se o desempenho obtido pelas técnicas na avaliação de redações generalizaria para a avaliação de respostas discursivas. Essa fase mostrou a necessidade de trabalhar com um artefato de código mais aderente a boas práticas de Engenharia de Software, e portanto passamos a trabalhar com um código mais organizado e generalizável. Esse desafio se mostrou presente ao longo do trabalho, conforme incrementávamos o escopo. Fizemos então os experimentos com o *pipeline* para os conjuntos

de respostas discursivas. Nesse ponto, ainda tínhamos duas técnicas de representação vetorial, mas elas já estavam sendo avaliadas para dois tipos textuais.

Após isso, decidimos avaliar se usar um modelo de representação vetorial treinado com textos dos conjuntos de redações e respostas discursivas traria um melhor desempenho preditivo. Passamos então a usar o modelo *Doc2vec*. Em seguida, sentimos falta de uma técnica mais simples para atuar como *Baseline* frente à avaliação das outras técnicas de representação vetorial. Então fizemos experimentos usando também a representação TF-IDF. Nesse estágio do trabalho, já tínhamos 4 técnicas de representação vetorial sendo analisadas em 2 tipos textuais.

Tendo a visão do desempenho das diferentes técnicas de representação vetorial, decidimos retomar uma discussão existente no início do trabalho sobre enquadrar o problema como classificação ou regressão. Adicionamos, portanto, mais uma dimensão para variar as instâncias do *pipeline* trabalhado e fizemos experimentos usando a abordagem de classificação e experimentos usando a abordagem de regressão. Finalmente, acrescentamos experimentos usando a abordagem de classificação ordinal.

Como resultado, percebemos que os melhores desempenhos preditivos são obtidos ao abordar o problema como regressão, vantagem que é mais pronunciada na tarefa de avaliação de redações do que na tarefa de avaliação de respostas discursivas. Percebemos também que usar técnicas de representação de maior dimensionalidade não causa aumento do desempenho preditivo, tendência que foi observada em ambas as tarefas.

Quanto as diferentes técnicas de representação vetorial, nota-se no emprego que fizemos das técnicas de *text embedding* a oposição de dois paradigmas, o paradigma empregado no USE que é um paradigma "generalista", em que o algoritmo de aprendizagem, não somente utiliza a ideia de aprendizado por transferência (*Transfer Learning*), como também é baseado em aprendizado multitarefa, utilizando um conjunto de tarefas como tarefa fonte ao invés de uma única tarefa fonte, característica que visa aumentar a generalização da representação vetorial criada, conjugado ao fato de utilizarmos uma versão pré-treinada do modelo, treinada usando vários conjuntos de dados, os quais pertencem a diversos domínios. Esse paradigma se opõe ao paradigma "especialista" usado no *Doc2vec*, em que não se faz uso do aprendizado multitarefa (*Multitask learning*) e o algoritmo é treinado apenas com os conjuntos de texto presentes no conjunto de dados obtido na competição, tando de redação, quanto de respostas discursivas. Observamos resultados semelhantes entre as instâncias do pipeline construído que utilizam *Doc2vec* e as instâncias que utilizam USE, com uma leve superioridade para o USE, mas não é

possível afirmar se a superioridade é devida a vantagem de uma técnica sobre outra, ou devido ao paradigma empregado, ou seja, a decisão de treinar a representação com o conjunto de textos que trabalhamos nesse trabalho. Não verificamos, por exemplo, se treinar o *Doc2vec* com dados de um domínio totalmente diverso causaria queda na desempenho preditivo, ou ainda se treinar o *Doc2vec* para a representação vetorial de cada conjunto de textos apenas com aquele conjunto de textos, causaria aumento no desempenho preditivo. Finalmente observamos que usar técnicas de representação vetorial mais sofisticadas não trouxe ganho de desempenho quando comparado ao uso do TF-IDF.

No estado em que se encontra, nosso trabalho apresenta algumas limitações. Por exemplo, devido aos recursos computacionais disponíveis para realização dos experimentos computacionais, não pudemos fazer uma aplicação do *Grid Search* de forma mais extensiva. Em particular, na abordagem de classificação ordinal, tivemos que abrir mão da otimização de hiperparâmetros. Além disso, devido à incompatibilidade de bibliotecas, não foi possível reproduzir o trabalho desenvolvido pelo Luis Tandalla (ganhador da competição em 2012), o que limitou nossa discussão acerca de resultados dos trabalhos relacionados. Finalmente, uma limitação relevante é a impossibilidade de treinar nossas próprias versões do USE, que empobreceu a discussão na comparação de desempenho preditivo frente ao *Doc2vec*.

## 5.2 Trabalhos Futuros

Considerando potenciais para a continuidade desse trabalho, elencamos alguns trabalhos futuros.

- **Aplicação de aprendizado por comitês**. No presente trabalho, a técnica de representação vetorial TF-IDF obteve, de forma geral, desempenho preditivo superior às outras técnicas. Uma proposta de trabalho futuro é comparar o TF-IDF com um comitê formado com as outras técnicas de representação vetorial.

- **Uso das respostas sugeridas para gerar *features***. O conjunto de textos de respostas discursivas contém também um conjunto de respostas sugeridas. Uma proposta de trabalho futuro é criar *features* baseadas na similaridade entre a resposta dada pelo aluno e a resposta sugerida. Para fazer isso, usaremos as quatro técnicas de representação vetorial já empregadas nesse trabalho, e dadas as representações vetoriais, usaremos técnicas como similaridade de cossenos para medir a distância entre as representações vetoriais.

- **Uso de novas *features* como bigramas e trigramas**. A técnica de representação vetorial TF-IDF obteve melhor desempenho preditivo que as outras técnicas, mesmo usando apenas unigramas (palavras individuais), queremos avaliar o ganho de desempenho preditivo trazido ao utilizar bigramas e trigramas.

- **Avaliação do pipeline em novos conjuntos de texto**. Outra via de trabalho futuro é avaliar o *pipeline* construído em novos conjuntos de texto, como por exemplo, o conjunto de textos apresentado em Dzikovska et al. [2013].

- **Avaliação do impacto da escolha de palavras na nota**. A técnica de representação vetorial que obteve melhor desempenho preditivo foi o TF-IDF, mesmo utilizando apenas unigramas. Isso levanta a hipótese de que escolher determinadas palavras está associado a uma melhor nota. Para investigar essa hipótese, pretendemos realizar um experimento usando a técnica de representação vetorial TF-IDF e usar técnicas de explicabilidade como o *SHAP* [Lundberg and Lee, 2017].

- **Treinar *Doc2vec* em outros domínios**. A comparação entre as abordagem *Doc2Vec* e USE, foi feita considerando condições de uso diferentes, o USE foi usado a partir de uma versão pré-treinada que foi exposta a conjuntos de textos de domínios bastantes diferentes dos conjuntos utilizados nesse trabalho. Como não é possível treinar uma versão do USE a partir dos conjuntos utilizados nesse trabalho, para equalizar a comparação, treinaremos o *Doc2vec* com textos de dominins diversos do domínio de redações ou de respostas discursivas. Para isso separamos um conjunto de dados baseado em *reviews* de filmes encontrado no Internet Movie Database (IMDB), o qual é usado em uma tarefa de analise de sentimentos originalmente proposta em Maas et al. [2011].

- **Avaliação do desempenho do *Pipeline* considerando a abordagem *Doc2vec* treinada em cada subconjunto para representar os textos do subconjunto**. Conforme pontuado, decidimos concatenar todos os conjuntos de redações e respostas discursivas por entender que isso criaria uma massa de dados maior para o treino dos modelos de representação vetorial, e traria um desempenho preditivo maior. Por outro lado, tal decisão acaba criando um cenário em que o modelo é treinado com dados de domínios diferentes, tornando o menos "especializado", queremos saber se ao treinar o modelo de representação vetorial para cada subconjunto apenas com os dados do subconjunto teremos uma melhoria, ou piora do desempenho preditivo.

# Referências Bibliográficas

Alex Adamson, Andrew Lamb, and Ralph Ma. Automated essay grading. https://cs229.stanford.edu/proj2014/Alex%20Adamson,%20Andrew%20Lamb,%20Ralph%20Ma,%20Automated%20Essay%20Grading.pdf., 2014. xiii, 22, 25, 27, 33, 43

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051. 23

Page EB Ajay HB, Tillett PI. Analysis of essays by computer (aec-ii). *DC: U.S. Department of Health, Education, and Welfare, Ofce of Education, National Center for Educational Research and Development*, 08 1973. 4

Lyle F Bachman, Nathan Carr, Greg Kamei, Mikyung Kim, Michael J Pan, Chris Salvador, and Yasuyo Sawaki. A reliable approach to automatic assessment of short answer free responses. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 2*, COLING '02, page 1–4, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1071884.1071907. URL https://doi.org/10.3115/1071884.1071907. 24, 25

Gavin Brown. The validity of examination essays in higher education: Issues and responses. *Higher Education Quarterly*, 64:276 – 291, 07 2010. doi: 10.1111/j.1468-2273.2010.00460.x. 1

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018. URL https://arxiv.org/abs/1803.11175. 19, 33

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second*

*Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/S13-2045. 66

Eibe Frank and Mark Hall. A simple approach to ordinal classification. volume 2167, pages 145–156, 08 2001. ISBN 978-3-540-42536-6. doi: 10.1007/3-540-44795-4_13. 10, 11, 38

Dennis Grinberg, John Lafferty, and Daniel Sleator. A robust parsing algorithm for link grammars. *CoRR*, abs/cmp-lg/9508003, 08 1995. 24

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652, 2017. 19, 20

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995. doi: 10.1109/ICDAR.1995.598994. 36

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1162. URL https://aclanthology.org/P15-1162. 19

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf. 19

Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014. URL https://arxiv.org/abs/1405.4053. 18, 33

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874. 66

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015. 66

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781. 13, 14, 15, 16

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1076. 23

Peter Oram. Wordnet: An electronic lexical database. christiane fellbaum (ed.). cambridge, ma: Mit press, 1998. pp. 423. *Applied Psycholinguistics*, 22(1):131–134, 2001. doi: 10.1017/S0142716401221079. 24

Shahzad Qaiser and Ramsha Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181, 07 2018. doi: 10.5120/ijca2018917395. 21, 22

Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-0612. URL https://aclanthology.org/W15-0612. xv, 8, 24, 25, 26

Dadi Ramesh and Suresh Kumar Sanampudi. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527, September 2021. doi: 10.1007/s10462-021-10068-2. URL https://doi.org/10.1007/s10462-021-10068-2. 2, 3, 4, 7

Abdulaziz Shehab, Mohamed Elhoseny, and Aboul Ella Hassanien. A hybrid scheme for

automated essay grading based on lvq and nlp techniques. pages 65–70, 12 2016. doi: 10.1109/ICENCO.2016.7856447. 23, 25

Mark D Shermis and Jill C Burstein, editors. *Automated Essay Scoring: a Cross-Disciplinary Perspective*. Routledge Member of the Taylor and Francis Group, New York, NY, January 2003. 1, 2, 7

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1123. URL https://aclanthology.org/N16-1123. 22, 25, 26

Luis Tandalla. Scoring short answer essays. asap short answer scoring competition–luis tandalla's approach. https://kaggle2.blob.core.windows.net/competitions/kaggle/2959/media/TechnicalMethodsPaper.pdf., 2012. xv, 26

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762. 19

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2019. URL https://arxiv.org/abs/1911.02685. 12, 13

# Apêndice A

# Descrição das características dos conjuntos de texto de respostas discursivas

**Tabela A.1:** Conjuntos de dados sobre respostas discursivas (a coluna tipo indica se estamos tratando de uma questão dependente ou não de um texto fonte).

| Conjunto | Matéria | Tipo | Tamanho | Média de palavras | Intervalo | Assunto |
|---|---|---|---|---|---|---|
| 1 | ciências | dependente | 1672 | 50 | 0-3 | chuva ácida |
| 2 | ciências | dependente | 1278 | 50 | 0-3 | elasticidade de polímeros |
| 3 | língua inglesa/artes | dependente | 1891 | 50 | 0-2 | espécies invasoras |
| 4 | língua inglesa/artes | dependente | 1738 | 50 | 0-2 | espécies invasoras |
| 5 | ciências | independente | 1795 | 60 | 0-3 | síntese proteica |
| 6 | ciências | independente | 1797 | 50 | 0-3 | difusão celular e transporte |
| 7 | língua inglesa | dependente | 1799 | 50 | 0-2 | característica de rose (personagem do texto-fonte) |
| 8 | língua inglesa | dependente | 1799 | 50 | 0-2 | contexto sobre Mr. Leonard (personagem do texto... |
| 9 | língua inglesa | dependente | 1798 | 40 | 0-2 | lixo espacial |
| 10 | ciências | dependente | 1640 | 60 | 0-2 | experimentos sobre cor e absorção de temperatura |

# Apêndice B

# Exemplo descritivo - Redação

Neste apêndice, para descrever o contexto do conjunto de dados que temos, apresentamos o enunciado das redações do conjunto número 1. Apresentamos também uma redação apresentada como resposta a este conjunto e ainda, os critérios de avaliação para redações feitas em resposta à esse enunciado.

## B.1 Enunciado

"More and more people use computers, but not everyone agrees that this benefits society. Those who support advances in technology believe that computers have a positive effect on people. They teach hand-eye coordination, give people the ability to learn about faraway places and people, and even allow people to talk online with other people. Others have different ideas. Some experts are concerned that people are spending too much time on their computers and less time exercising, enjoying nature, and interacting with family and friends.

Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you."

## B.2 Redação avaliada com 12 (nota máxima)

'Dear @ORGANIZATION1, The computer blinked to life and an image of a blonde haired girl filled the screen. It was easy to find out how life was in @LOCATION2, thanks to the actual @CAPS1 girl explaining it. Going to the library wouldnt́ have filled one with this priceless information and human interection. Computers are a nessessity of life if soceity wishes to grow and expand. They should be supported because they teach hand eye coordination, give people the ability to learn about faraway places, and allow people to talk to others online. Firstly, computers help teach hand eye coordination. Hand-eye coordination is a useful ability that is usod to excel in sports. In a recent survey, @PERCENT1 of kids felt their hand eye coordination improves after computer use. Even a simple thing like tying can build up this skill. Famous

neurologist @CAPS2 @PERSON1 stated in an article last week that, "@CAPS3 and computer strength the @CAPS2. When on the computer, you automatically process what the eyes see into a command for your hands."@CAPS4 hand eye coordination can improve people in sports such as baseball and basketball. If someone wanť to become better in these sports, all theyd́ need to do was turn on the computer. Once people become better at sports, theyŕe more likely to play them and become more healthy. In reality, computers can help with exercising instead of decreasing it. Additionaly, computers allow people to access information about faraway places and people. If someone wanted to reasearch @LOCATION1, all theyd́ need to do was type in a search would be presented to them in it would link forever to search through countless things. Also, having the ability to learn about cultures can make peole peole and their cultures, they understand others something. Increase tolerance people are. Computers are a resourceful tool that they can help people in every different aspect of life. Lastly, computer and in technology can allow people to chat. Computer chat and video chat can help the all different nations. Bring on good terms places other than can help us understand story comes out about something that happend in @LOCATION3, people can just go on their computer and ask an actual @LOCATION3 citizen their take on the matter. Also, video chat and online conversation can cut down on expensive phone bills. No one wants to pay more than they have to in this economy. Another good point is that you can acess family members you scaresly visit. It can help you connect within your own family more. Oviously, computers are a useful aid in todays era. their advancements push the world foreward to a better place. Computers can help people because they help teach handeye coordination, give people the bility to learn about faraway places and people, and allow people to talk online with others. Think of a world with no computers or technologicall advancements. The world would be sectored and unified, contact between people scare, and information even. The internet is like thousands or librarys put together. Nobody would know much about other nations and news would travel slower. Is that the kind of palce you want people to live in?'

## B.3 Redação avaliada com 8 (nota intermediária)

'Computers, a @LOCATION1 topic if you ask me. Sure they arnt very good for you dosnt́ mean if you use it correctly then it can be a souce of life. Computers are helpful in many ways, they can be an information post. The social networks can help your kids get more interactive with the world. During school kids have projects and without computers. Read on as I will explain in details. Computers, as you know, are one of the worlds greatest information posts.

@NUM1 out of ten people survayed said that the computers are a value of everyday life, The earthquake in haiti is recovalng now, but half the world wouldnt know that with out computers so have a heart, and dont take away the most valued info in the world. Secondly, what would you say if I told, you that kids are more socialy retarded than ever! Well its true. Over @NUM2 of the worlds people rely on intractive websites to comunicate with each other. Fact, children have more friends over the internet then ever. You dont want to be the @LOCATION1 in all this do yours ar know as the person who kept kids from living? I dont think so. Lastly, all people in school use computers to do projects, for instance @PERSON2 had horrible hand writing and this was the bigest project of the year. So he turned his computer to type it, he gets on at. But what if there was (@CAPS1) computer! @PERSON2 would have gotten a @CAPS2 - for the project. In fact teachers @CAPS3 wear tell kids to type hw/projects. @PERSON1 says "@CAPS3 year kids pass my class by typing and if they then they would all flunk!"@CAPS4 got the point but more importantly do you?! To sumit all up for those of you lazy enough to not read there story essay. Computers are used for everyday needs such as finding out information. Makeing friends, or doing projects. So are you gonna be the person who sits back and lets that happen or are you going to get up and stop the maddness once and for all. Its you choice but think of the socity retarded, dumb kids that fail the grade @CAPS3 year.'

## B.4 Redação avaliada com 4 (nota baixa)

'Computers a good because you can get infermation, you can play games, you can get pictures, But when you on the computer you might find something or someone that is bad or is viris. If ther is a vris you might want shut off the computers so it does not get worse. The are websites for kids, like games, there are teen games, there are adult games. Also pictures are bad for kids because most of the time they lead to inapropreit pictures. You should only look up infermation that you need not things like wepons or knifes. Also there are differnt kinds of companies like @CAPS1 @CAPS2. @CAPS2 is a good place to get computers @CAPS1 so is @CAPS1'

## B.5 Critérios de avaliação

Score Point 1: An undeveloped response that may take a position but offers no more than very minimal support. Typical elements:

- Contains few or vague details.
- Is awkward and fragmented.
- May be difficult to read and understand.
- May show no awareness of audience.

Score Point 2: An under-developed response that may or may not take a position. Typical elements:

- Contains only general reasons with unelaborated and/or list-like details.
- Shows little or no evidence of organization.
- May be awkward and confused or simplistic.
- May show little awareness of audience.

Score Point 3: A minimally-developed response that may take a position, but with inadequate support and details. Typical elements:

- Has reasons with minimal elaboration and more general than specific details.
- Shows some organization.
- May be awkward in parts with few transitions.
- Shows some awareness of audience.

Score Point 4: A somewhat-developed response that takes a position and provides adequate support. Typical elements:

- Has adequately elaborated reasons with a mix of general and specific details.
- Shows satisfactory organization.
- May be somewhat fluent with some transitional language.
- Shows adequate awareness of audience.

Score Point 5: A developed response that takes a clear position and provides reasonably persuasive support. Typical elements:

- Has moderately well elaborated reasons with mostly specific details.
- Exhibits generally strong organization.
- May be moderately fluent with transitional language throughout.
- May show a consistent awareness of audience.

Score Point 6: A well-developed response that takes a clear and thoughtful position and provides persuasive support. Typical elements:

- Has fully elaborated reasons with specific details.
- Exhibits strong organization.
- Is fluent and uses sophisticated transitional language.
- May show a heightened awareness of audience.

mRNA leaves the nucleus when it joins with Trna. TRNA reads Mrna, it creates protein based on the códon that was read. The proteins are continually created until tRNA hits a stop códon. The proteins created are now a protein chain.

# Apêndice C

# Exemplo descritivo - Resposta discursiva

## C.1 Enunciado

Prompt—Protein Synthesis Item

Starting with mRNA leaving the nucleus, list and describe four major steps involved in protein synthesis.

## C.2 Orientações de correção

Rubric for Protein Synthesis

Key Elements:

- mRNA exits nucleus via nuclear pore.
- mRNA travels through the cytoplasm to the ribosome or enters the rough endoplasmic reticulum.
- mRNA bases are read in triplets called codons (by rRNA).
- tRNA carrying the complementary (U=A, C+G) anticodon recognizes the complementary codon of the mRNA.
- The corresponding amino acids on the other end of the tRNA are bonded to adjacent tRNA's amino acids.
- A new corresponding amino acid is added to the tRNA.
- Amino acids are linked together to make a protein beginning with a START codon in the P site (initiation).
- Amino acids continue to be linked until a STOP codon is read on the mRNA in the A site (elongation and termination).

Rubric: 3 points Four key elements 2 points Three key elements 1 point One or two key elements 0 points Other

## C.3 Resposta avaliada com 3 pontos

After mRNA is transcribed in the nucleus, it leaves and goes into the ribosomes. Then the mRNA is paired with its anticódon carried by tRNA, connected to na aminoacid. As the mRNA gets translated the tRNA breaks off and its aminoacid connects with the next tRNA molecule. This continues until there is a long chain of aminoacids that forms a polypeptide, or protein.

## C.4 Resposta avaliada com 3 pontos

The mRNA moves from the nucleus to ribosomes. Then Trna will match anticódons with mRNA códons, the tRNA molecules have aminoacids attached to them, so the aminoacids are bonded in the same order as the códons instructed. When the Mrna is done translating, the protein is formed.

## C.5 Resposta avaliada com 2 pontos

mRNA leaves the nucleus when it joins with Trna. TRNA reads Mrna, it creates protein based on the códon that was read. The proteins are continually created until tRNA hits a stop códon. The proteins created are now a protein chain.

## C.6 Resposta avaliada com 1 pontos

1) mRNA leaves the nucleus 2) mRNA is transcribed 3) tRNA brings códons to match with anticódons 4) aminoacids are assembled, making proteins

## C.7 Resposta avaliada com 0 pontos

1) the mRNA connects with DNA 2) Protein polymerse codes their basepairs 3) Base pairs combine to DNA and mRNA 4) There are 4 copies of DNA

## C.8 Resposta avaliada com 0 pontos

1 mRNA leaves the nucleus 2 mRNA creates aminoacids 3 aminoacids transcribe tRNA 4 tRNA is used to create proteins

# Apêndice D

# Análise exploratória - Redações

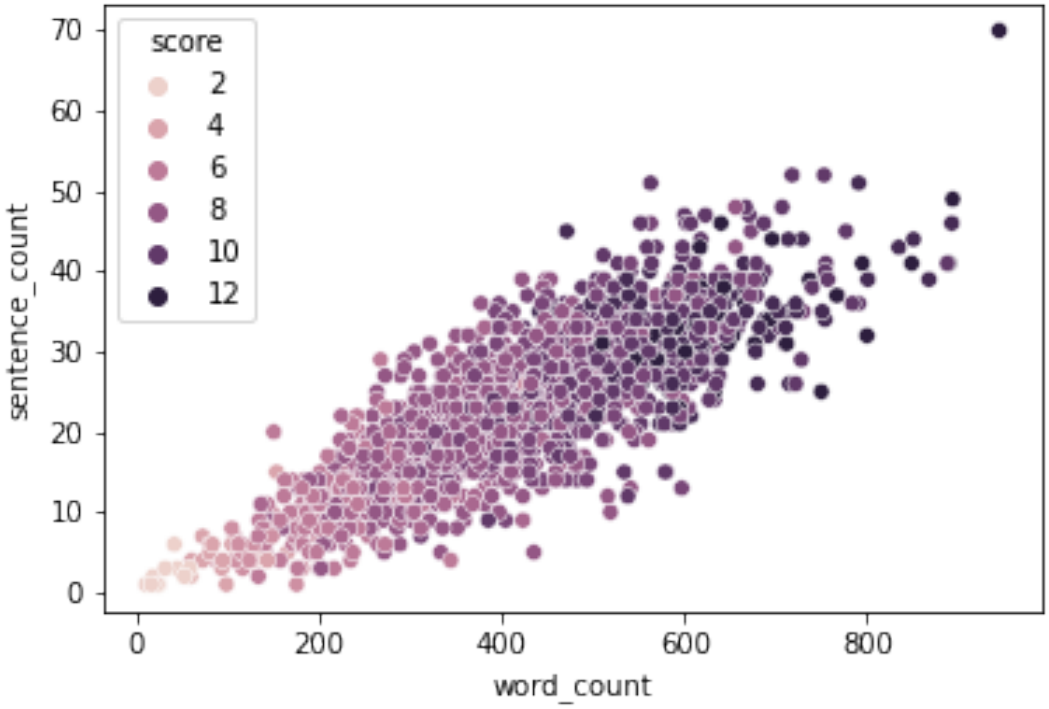



**Figura D.1:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o primeiro conjunto de redações.

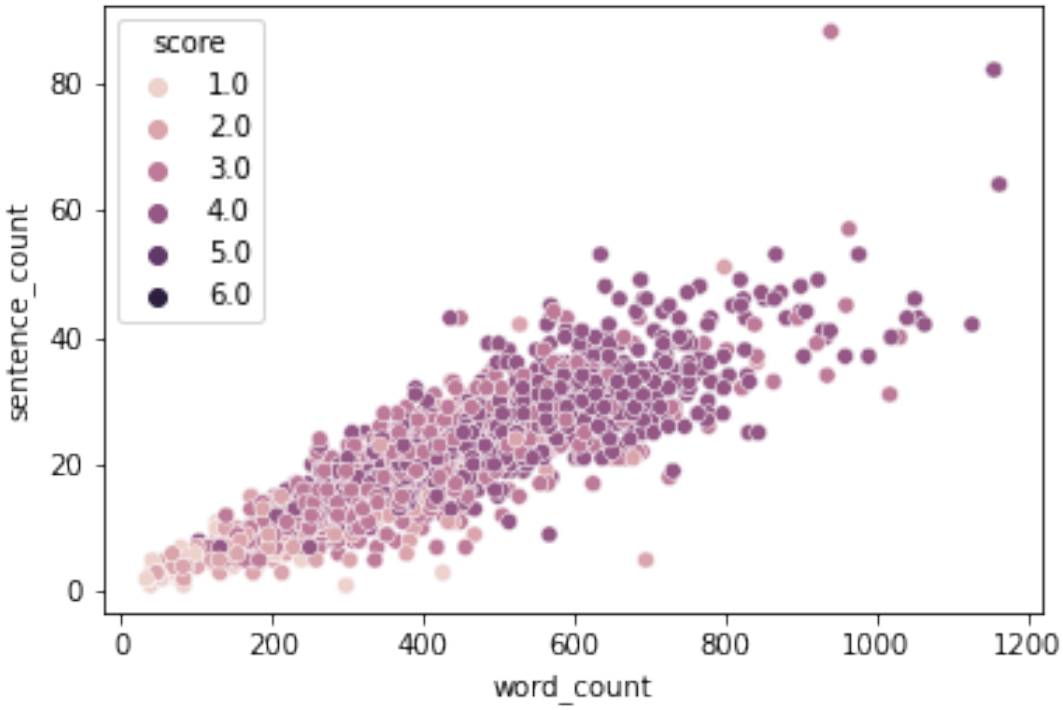



**Figura D.2:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o segundo conjunto de redações.

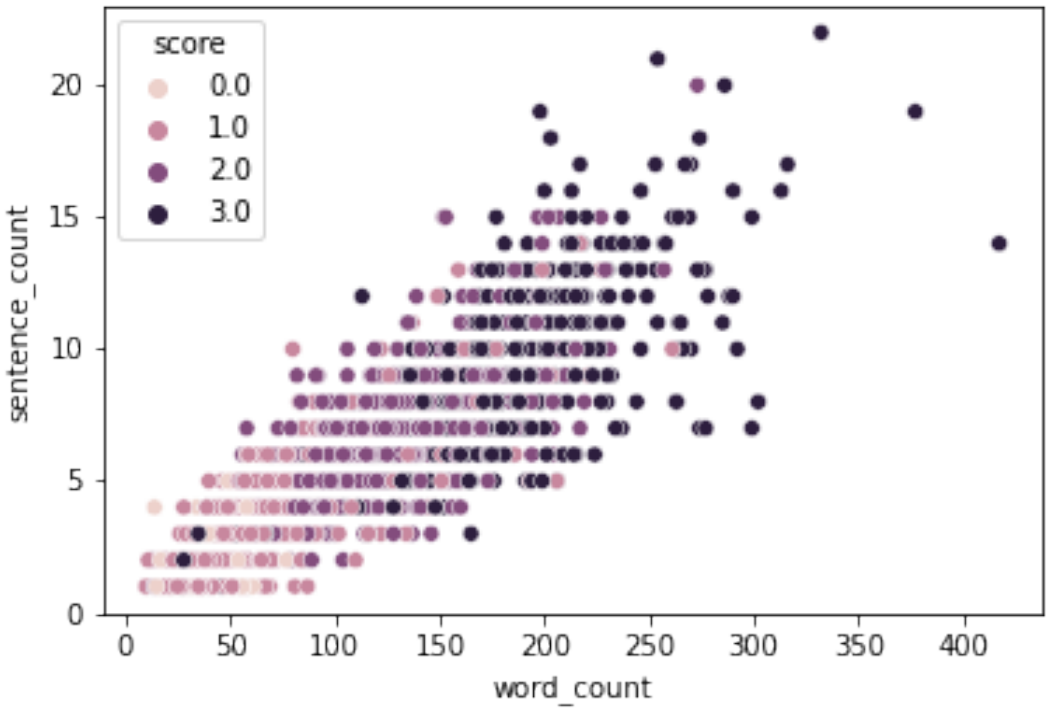



**Figura D.3:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o terceiro conjunto de redações.

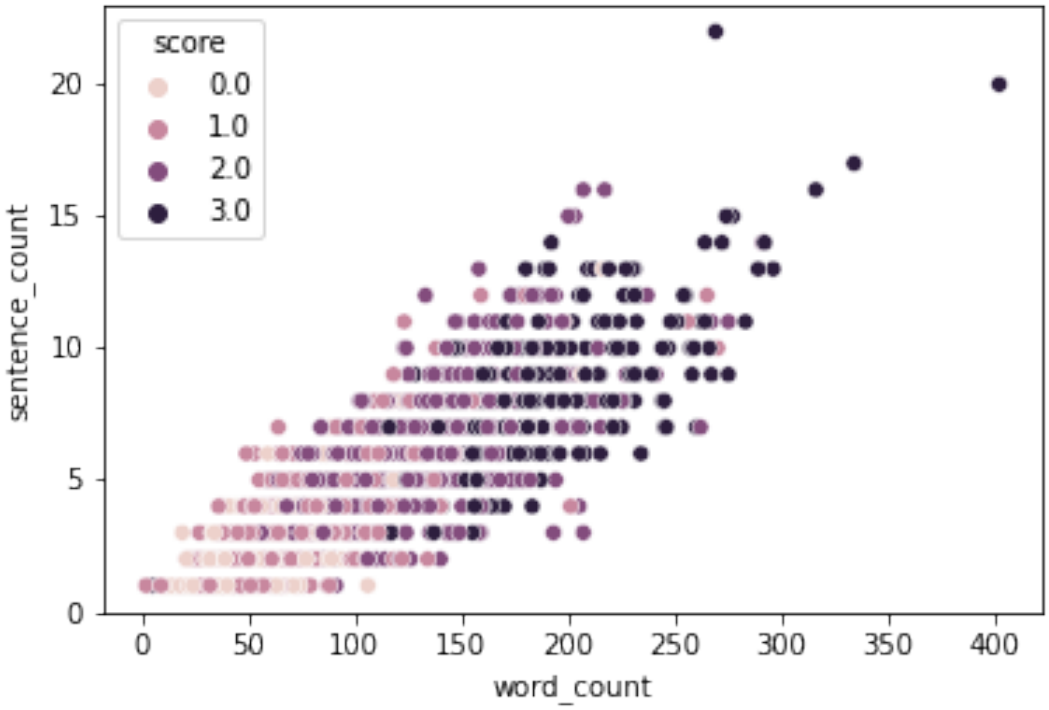



**Figura D.4:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o quarto conjunto de redações.

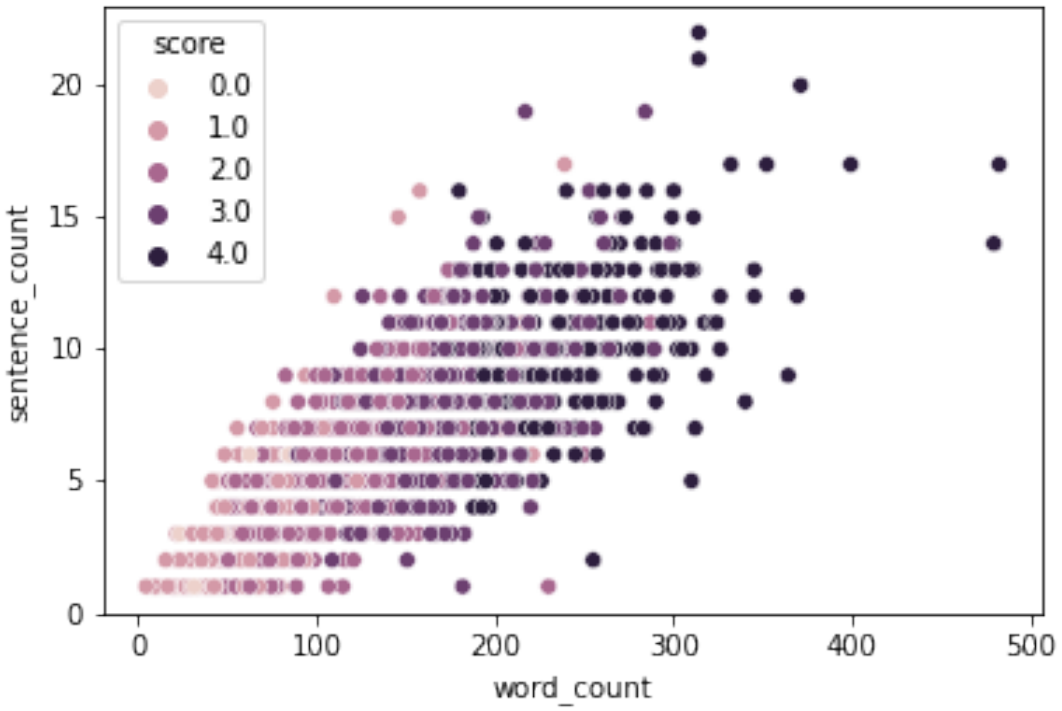



**Figura D.5:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o quinto conjunto de redações.

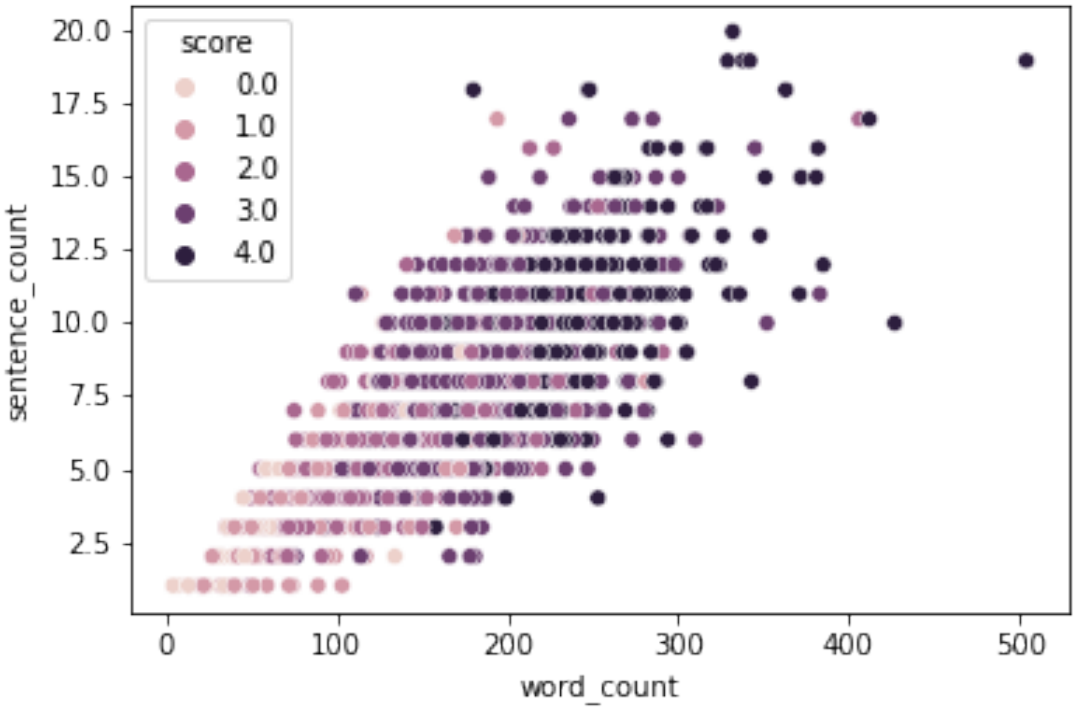



**Figura D.6:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o sexto conjunto de redações.

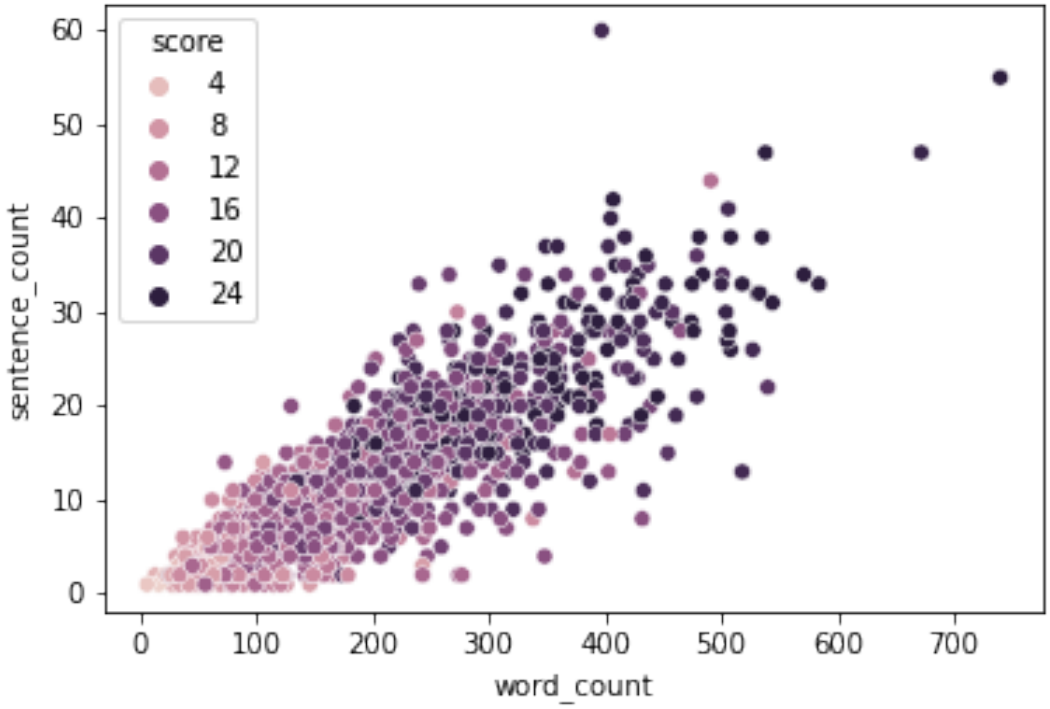



**Figura D.7:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o sétimo conjunto de redações.

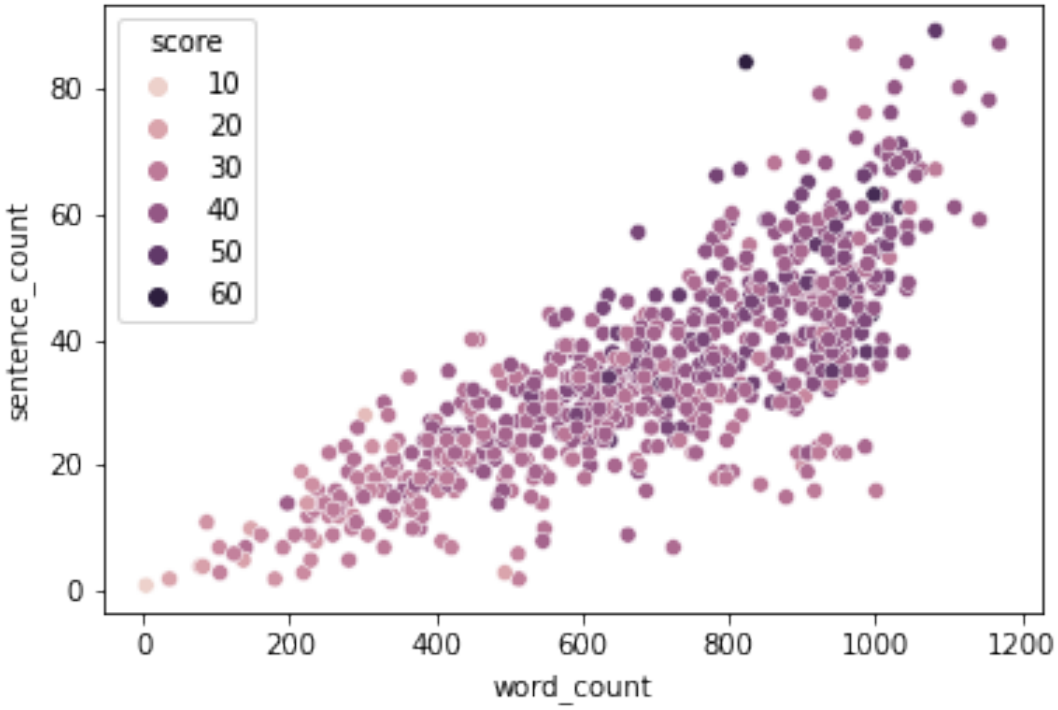



**Figura D.8:** Dispersão da relação entre contagem de palavras, contagem de sentenças e nota atribuída para o oitavo conjunto de redações.

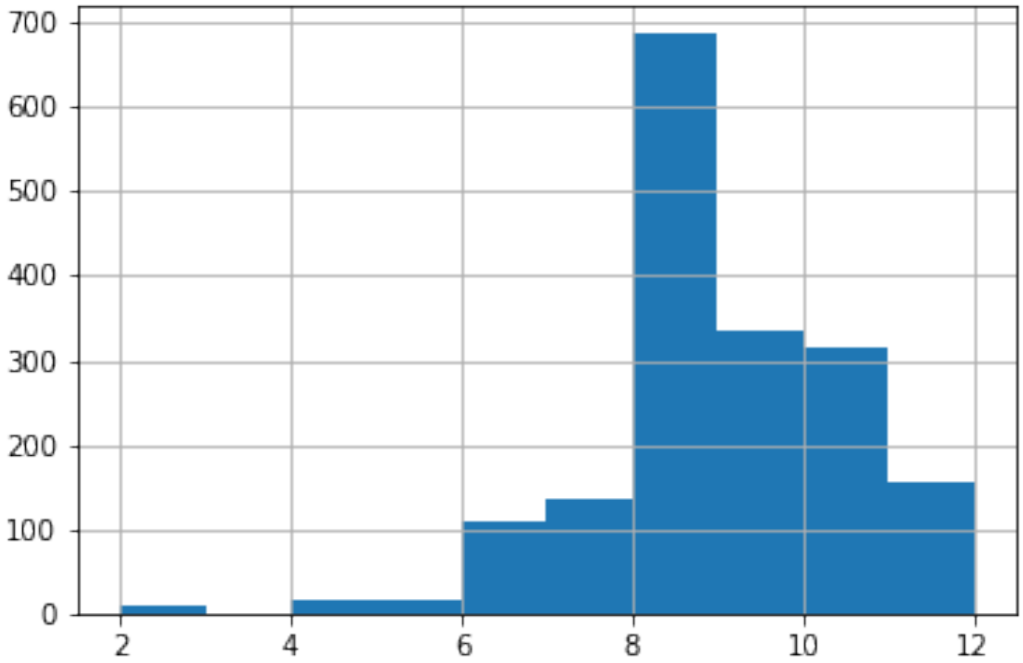



**Figura D.9:** A imagem mostra um histograma que representa a distribuição das redações por nota no primeiro conjunto, no eixo X, estão representadas as notas, enquanto no eixo Y está representada a quantidade de redações que obtiveram aquela nota

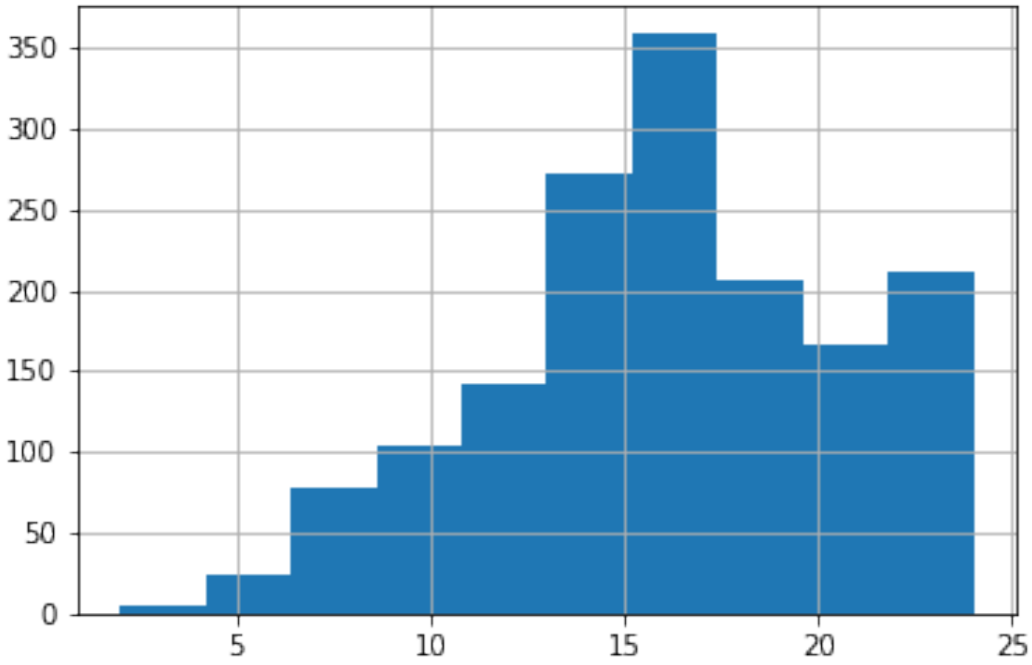



**Figura D.10:** A imagem mostra um histograma que representa a distribuição das redações por nota no sétimo conjunto, no eixo X, estão representadas as notas, enquanto no eixo Y está representada a quantidade de redações que obtiveram aquela nota

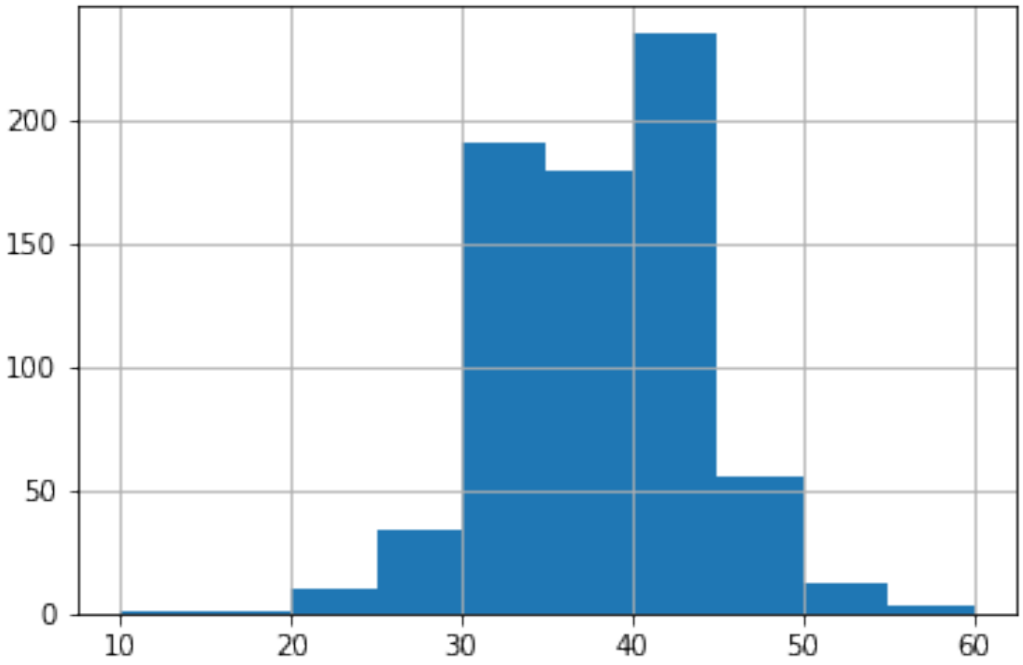



**Figura D.11:** A imagem mostra um histograma que representa a distribuição das redações por nota no oitavo conjunto, no eixo X, estão representadas as notas, enquanto no eixo Y está representada a quantidade de redações que obtiveram aquela nota

# Apêndice E

# Análise exploratória - Respostas discursivas

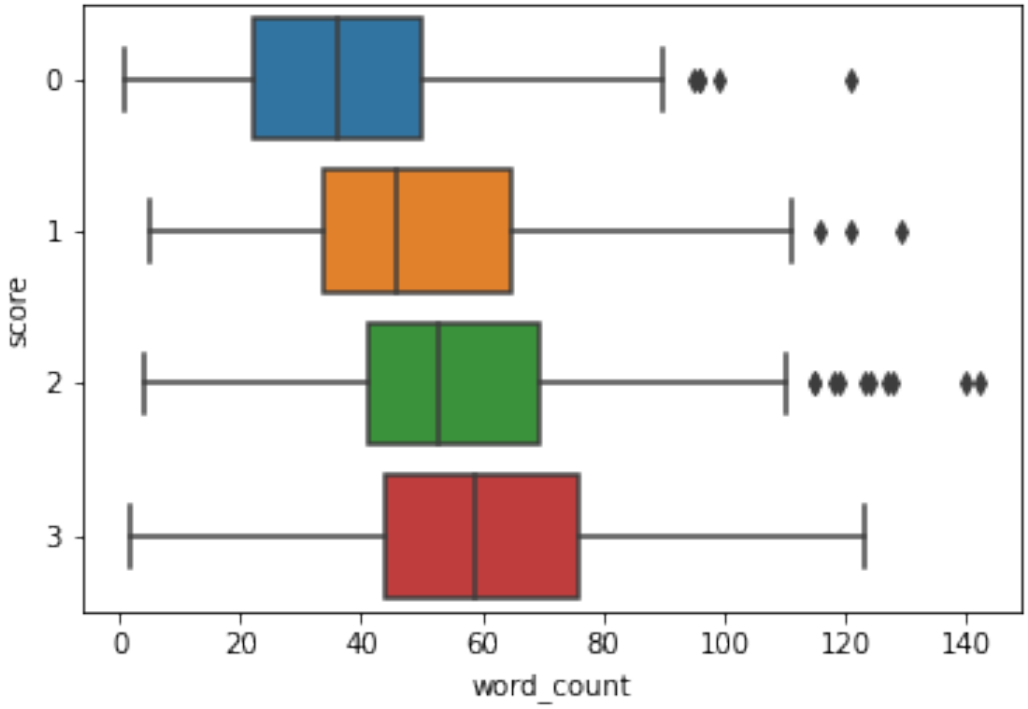



**Figura E.1:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o primeiro conjunto de respostas discursivas.

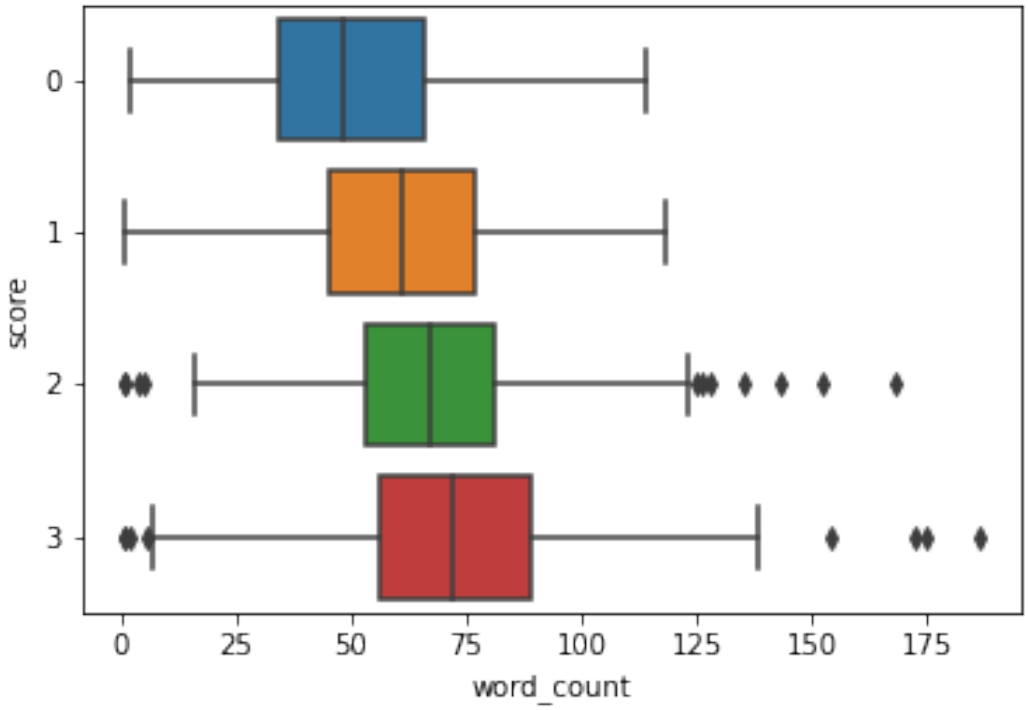



**Figura E.2:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o segundo conjunto de respostas discursivas.

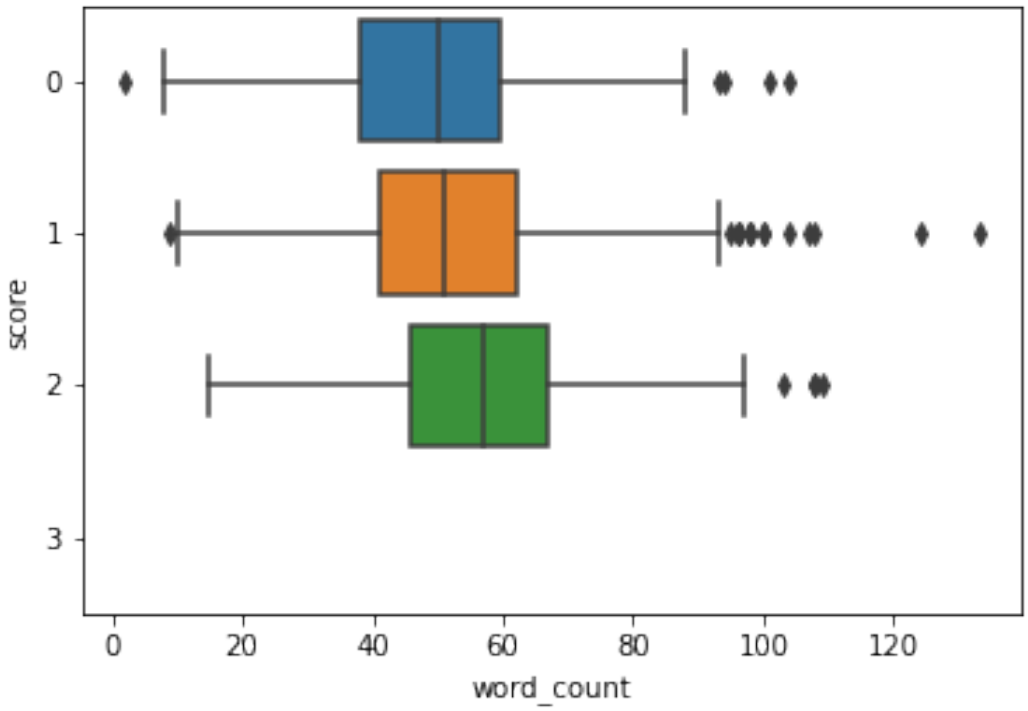



**Figura E.3:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o terceiro conjunto de respostas discursivas.

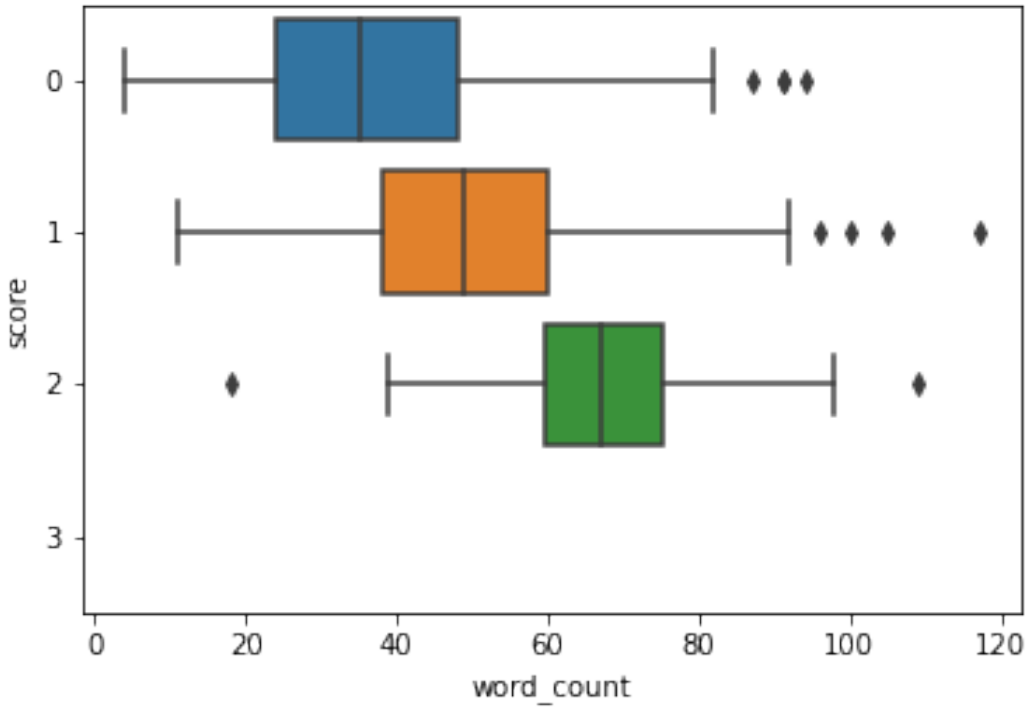



**Figura E.4:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o quarto conjunto de respostas discursivas.

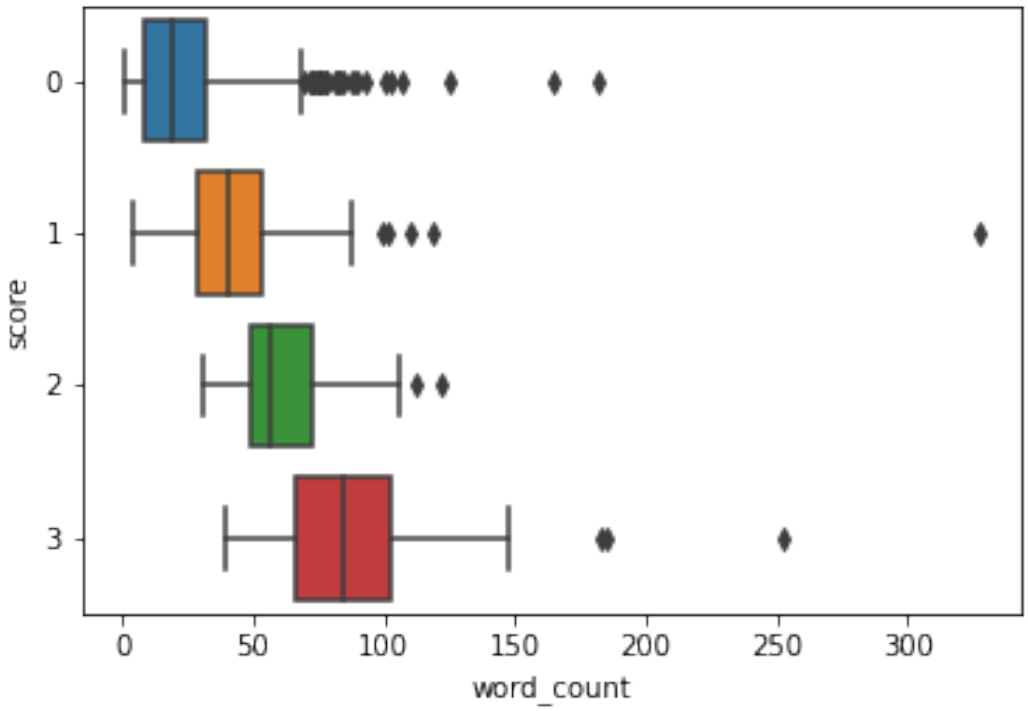



**Figura E.5:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o quinto conjunto de respostas discursivas.

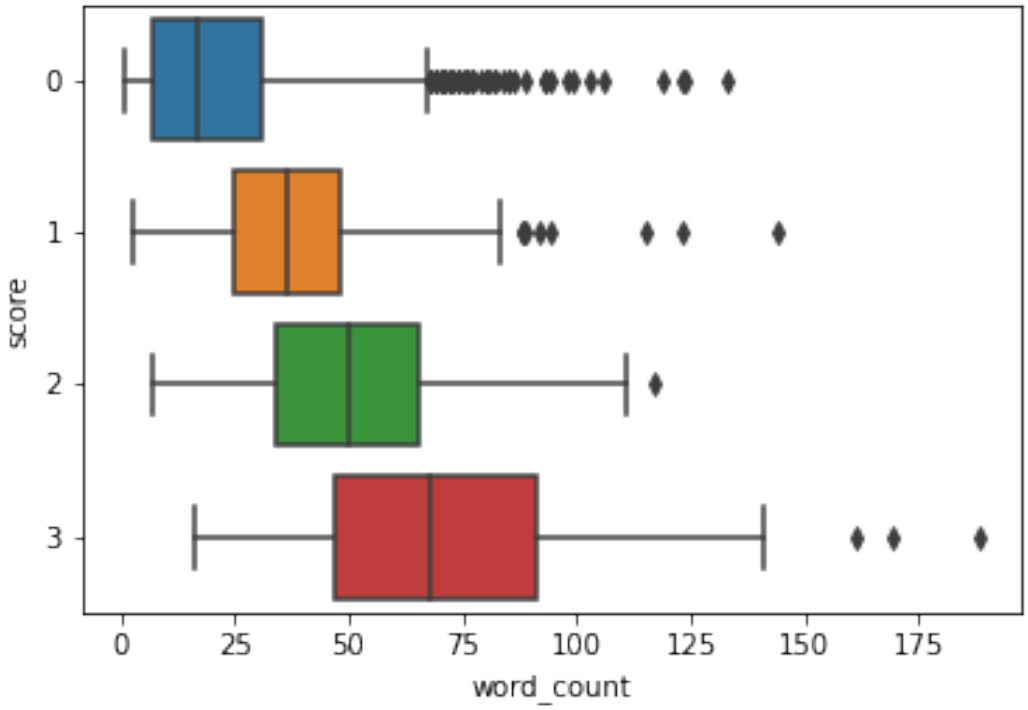



**Figura E.6:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o sexto conjunto de respostas discursivas.

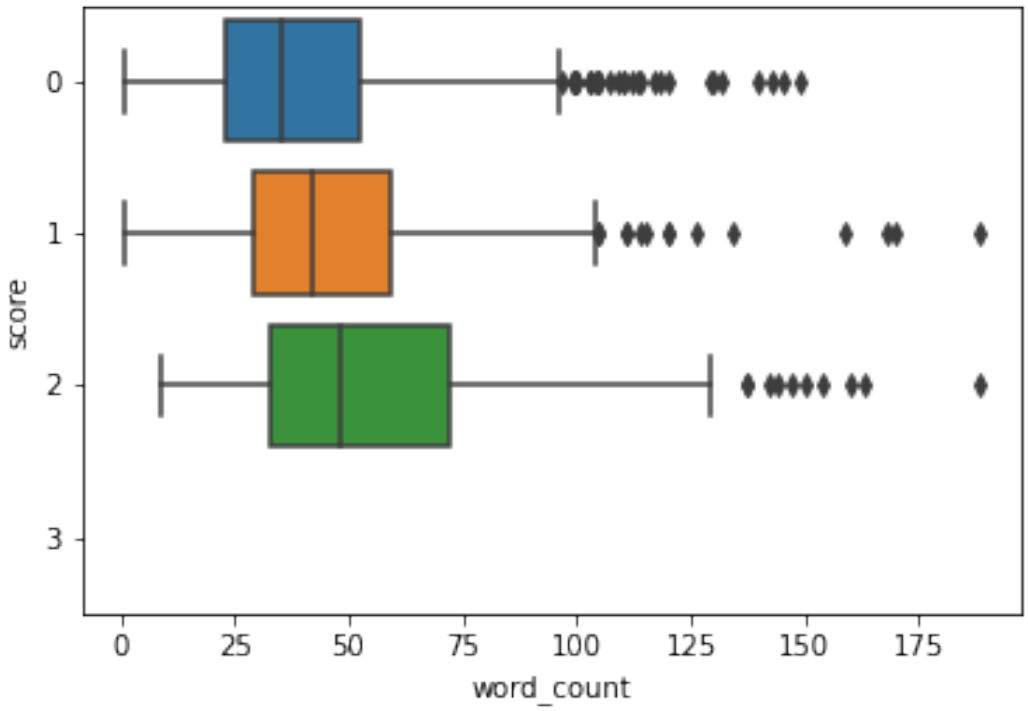



**Figura E.7:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o sétimo conjunto de respostas discursivas.

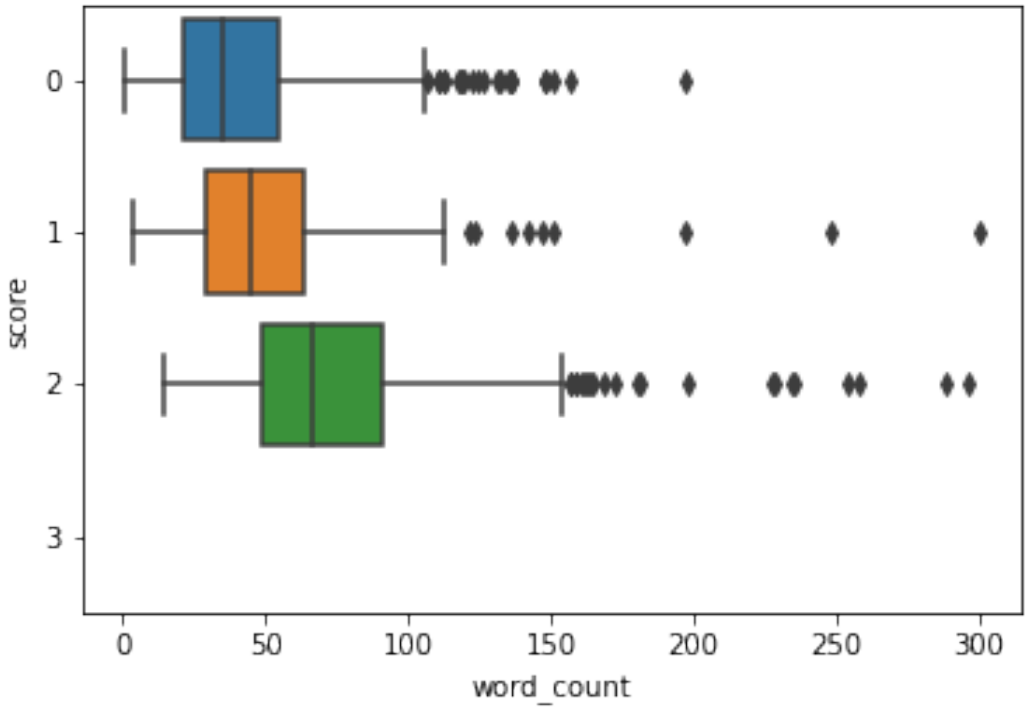



**Figura E.8:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o oitavo conjunto de respostas discursivas.

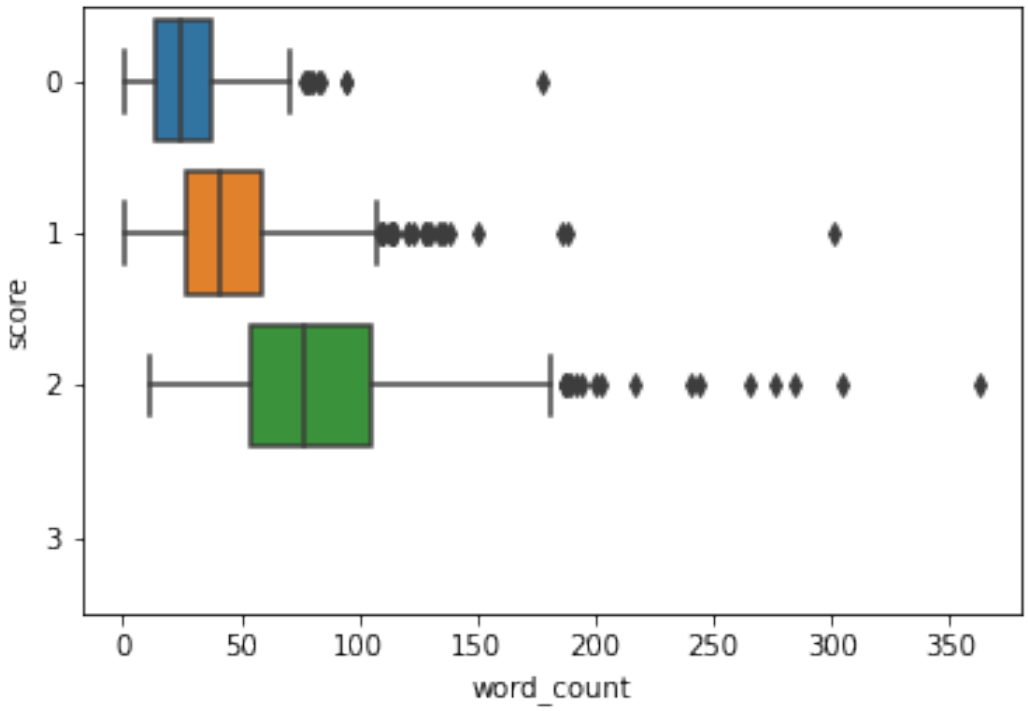



**Figura E.9:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o nono conjunto de respostas discursivas.

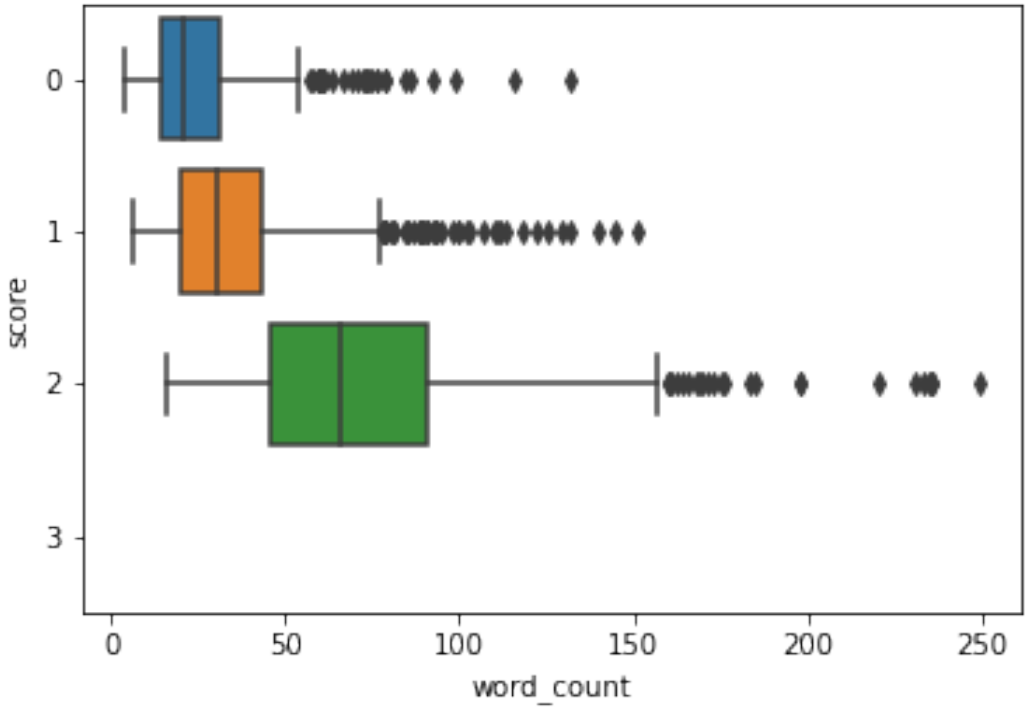



**Figura E.10:** Diagrama de caixa mostrando a distribuição de palavras por cada conceito para o décimo conjunto de respostas discursivas.

# Apêndice F

# Regras de formação das notas- redações

A Tabela F.1 descreve as regras de formação para as notas das redações nos conjuntos de 1 a 7. As regras para a formação da nota no oitavo conjunto são mais complexas e serão explicadas na Seção F.2.

## F.1 Regras de Formação das notas finais das redaçoes

**Tabela F.1:** Descritivo das regras de formação das notas.

| conjunto | regra |
|---|---|
| 1 | Se as notas dois avaliadores forem iguais ou adjacentes, a nota final será a soma, caso contrário a nota final será atribuída por um terceiro avaliador |
| 2 | Para cada domínio a nota será somente àquela dada pelo primeiro avaliador, a nota dada pelo segundo avaliador não é considerada no calculo da nota final |
| 3 | Se as notas dos dois avaliadores forem iguais ou adjacentes, a nota final será a média, caso contrário a nota será atribuída por um terceiro avaliador |
| 4 | Se as notas dos dois avaliadores forem iguais ou adjacentes, a nota final será a média, caso contrário a nota será atribuída por um terceiro avaliador. |
| 5 | Maior entre as duas notas |
| 6 | Maior entre as duas notas |
| 7 | Soma das notas dos dois avaliadores |
| 8 | Verificar tópico específico |

## F.2 Descrição da formação da nota no oitavo conjunto de redações

- Ideas and Content - I

- Organization - O
- Sentence Fluency - S
- Conventions - C
- Voice - V
- Word choice - W

Each student essay is rated for six Writing traits (I, O, V, W, S, C), by two independent raters: Rater 1 and Rater 2. Rater 3 provides a third (resolution) rating for each trait, triggered by the following rules:

- Standard Rule: Non-adjacency between the 1st and 2nd scorer on any of the 6 traits generates a resolution read.
- Cusp Rule: If first or second score has all 4s on:
  - Ideas and Content
  - Sentence Fluency
  - Conventions,

  and the other (1st or 2nd score) has one 3 and three 4s in these categories, require a resolution. Voice and Word Choice are excluded – it does not matter what scores occur for Voice or Word choice (though non-adjacent Voice and Word Choice scores will still cause failure on (1)).

Total Composite Score: For most essays:

$score = (I_R1 + I_R2) + (O_R1 + O_R2) + (S_R1 + S_R2) + 2(C_R1 + C_R2)$

When there is Rater 3 set of scores for the essay then the Total Composite Score formula changes to:

$score = 2(I_R3) + 2(O_R3) + 2(S_R3) + 4(C_R3)$ or equivalently $score = 2(I + O + S + C) + 2(C)$

Note the use of only four of the six traits.

# Apêndice G

# Visão completa dos resultados em ambos os tipos textuais avaliados

**Tabela G.1:** Visão do desempenho preditivo obtido na tarefa de avaliação de respostas discursivas para instâncias que usam a abordagem de classificação ordinal, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas, D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.02 | 0.04 | -0.01 | 0.02 | 0.01 | 0.00 | 0.05 | 0.31 | 0.50 | 0.54 |
| 64 | D2V | 0.02 | 0.03 | -0.01 | 0.02 | 0.01 | 0.00 | 0.12 | 0.32 | 0.52 | 0.51 |
| 128 | D2V | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.00 | 0.07 | 0.35 | 0.51 | 0.50 |
| 256 | D2V | 0.02 | 0.03 | 0.01 | 0.00 | 0.01 | 0.00 | 0.09 | 0.36 | 0.48 | 0.54 |
| 512 | D2V | 0.02 | 0.02 | 0.00 | 0.02 | 0.01 | 0.00 | 0.05 | 0.30 | 0.52 | 0.55 |
| 10 | LSI | 0.04 | 0.02 | 0.00 | 0.02 | 0.00 | 0.01 | 0.01 | 0.27 | 0.37 | 0.37 |
| 20 | LSI | 0.02 | 0.04 | -0.01 | 0.01 | 0.00 | 0.00 | 0.05 | 0.22 | 0.42 | 0.31 |
| 30 | LSI | 0.03 | 0.03 | 0.01 | 0.02 | 0.00 | 0.00 | 0.01 | 0.27 | 0.42 | 0.40 |
| 40 | LSI | 0.01 | -0.00 | 0.00 | 0.02 | 0.00 | 0.00 | 0.04 | 0.26 | 0.42 | 0.39 |
| 50 | LSI | 0.02 | 0.01 | -0.02 | 0.02 | 0.01 | 0.00 | 0.03 | 0.28 | 0.41 | 0.39 |
| 100 | LSI | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | -0.00 | 0.12 | 0.40 | 0.42 |
| 32 | TF IDF | 0.34 | 0.45 | 0.66 | 0.34 | 0.03 | 0.04 | 0.54 | 0.71 | 0.72 | 0.77 |
| 64 | TF IDF | 0.34 | 0.45 | 0.66 | 0.34 | 0.03 | 0.04 | 0.54 | 0.71 | 0.72 | 0.77 |
| 128 | TF IDF | 0.34 | 0.45 | 0.66 | 0.34 | 0.03 | 0.04 | 0.54 | 0.71 | 0.72 | 0.77 |
| 256 | TF IDF | 0.34 | 0.45 | 0.66 | 0.34 | 0.03 | 0.04 | 0.54 | 0.71 | 0.72 | 0.77 |
| 512 | TF IDF | 0.34 | 0.45 | 0.66 | 0.34 | 0.03 | 0.04 | 0.54 | 0.71 | 0.72 | 0.77 |
| 512 | USE | 0.02 | 0.03 | 0.10 | 0.09 | 0.01 | 0.00 | 0.13 | 0.32 | 0.50 | 0.53 |

**Tabela G.2:** Visão do desempenho preditivo obtido na tarefa de avaliação de respostas discursivas para instâncias que usam a abordagem de classificação, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas, D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.57 | 0.24 | 0.00 | 0.44 | 0.37 | 0.21 | 0.16 | 0.41 | 0.61 | 0.59 |
| 64 | D2V | 0.56 | 0.26 | 0.00 | 0.45 | 0.32 | 0.21 | 0.11 | 0.35 | 0.62 | 0.56 |
| 128 | D2V | 0.67 | 0.17 | 0.00 | 0.45 | 0.25 | 0.17 | 0.14 | 0.41 | 0.61 | 0.59 |
| 256 | D2V | 0.62 | 0.18 | 0.00 | 0.44 | 0.31 | 0.16 | 0.13 | 0.41 | 0.61 | 0.61 |
| 512 | D2V | 0.66 | 0.23 | 0.00 | 0.46 | 0.33 | 0.17 | 0.16 | 0.39 | 0.64 | 0.58 |
| 10 | LSI | 0.32 | 0.29 | -0.01 | 0.34 | 0.30 | 0.20 | 0.02 | 0.34 | 0.51 | 0.47 |
| 20 | LSI | 0.32 | 0.21 | 0.01 | 0.30 | 0.36 | 0.15 | 0.08 | 0.32 | 0.55 | 0.44 |
| 30 | LSI | 0.29 | 0.26 | 0.00 | 0.29 | 0.10 | 0.09 | 0.07 | 0.33 | 0.54 | 0.49 |
| 40 | LSI | 0.27 | 0.25 | 0.00 | 0.29 | 0.16 | 0.03 | 0.06 | 0.27 | 0.52 | 0.41 |
| 50 | LSI | 0.33 | 0.15 | 0.00 | 0.29 | 0.26 | 0.00 | 0.07 | 0.38 | 0.52 | 0.40 |
| 100 | LSI | 0.23 | 0.08 | -0.01 | 0.29 | 0.00 | 0.00 | 0.03 | 0.31 | 0.52 | 0.46 |
| 32 | TF IDF | 0.64 | 0.52 | 0.08 | 0.49 | 0.61 | 0.74 | 0.55 | 0.54 | 0.71 | 0.65 |
| 64 | TF IDF | 0.71 | 0.51 | 0.06 | 0.53 | 0.58 | 0.66 | 0.55 | 0.56 | 0.73 | 0.64 |
| 128 | TF IDF | 0.73 | 0.49 | 0.03 | 0.52 | 0.50 | 0.53 | 0.49 | 0.55 | 0.72 | 0.60 |
| 256 | TF IDF | 0.68 | 0.40 | 0.03 | 0.53 | 0.41 | 0.44 | 0.44 | 0.54 | 0.73 | 0.59 |
| 512 | TF IDF | 0.69 | 0.41 | 0.01 | 0.50 | 0.42 | 0.43 | 0.39 | 0.52 | 0.70 | 0.58 |
| 512 | USE | 0.72 | 0.30 | 0.02 | 0.53 | 0.45 | 0.45 | 0.28 | 0.45 | 0.70 | 0.66 |

**Tabela G.3:** Visão do desempenho preditivo obtido na tarefa de avaliação de respostas discursivas para instancias que usam a abordagem de regressão, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.52 | 0.23 | 0.00 | 0.48 | 0.61 | 0.64 | 0.27 | 0.39 | 0.63 | 0.57 |
| 64 | D2V | 0.54 | 0.25 | 0.00 | 0.46 | 0.60 | 0.57 | 0.22 | 0.39 | 0.63 | 0.54 |
| 128 | D2V | 0.56 | 0.24 | 0.00 | 0.55 | 0.61 | 0.58 | 0.28 | 0.41 | 0.64 | 0.53 |
| 256 | D2V | 0.53 | 0.25 | 0.00 | 0.47 | 0.66 | 0.60 | 0.28 | 0.32 | 0.64 | 0.60 |
| 512 | D2V | 0.56 | 0.25 | 0.00 | 0.46 | 0.63 | 0.57 | 0.30 | 0.32 | 0.62 | 0.57 |
| 10 | LSI | 0.22 | 0.21 | 0.00 | 0.38 | 0.31 | 0.29 | 0.12 | 0.30 | 0.51 | 0.44 |
| 20 | LSI | 0.32 | 0.19 | 0.00 | 0.37 | 0.50 | 0.28 | 0.18 | 0.28 | 0.54 | 0.41 |
| 30 | LSI | 0.30 | 0.13 | 0.00 | 0.45 | 0.60 | 0.32 | 0.18 | 0.31 | 0.53 | 0.52 |
| 40 | LSI | 0.32 | 0.16 | 0.00 | 0.32 | 0.52 | 0.40 | 0.10 | 0.31 | 0.46 | 0.39 |
| 50 | LSI | 0.18 | 0.02 | 0.00 | 0.37 | 0.49 | 0.30 | 0.04 | 0.29 | 0.53 | 0.38 |
| 100 | LSI | -0.06 | 0.02 | 0.00 | 0.34 | 0.23 | 0.27 | 0.14 | 0.17 | 0.40 | 0.39 |
| 32 | TF IDF | 0.60 | 0.48 | 0.02 | 0.59 | 0.77 | 0.85 | 0.58 | 0.58 | 0.73 | 0.68 |
| 64 | TF IDF | 0.71 | 0.48 | 0.01 | 0.59 | 0.78 | 0.86 | 0.59 | 0.61 | 0.76 | 0.70 |
| 128 | TF IDF | 0.70 | 0.49 | 0.01 | 0.58 | 0.78 | 0.86 | 0.61 | 0.60 | 0.76 | 0.70 |
| 256 | TF IDF | 0.72 | 0.48 | 0.01 | 0.59 | 0.79 | 0.86 | 0.62 | 0.60 | 0.76 | 0.70 |
| 512 | TF IDF | 0.71 | 0.48 | 0.01 | 0.58 | 0.81 | 0.87 | 0.61 | 0.61 | 0.76 | 0.69 |
| 512 | USE | 0.63 | 0.30 | 0.01 | 0.56 | 0.73 | 0.66 | 0.40 | 0.43 | 0.67 | 0.64 |

**Tabela G.4:** Visão do desempenho preditivo obtido na tarefa de avaliação de redações para instâncias que usam a abordagem de classificação ordinal, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas, D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.00 | 0.04 | 0.42 | 0.19 | 0.25 | 0.20 | 0.01 | 0.0 |
| 64 | D2V | 0.00 | 0.04 | 0.40 | 0.18 | 0.23 | 0.18 | 0.00 | 0.0 |
| 128 | D2V | 0.00 | 0.00 | 0.39 | 0.18 | 0.24 | 0.14 | 0.00 | 0.0 |
| 256 | D2V | 0.00 | 0.00 | 0.36 | 0.13 | 0.21 | 0.13 | 0.00 | 0.0 |
| 512 | D2V | 0.00 | 0.00 | 0.35 | 0.08 | 0.17 | 0.11 | 0.00 | 0.0 |
| 10 | LSI | 0.00 | 0.02 | 0.38 | 0.18 | 0.24 | 0.19 | 0.01 | 0.0 |
| 20 | LSI | 0.00 | 0.02 | 0.38 | 0.19 | 0.23 | 0.16 | 0.01 | 0.0 |
| 30 | LSI | 0.00 | 0.00 | 0.39 | 0.18 | 0.24 | 0.18 | 0.01 | 0.0 |
| 40 | LSI | 0.00 | 0.00 | 0.38 | 0.18 | 0.23 | 0.19 | 0.01 | 0.0 |
| 50 | LSI | 0.00 | 0.00 | 0.40 | 0.20 | 0.24 | 0.18 | 0.01 | 0.0 |
| 100 | LSI | 0.00 | 0.00 | 0.35 | 0.12 | 0.14 | 0.07 | 0.00 | 0.0 |
| 32 | TF-IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.0 |
| 64 | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.0 |
| 128 | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.0 |
| 256 | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.0 |
| 512 | TF IDF | 0.07 | 0.18 | 0.56 | 0.30 | 0.28 | 0.39 | 0.05 | 0.0 |
| 512 | USE | 0.00 | 0.02 | 0.38 | 0.13 | 0.20 | 0.17 | 0.00 | 0.0 |

**Tabela G.5:** Visão do desempenho preditivo obtido na tarefa de avaliação de redações para instâncias que usam a abordagem de classificação, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas, D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.72 | 0.60 | 0.69 | 0.70 | 0.77 | 0.65 | 0.67 | 0.49 |
| 64 | D2V | 0.67 | 0.58 | 0.70 | 0.69 | 0.77 | 0.66 | 0.66 | 0.42 |
| 128 | D2V | 0.54 | 0.50 | 0.70 | 0.67 | 0.76 | 0.59 | 0.59 | 0.22 |
| 256 | D2V | 0.50 | 0.53 | 0.68 | 0.66 | 0.74 | 0.55 | 0.52 | 0.33 |
| 512 | D2V | 0.42 | 0.44 | 0.67 | 0.66 | 0.71 | 0.53 | 0.46 | 0.25 |
| 10 | LSI | 0.78 | 0.58 | 0.69 | 0.70 | 0.76 | 0.62 | 0.65 | 0.40 |
| 20 | LSI | 0.77 | 0.56 | 0.69 | 0.70 | 0.77 | 0.61 | 0.65 | 0.39 |
| 30 | LSI | 0.76 | 0.55 | 0.68 | 0.67 | 0.77 | 0.65 | 0.67 | 0.42 |
| 40 | LSI | 0.75 | 0.52 | 0.67 | 0.68 | 0.76 | 0.61 | 0.61 | 0.31 |
| 50 | LSI | 0.75 | 0.57 | 0.67 | 0.68 | 0.77 | 0.59 | 0.61 | 0.29 |
| 100 | LSI | 0.64 | 0.54 | 0.64 | 0.62 | 0.72 | 0.56 | 0.55 | 0.24 |
| 32 | TF IDF | 0.76 | 0.72 | 0.73 | 0.76 | 0.83 | 0.69 | 0.70 | 0.40 |
| 64 | TF IDF | 0.73 | 0.72 | 0.73 | 0.77 | 0.82 | 0.69 | 0.68 | 0.39 |
| 128 | TF IDF | 0.63 | 0.69 | 0.74 | 0.75 | 0.82 | 0.73 | 0.62 | 0.34 |
| 256 | TF IDF | 0.60 | 0.64 | 0.74 | 0.73 | 0.81 | 0.73 | 0.56 | 0.28 |
| 512 | TF IDF | 0.53 | 0.61 | 0.73 | 0.68 | 0.78 | 0.64 | 0.46 | 0.23 |
| 512 | USE | 0.58 | 0.58 | 0.70 | 0.72 | 0.75 | 0.66 | 0.57 | 0.29 |

**Tabela G.6:** Visão do desempenho preditivo obtido na tarefa de avaliação de redações para instancias que usam a abordagem de regressão, mostramos os resultados através das diferentes técnicas de representação vetorial usadas e em suas diferentes dimensionalidades. Cabe ressaltar que, nas colunas, D significa dimensionalidade e M significa método. Nos valores da coluna M usamos D2V para indicar doc-to-vec.

| D | M | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 |
|---|---|---|---|---|---|---|---|---|---|
| 32 | D2V | 0.84 | 0.64 | 0.69 | 0.72 | 0.80 | 0.74 | 0.75 | 0.63 |
| 64 | D2V | 0.84 | 0.64 | 0.69 | 0.73 | 0.79 | 0.74 | 0.76 | 0.61 |
| 128 | D2V | 0.83 | 0.62 | 0.69 | 0.70 | 0.80 | 0.73 | 0.75 | 0.57 |
| 256 | D2V | 0.84 | 0.61 | 0.69 | 0.71 | 0.79 | 0.73 | 0.74 | 0.56 |
| 512 | D2V | 0.83 | 0.61 | 0.70 | 0.71 | 0.81 | 0.73 | 0.74 | 0.56 |
| 10 | LSI | 0.82 | 0.57 | 0.68 | 0.66 | 0.76 | 0.56 | 0.75 | 0.51 |
| 20 | LSI | 0.82 | 0.60 | 0.69 | 0.70 | 0.78 | 0.60 | 0.75 | 0.44 |
| 30 | LSI | 0.83 | 0.57 | 0.68 | 0.63 | 0.78 | 0.65 | 0.75 | 0.51 |
| 40 | LSI | 0.82 | 0.57 | 0.68 | 0.66 | 0.76 | 0.62 | 0.66 | 0.45 |
| 50 | LSI | 0.82 | 0.60 | 0.68 | 0.67 | 0.75 | 0.66 | 0.65 | 0.52 |
| 100 | LSI | 0.82 | 0.50 | 0.68 | 0.61 | 0.77 | 0.66 | 0.37 | 0.48 |
| 32 | TF IDF | 0.88 | 0.78 | 0.76 | 0.80 | 0.86 | 0.80 | 0.80 | 0.72 |
| 64 | TF IDF | 0.88 | 0.79 | 0.77 | 0.81 | 0.86 | 0.81 | 0.81 | 0.74 |
| 128 | TF IDF | 0.88 | 0.79 | 0.77 | 0.82 | 0.86 | 0.82 | 0.81 | 0.76 |
| 256 | TF IDF | 0.89 | 0.79 | 0.77 | 0.82 | 0.86 | 0.82 | 0.81 | 0.76 |
| 512 | TF IDF | 0.88 | 0.78 | 0.76 | 0.82 | 0.86 | 0.82 | 0.82 | 0.77 |
| 512 | USE | 0.83 | 0.62 | 0.70 | 0.75 | 0.79 | 0.75 | 0.76 | 0.50 |