UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DIEGO EIZI TAKEI NETO

APERFEIÇOANDO O PROCESSO DE CLASSIFICAÇÃO AUTOMÁTICA DE QUESTÕES DE MATEMÁTICA QUANTO ÀS COMPETÊNCIAS DO PENSAMENTO COMPUTACIONAL

CAMPINA GRANDE - PB

2021

DIEGO EIZI TAKEI NETO

APERFEIÇOANDO O PROCESSO DE CLASSIFICAÇÃO AUTOMÁTICA DE QUESTÕES DE MATEMÁTICA QUANTO ÀS COMPETÊNCIAS DO PENSAMENTO COMPUTACIONAL

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Cláudio Elízio Calazans Campelo.

CAMPINA GRANDE - PB

2021

T136a Takei Neto, Diego Eizi.

Aperfeiçoando o processo de classificação automática de questões de matemática quanto às competências do pensamento computacional. / Diego Eizi Takei Neto. – 2021.

12 f.

Orientador: Prof. Dr. Cláudio Elízio Calazans Campelo.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Classificação de textos. 2. Pensamento computacional. 3. Extração de características. 4. Ensino de matemática. 5. Term Frequency-Inverse Document Frequency. 6. Latent Dirichlet Allocation. 7. Matemática e pensamento computacional. 8. Processamento de linguagem natural. I. Campelo, Cláudio Elízio Calazans. II. Título.

CDU:004.912(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

DIEGO EIZI TAKEI NETO

APERFEIÇOANDO O PROCESSO DE CLASSIFICAÇÃO AUTOMÁTICA DE QUESTÕES DE MATEMÁTICA QUANTO ÀS COMPETÊNCIAS DO PENSAMENTO COMPUTACIONAL

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Cláudio Elízio Calazans Campelo
Orientador – UASC/CEEI/UFCG

Professor Dr. Nazareno Ferreira de Andrade
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 25 de maio de 2021.

CAMPINA GRANDE – PB

ABSTRACT

Computational thinking (CT) is a reasoning process that consists of formulating a problem and solving it in steps that a computer is capable of solving. This process is so important that authors consider it as an enhancer of the competences operational aspects of human beings, which can be used in some strands, such as for example, in interdisciplinary development in basic education subjects, such as Mathematics and Physics, and in development from specific disciplines of Computer Science. In the context of the Mathematics discipline, we can relate an issue among nine computational thinking competencies. Identifying issues that explore these competencies can be extremely useful for students and teachers who have an interest in going deeper into this topic, as the stimulus to CT can increase the ability to solve problems. In this context, the design of intelligent models capable of predicting automatically CT skills in math issues would be a great facilitator in the process of stimulating the resolution of problems. In this work, we use a new database of questions to extract features from excerpts from the text assigned to questions by manual assessments from experts. From these questions, we developed classifiers using Term Frequency-Inverse Document Frequency (TF-IDF) and Latent Dirichlet Allocation (LDA) as model features, recalculating the values of these characteristics through the use of the excerpts, with the objective to increase the importance of those that belong to the excerpts highlighted by the evaluators, and increasing the effectiveness of the classification of the questions in relation to the stimulated computational thinking competencies.

Aperfeiçoando o processo de classificação automática de questões de Matemática quanto às competências do Pensamento Computacional

Trabalho de Conclusão de Curso

Diego Eizi Takei Neto (Aluno), Cláudio Campelo (Orientador)
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Paraíba - Brasil

RESUMO

O pensamento computacional (PC) é um processo de raciocínio que consiste em formular um problema e sua solução em passos que um computador é capaz de realizar. Este processo é tão importante que autores o consideram como um potencializador das competências operacionais do ser humano, que pode ser usado em algumas vertentes, como por exemplo, no desenvolvimento interdisciplinar em disciplinas do ensino básico, tais como Matemática e Física, e no desenvolvimento a partir de disciplinas específicas da Ciência da Computação. No contexto da disciplina de Matemática, pode-se relacionar uma questão dentre nove competências do pensamento computacional. Identificar questões que exploram estas competências pode ser extremamente útil para alunos e professores que possuem interesse em se aprofundar neste tema, pois o estímulo ao PC pode aumentar a capacidade de resolução de problemas. Neste contexto, a concepção de modelos inteligentes capazes de predizer automaticamente competências do PC em questões de matemática seria um grande facilitador no processo de estímulo à resolução de problemas. Neste trabalho, utilizamos uma nova base de dados de questões para extrair características a partir de destaques atribuídos às questões por avaliações manuais advindas de especialistas. A partir destas questões, foram desenvolvidos classificadores utilizando Term Frequency-Inverse Document Frequency (TF-IDF) e Latent Dirichlet Allocation (LDA) como características do modelo, recalculando os valores destas características através do uso dos destaques, com o objetivo de aumentar a importância daqueles termos que pertecem ao trechos destacados pelos avaliadores, e com isso aumentar a eficácia da classificação de questões em relação às competências do pensamento computacional estimuladas.

PALAVRAS-CHAVE

Extração de Características, Pensamento Computacional, Classificação de Textos

1 INTRODUÇÃO

O uso do pensamento computacional tem sido estudado principalmente para estimular a capacidade de resolução de problemas [6], especialmente no ensino fundamental. O mesmo possui duas principais vertentes destacadas na literatura, sendo a primeira focada no desenvolvimento das competências a partir de disciplinas da Ciência da Computação, e a segunda vertente voltada para o desenvolvimento interdisciplinar do PC em conjunto com disciplinas do Ensino Básico, tais como Literatura e Matemática, por exemplo.

No contexto da disciplina de Matemática, o pensamento computacional pode ser utilizado para estimular habilidades, em geral, pouco exploradas pelas questões comumente ministradas em sala de aula, como por exemplo, o reconhecimento de padrões, deduções e o processo de divisão e conquista. Segundo Barr e Stephenson [1], as competências estimuladas com a prática do PC são:

- Coleta de dados;
- Análise de dados;
- Representação de dados;
- Decomposição de problemas;
- Abstração;
- Algoritmos e Procedimentos;
- Automação;
- Paralelização;
- Simulação;

Um maior detalhamento sobre estas competências é apresentado na Seção 2.1.

De acordo com Costa et al [3], pode-se identificar estas competências em enunciados de questões de modo a utilizar os conceitos do pensamento computacional para a sua resolução. Este agrupamento é útil para professores e alunos, facilitando a prática e ensino do pensamento computacional na educação, o que pode trazer benefícios na carreira profissional e na vida pessoal de ambos [7]. Entretanto, de acordo com Costa [4], tal classificação pode ser bastante custosa e trabalhosa de se fazer manualmente, ainda mais considerando um grande número de questões.

Dessa forma, pode-se usar classificadores já existentes, como o proposto por Costa et al [5], que usam o estado da arte de modelos de predição para realizar esta tarefa automaticamente. Estes classificadores atuam utilizando características do texto de cada questão e agrupando-as com questões semelhantes, de modo a fornecer como resultado qual competência cada questão possui. Assim, estas características têm grande impacto na eficiência do classificador, de modo que é fundamental prover boas características para um melhor resultado.

Costa et al [5] propõe um classificador utilizando técnicas de Aprendizado de Máquina e Processamento de Linguagem Natural.

O estudo levou em conta 402 questões, extraídas de exames e olimpíadas de Matemática. Em seu experimento, Costa [5] utilizou o enunciado destas questões para desenvolver os classificadores a partir da extração de informações utilizando a frequência dos termos (TF-IDF) e a distância semântica dos dados, através de *Word Mover Distance - WMD*, como características do modelo. Estas características foram usadas para treinamento do classificador cujo resultados estimulam o uso de processos automáticos de classificação para as competências do Pensamento Computacional.

Neste trabalho, utilizamos uma nova base de dados fornecida por Costa, a partir dos estudos conduzidos em [4]. A base de dados é composta por 80 questões, que contém além do enunciado, trechos destacados por especialistas para cada competência. Estes trechos fazem parte do estudo conduzido por Costa [4], onde voluntários especialistas em Matemática realizam o processo de classificação de questões manualmente. Ao receber uma questão, os avaliadores podem indicar a quais competências à questão pertence, e após isso, indicar os trechos que mais caracterizam a questão dentro da competência alvo.

Fizemos uso destes trechos para extrair novas características de modo a aperfeiçoar o processo de classificação automática de questões. Focamos no uso destes trechos destacados como um fator importante para a eficácia dos classificadores, recalculando o peso das características a partir da presença ou não destes destaques nas questões.

Para isso, utilizamos técnicas como Term Frequency-Inverse Document Frequency (TF-IDF) e Latent Dirichlet Allocation (LDA) para implementar novos classificadores seguindo a abordagem apresentada por Costa [4] com o intuito de melhorar os resultados obtidos pelo mesmo.

Como resultado, conseguimos um significativo aumento na eficácia dos classificadores. Isso sugere que o uso destes trechos de destaque, como um potencializador das características dos modelos, são valiosos e decisivos para um melhor resultado no processo automático de classificação de questões.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta as definições acerca do Pensamento Computacional no contexto da disciplina de Matemática, bem como a definição teórica das técnicas utilizadas.

2.1 O Pensamento Computacional e a Matemática

O pensamento computacional (PC) é um processo de raciocínio que consiste em formular um problema e sua solução em passos que um computador é capaz de realizar. O PC consiste em um conjunto de competências que podem ser usadas para formular soluções para problemas que podem ou não serem executadas por um computador. O PC não se trata, especificamente, do uso de computadores, mas de estratégias para resolução de problemas que podem resultar em uma solução executável por um. As competências nada mais são que formalização do que o PC pode estimular nas pessoas e como esses estímulos podem otimizar a resolução de problemas. Este processo é tão importante que autores o consideram como um potencializador das competências operacionais do ser humano, que pode ser usado em algumas vertentes, como por exemplo, no desenvolvimento interdisciplinar em disciplinas do ensino básico, tais como Matemática e Física.

Wing [9] define o PC como um conjunto de habilidades e capacidades inerentes ao ser humano que não estão relacionados com a manipulação de computadores, mas sim na concepção de problemas que podem ser solucionados por um. As competências listadas em 1 por Barr e Stephenson são apresentadas como o núcleo do Pensamento Computacional. A descrição sobre estas competências apresentadas pelos autores [1] encontram-se abaixo:

- **Abstração**: Uso de variáveis algébricas, identificação de fatos essenciais ao problema.
- **Algoritmos e Procedimentos**: Sequência de passos, divisão em partes.
- **Análise de dados**: Contagem de ocorrências, análises de resultados.
- **Automação**: Uso de ferramentas para solucionar o problema, como calculadora e objetos geométricos.
- **Coleta de dados**: Extração de informações a partir dos dados.
- **Decomposição de problemas**: Divisão do problema em partes, e solucionando a partir da definição de uma ordem.
- **Paralelização**: Aplicação de soluções que permitam a resolução do problema em paralelo, como o cálculo de sistemas lineares e operações em matrizes.
- **Representação de dados**: Uso de histogramas, gráficos de pizza e barra para representar os dados. Uso de listas e grafos.
- **Simulação**: A partir de uma função, mudar os valores das variáveis e observar os resultados da modificação.

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural é a subárea da Inteligência Artificial que estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos. O objetivo do PLN é fornecer aos computadores a capacidade de entender e compor textos.

Um dos problemas mais comuns quando se trata do Processamento de Linguagem Natural é sobre como extrair informações relevantes de um texto. Uma técnica muito utilizada para extração destas características é fazendo uso de um *Bag of Words*. A ideia central desta técnica é dividir o texto em um vetor de palavras, onde cada posição do vetor contém uma determinada palavra e a frequência em que a mesma aparece no documento.

2.2.1 TF-IDF Vectorizer. O TF-IDF é uma das principais soluções que faz uso de uma *Bag of Words*. Esta técnica é aplicada para medir uma palavra chave em um documento e assimilar sua importâcnia baseada no número de vezes em que a palavra aparece em um documento.

Para um termo **t** em um documento **d**, a relevância de um termo no documento é dada por:

$$W_{d,t} = tf_{d,t} \times \log \frac{N}{df}$$

Onde:

- $tf_{d,d}$ é o número de ocorrências de t no documento d;
- df é o número de documento que contém o termo t;
- N é o número total de documentos.

Considerando um exemplo onde em um documento de 100 palavras o termo **"computador"** aparece 15 vezes, a frequência do termo (TF) para a palavra computador será:

$$\frac{15}{100} = 0.15$$

Agora, se assumirmos que temos 10 milhões de documentos e que a palavra computador aparece em mil destes, teremos a frequência inversa (IDF) como sendo:

$$\log \frac{10.000.000}{1.000} = 4$$

Assim, o peso $W_{d,t}$ será o produto entre estes valores.
Logo,

$$W_{d,t} = 0.15 \times 4 = 0.6$$

2.2.2 Latent Dirichlet Allocation (LDA). O LDA é um modelo estatístico que tem como objetivo realizar uma modelagem de tópicos, ou seja, classificar, descobrir e organizar documentos em temas. Cada documento é feito por várias palavras e cada tópico possui várias palavras (i.e. *keywords*) que pertecem ao mesmo.

Segundo Blei et al [2], o LDA é um modelo probabilístico generalista, onde cada documento é representado por uma mistura aleatória de tópicos, onde cada tópico é caracterizado por uma distribuição sobre as palavras.

O princípio básico do LDA é que para cada documento **w** em um documento **D**:

(1) Escolher um N ~ Poisson(ξ).
(2) Escolher um $\theta \sim$ Dir(α).
(3) Para cada uma das N palavras w_n:
   (a) Escolher um tópico $z_n \sim Multinomial(\theta)$.
   (b) Escolher uma palavra w_n de $p(w_n|z_n, \beta$, como sendo uma probabilidade multinomial condicionada ao tópico z_n.

Uma representação gráfica desse modelo pode ser observada na Figura 1:

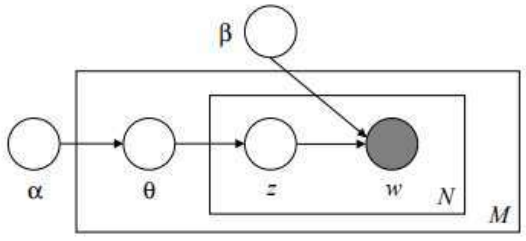

Figura 1: Representação gráfica do LDA

3 METODOLOGIA

O objetivo deste trabalho é analisar questões de matemática e propor novas características, de modo que possam ser utilizadas em classificadores para aperfeiçoar o mecanismo de classificação automática de questões.

A classificação das questões de Matemática de acordo com as competências do Pensamento Computacional é feita de forma manual por especialistas, segundo descrito por Costa et al [3]. Nesta metodologia de rotulação de questões, a classificação é feita a partir da avaliação individual de três especialistas, sendo a competência atribuída à questão se houver maioria (mais de uma) entre as avaliações.

Neste processo de avaliação, além de indicar as competências que se aplicam à questão, os avaliadores também possuem a opção de destacar trechos da questão, que neste trabalho denominaremos de destaques, que mais se sobressaem para determinar a presença da competência marcada.

Por exemplo, para uma determinada questão, cada avaliador atribui competências para a mesma. Para cada competência marcada, o avaliador pode ou não destacar trechos que o levaram a classificar a presença da competência na questão. Após as três avaliações, as competências junto com os destaques demarcados são atribuídos à questão se a competência estiver presença em pelo menos duas avaliações.

Assim, focaremos principalmente na presença destes destaques, de modo a analisar se os mesmos possuem alguma influência no processo de classificação e se podem melhorar os resultados obtidos por Costa et al em [5].

3.1 Conjunto de Dados

O conjunto de dados disponibilizados a partir da metodologia de rotulação descrita anteriormente, é composto por 80 questões de Matemática, sendo estas questões extraídas do estudo conduzido por Costa em [4]. Todas as questões possuem avaliação manual com a presença dos trechos de destaque, totalizando 236 avaliações.

Para este experimento foi desconsiderado as alternativas presentes nas questões. Assim, **apenas o enunciado foi considerado**.

Na Figura 2, é possível observar a relação entre as competências e o número de questões que cada uma possui. Nota-se que as competências mais presentes são "Análise" e "Abstração", enquanto que "Simulação" e "Algoritmos" são as competências menos presentes nesta base de dados.

Importante destacar também que a competência "Paralelização" não está presente no gráfico, uma vez que nenhuma questão foi classificada com esta competência pelos especialistas.

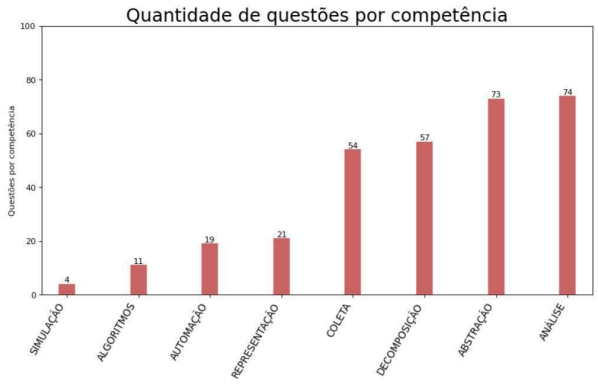

Figura 2: Quantidade de questões por competência.

Além disso, considerando as 80 questões utilizadas, nota-se que as competências "Análise" e "Abstração" estão presentes em 92.50% e

91.25% das questões, respectivamente. Por outro lado, a competência "Simulação" só está presente em 5% das questões

Cada avaliação é realizada por um autor, e possui um campo "competências" indicando quais competências foram marcadas pelo avaliador, o identificador da questão alvo e os trechos destacados para cada competência marcada, se houver. Por exemplo, se o avaliador considerou que uma questão possui as competências "Análise" e "Coleta", então haverá uma coluna para seu trecho destacado para a competência "Análise" e outro para a competência "Coleta".

Na tabela 1, pode-se observar um exemplo com alguns dos campos presentes em uma avaliação:

Tabela 1: Avaliação feita para uma determinada questão onde o avaliador destacou as competências "Análise" e "Coleta" como presentes na questão, e os respectivos trechos de destaque.

Avaliação			
Enunciado	Competências	Análise	Coleta
Em um grupo de pessoas, as idades são: 10, 12, 15 e 17 anos. Caso uma pessoa de 16 anos junte-se ao grupo, o que acontece com a média das idades do grupo?	Análise, Coleta	as idades são	grupo de pessoas

3.2 Pré processamento dos dados

Inicialmente, os dados foram obtidos a partir de um formato *JSON*, onde as informações da questão estavam dispostas sobre tags *HTML*. Assim, esta base de dados foi pré processada de modo a remover todas as tags *HTML* e outros elementos, tais como imagens, gráficos e figuras de modo a obter apenas o texto presente em cada questão.

Os destaques foram extraídos destas tags, e movidos para uma coluna específica de acordo com a competência correspondente, conforme mostrado na tabela 1.

Após isso, foi utilizado um dicionário de *stop words* em Português, de modo a remover do documento palavras muito frequentes que não agregam na classificação da questão. Sinais de pontuação, símbolos, siglas e outras informações desta natureza também foram removidos.

Além disso, foi aplicado um processo de lematização [8] aos dados, que visa transformar as palavras presentes para sua forma mais básica. Assim, palavras diferentes, porém com o mesmo sentido, serão convertidas para a mesma palavra base, aumentando sua frequência no documento.

3.3 Extração de Características

Após o processamento inicial, os enunciados das questões foram utilizados para gerar uma matriz de características, contendo as informações mais relevantes para que sejam utilizadas nos classificadores.

A matriz de características foi construída utilizando vetores extraídos de duas técnicas principais: A frequência dos termos (Term Frequency-Inverse Document Frequency - TD-IDF) e a modelagem atráves de tópicos (Latent Dirichlet Allocation - LDA).

O LDA foi inicialmente utilizado de modo a obter tópicos e agrupar as questões semelhantes. Para a configuração do algoritmo, foram utilizados oito tópicos (se equivalendo a quantidade de competências identificadas na base de dados de 80 questões) e 100 repetições. Para a visualização dos tópicos foi utilizado o método estatístico t-SNE(t-Distributed Stochastic Neighbor Embedding) para representar os dados em um mapa de três dimensões, que pode ser visto na Figura 3.

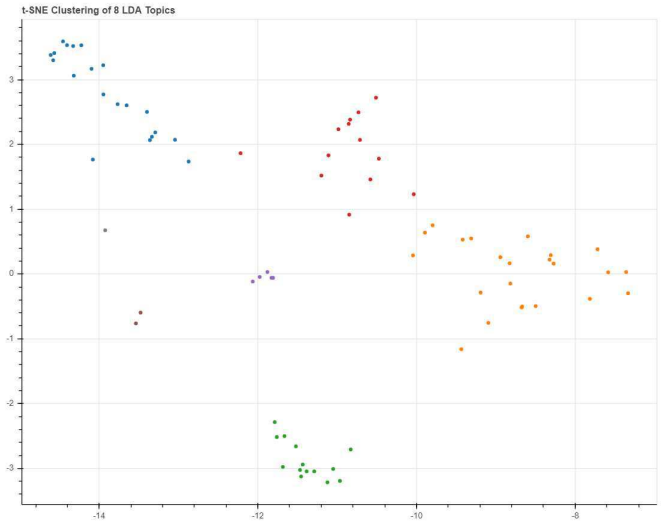

Figura 3: Distribuição dos tópicos encontrados.

Como é possível observar na figura, há uma certa diferenciação entre cada tópico, indicando que as questões possuem diferenças significativas entre si, no que diz respeito às palavras presentes em cada uma delas, e utilizar os tópicos gerados pelo LDA podem ser importantes características para o modelo. Encontram-se também palavras que aparentam não ter relação direta às competências, tais como "vendedor", "consumidor" e "brasileiro", por exemplo. Deste modo, uma abordagem desconsiderando classes de palavras como substantivos pode ser útil em trabalhos futuros para aprimorar o processo de extração dos tópicos utilizando LDA.

Dessa forma, em cada linha da matriz de características é adicionada duas novas características relativas ao LDA: O valor do tópico, e a importância da questão no tópico específico. Na figura 4 pode-se observar os termos mais importantes em cada tópico.

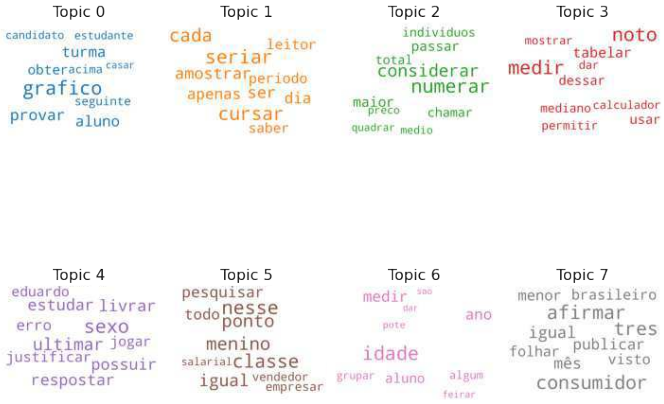

Figura 4: Palavras mais presentes em cada tópico

As características restantes são extraídas a partir dos vetores gerados pelo TD-IDF. Para isso, foram considerados termos compostos por uma ou duas palavras, também conhecidos como **unigramas** e **bigramas**, respectivamente. Por exemplo, para a frase "Calcule o valor" os seguintes termos são considerados:

- **Termos**: ["Calcule", "Calcule o", "o valor", "valor"]

3.4 *Boosting* de Características

Com o intuito de maximizar a importância dos destaques e avaliar sua eficácia, foi gerado um novo conjunto de dados contendo apenas os destaques de cada competência, obtidos a partir das avaliações feito pelos especialistas, onde cada destaque está associado a uma competência.

Esta base de dados foi processada da mesma forma que a base de dados principal: remoção das *stop words*, lematização dos termos, extração dos tópicos com LDA e contagem da frequência dos termos com TF-IDF.

Após isso, este conjunto de dados foi utilizado para incrementar (*boosting*) a base de dados principal, onde recalculamos as colunas de uma base com a de outra para formar um novo conjunto de dados. A ideia é enfatizar no texto principal os destaques demarcadados pela avaliação manual, dando a estes termos uma maior importância no documento.

O processo de recálculo dos termos foi realizado da seguinte forma:

- Do vetor de características dos destaques foram extraídas os 100 termos de maior frequência.
- Como a frequência destes termos eram valores muito baixos, eles foram multiplicados por 10 para que tenham maior influência nos termos alvo.
- Para cada documento na base de dados principal, cada termo tem seu peso recalculado de acordo com a seguinte expressão:

$$P = (P_1 \times 0.2) + (P_2 \times 0.8)$$

Onde:

  - P_1 é o peso do termo no documento original;
  - P_2 é a média da frequência do termo em todos os documentos da base de dados com destaques.

- A proporção de 80% de peso para os destaques e 20% para o termo original foi o parâmetro que gerou o melhor resultado após alguns testes.
- Os termos que não existiam no vetor de características dos destaques tiveram seus pesos mantidos.

Na tabela 2, é possível observar um exemplo para o *boosting* realizado considerando dois termos: "Abaixar" e "Importance".

Tabela 2: Uso dos destaques para incrementar a relevância dos termos.

Base de dados	Abaixar	Importance
Enunciado completo	0.056248	0.9810
Destaques	0.184552	0.574421
Resultado	0.158891	0.65573

3.5 Métricas de Avaliação

Para avaliar os resultados, serão utilizadas quatro métricas comumente utilizadas em problemas de classificação:

- **Acurácia**: Essa métrica é determinada pela relação entre a quantidade de VerdadeirosPositivos (i.e., resultado positivo que está presente na questão alvo) mais a quantidade de VerdadeirosNegativos (i.e., resultado **negativo** que **não** está presente na questão alvo), divididos pelo total de questões. A equação para a Acurácia está descrita em 1:

$$A = \frac{VP + VN}{Total} \tag{1}$$

- **Precisão**: A precisão pode ser entendida como o número de VerdadeirosPositivos sobre a soma entre VerdadeirosPositivos e FalsosPositivos (i.e., resultados positivo que **não** está presente na questão), e sua equação é descrita em 2:

$$P = \frac{VP}{VP + FP} \tag{2}$$

- **Cobertura**: A cobertura é a relação entre o número de VerdadeirosPositivo sobre a soma entre VerdadeirosPositivos e FalsosNegativos (i.e., resultado **negativo** que está presente na questão), e está descrita na equação 3:

$$C = \frac{VP}{VP + FN} \tag{3}$$

- **Medida-F**: A medida-F é um balanceamento entre a Precisão e a Cobertura e sua equação está descrita em 4:

$$P = 2 \times \frac{P \times C}{P + C} \tag{4}$$

4 RESULTADOS E DISCUSSÃO

Para teste e comparação dos resultados utilizamos a mesma base de dados que o estudo conduzido por Costa em [4]. Esta base de dados contém 24 questões de múltiplas competências, conforme mostrado na Tabela 3.

Nesse estudo, foram considerados os mesmos modelos que os utilizados por Costa [4]: Linear Ridge, Logistic Regression, Multinomial Naive Bayes, Random Forest e XGBoost. Estes modelos foram

Tabela 3: Quantidade de questões por competência na base de dados de teste.

Competência	Quantidade de questões
Coleta	22
Análise	14
Representação	23
Decomposição	18
Algoritmos	15
Abstração	19
Automação	6
Paralelização	10
Simulação	4

escolhidos devido à facilidade de implementação e por reunir uma variação de modelos mais tradicionais até modelos mais complexos.

Para que os modelos sejam comparáveis aos resultados apresentados por Costa [4], todos os classificadores deste estudo foram executados utilizando o mesmo ambiente, hiperparâmetros e técnicas de avaliação (por exemplo, validação cruzada), que os utilizados por Costa [4].

Como na base de dados de 80 questões que foi utilizada para treinamento não havia nenhuma questão classificada com a competência "Paralelização", os resultados obtidos pelo novo modelo discutidos nesta seção estão zerados para esta competência.

Na Tabela 4, podemos observar os resultados obtidos pelos modelos de referência, disponibilizado por Costa. Estes modelos de referência foram os mesmos modelos utilizados por Costa em [5], porém em um formato que permitisse que uma comparação fosse realizada. Este formato consistiu em separar os dados de treino e validação, e utilizar as 24 questões citadas no início desta seção como teste.

Nota-se que todos os modelos possuem valores muito parecidos em termos de Cobertura e Medida F, com uma média aproximada de 25%. Os modelos *Linear Ridge* e *XGBoost* se destacam com uma Precisão maior que os demais (o que significa que estes modelos conseguem acertar mais questões corretamente), sendo o ***Linear Ridge* o modelo com o melhor resultado obtido por Costa**.

Tabela 4: Resultados obtidos pelo modelo de referência

Modelo	Acurácia	Precisão	Cobertura	Medida F
Linear Ridge	49,07%	44,44%	25,31%	25,21%
Logistic Regression	47,69%	22,22%	23,82%	22,43%
Random Forest	47,22%	22,35%	23,45%	22,42%
Multinomial NB	49,54%	23,18%	26,46%	24,32%
XGBoost	48,61%	44,31%	25,46%	24,79%

A Tabela 5 mostra os novos resultados obtidos para todos os cinco modelos avaliados, a partir do uso das 80 questões seguindo o procedimento descrito na Seção 3. Em geral, o resultado foi superior em todos os modelos e métricas, atingindo mais de 50% de Acurácia e Precisão em quase todos os modelos. Importante destacar que este resultado poderia ser ainda melhor se a base de dados utilizada possuíse a competência "Paralelização", que por ter zerado, reduziu os percentuais do resultado obtido.

Assim como no modelo de referência, o modelo *Linear Ridge* destaca-se pelos resultados obtidos onde, mesmo um modelo mais simples foi capaz de obter um resultado apenas 1.51% menor em termos de Medida F, e um valor superior de 13.43% de Precisão, em comparação ao melhor modelo obtido, o modelo *Multinomial Naive Bayes*.

Tabela 5: Resultados obtidos pelo novo modelo

Modelo	Acurácia	Precisão	Cobertura	Medida F
Linear Ridge	56,02%	61,25%	45,58%	45,76%
Logistic Regression	54,17%	41,48%	42,38%	38,35%
Random Forest	54,63%	51,52%	45,49%	44,92%
Multinomial NB	55,97%	53,03%	52,80%	46,46%
XGBoost	56,02%	54,20%	46,69%	46,76%

O *Linear Ridge* foi o melhor modelo nos resultados apresentados pelo modelo de referência (Tabela 4) em termos de média e quantidade de competências classificadas (desconsiderando as competências que zeraram o resultado).

Na Figura 5, observa-se os resultados completos obtidos com o *Linear Ridge*, em termos de Acurácia, Precisão, Cobertura e Medida F, para todas as competências.

Nota-se que o novo modelo apresentou um aumento significativo dos resultados base em todas as métricas, com um aumento médio de 25% na Acurácia e Precisão, e uma melhora de mais de 80% na Cobertura e Medida F. Estes resultados indicam que o novo modelo é capaz de acertar mais questões, e errar menos. Em especial, o aumento de 80.08% na Cobertura aponta que o modelo comete menos erros do tipo FalsoNegativo.

Além disso, o novo modelo também foi capaz de classificar as questões que pertencem à competência "Automação", cujo modelo de referência zerou o resultado. Este resultado demostra que o novo modelo conseguiu se adaptar melhor as diferentes questões fornecidas.

LinearRidge (Modelo de Referência)

	Acurácia	Precisão	Cobertura	Medida F
Coleta	16,67	100	9,09	16,67
Análise	54,17	57,14	85,71	68,57
Representação	8,33	100,00	4,35	8,33
Decomposição	41,67	66,67	44,44	53,33
Abstração	66,67	76,19	84,21	80,00
Algoritmos	37,50	0,00	0,00	0,00
Automação	75,00	0,00	0,00	0,00
Paralelização	58,33	0,00	0,00	0,00
Simulação	83,33	0,00	0,00	0,00
Média	49,07	44,44	25,31	25,21

LinearRidge (Novo Modelo)

	Acurácia	Precisão	Cobertura	Medida F
Coleta	79,17	94,74	81,82	87,80
Análise	58,33	58,33	100,00	73,68
Representação	20,83	83,33	21,74	34,48
Decomposição	66,67	85,71	66,67	75,00
Abstração	79,17	79,17	100,00	88,37
Algoritmos	41,67	100,00	6,67	12,50
Automação	75,00	50,00	33,33	40,00
Paralelização	0,00	0,00	0,00	0,00
Simulação	83,33	0,00	0,00	0,00
Média	56,02	61,25	45,58	45,76

Figura 5: Resultados com *Linear Ridge*.

Multinomial Naive Bayes (Modelo de Referência)

	Acurácia	Precisão	Cobertura	Medida F
Coleta	8,33	0,00	0,00	0,00
Análise	58,33	59,09	92,86	72,22
Representação	4,17	0,00	0,00	0,00
Decomposição	54,17	73,33	61,11	66,67
Abstração	66,67	76,19	84,21	80,00
Algoritmos	37,50	0,00	0,00	0,00
Automação	75,00	0,00	0,00	0,00
Paralelização	58,33	0,00	0,00	0,00
Simulação	83,33	0,00	0,00	0,00
Média	49,54	23,18	26,46	24,32

Multinomial Naive Bayes (Novo Modelo)

	Acurácia	Precisão	Cobertura	Medida F
Coleta	75,00	94,44	77,27	85,00
Análise	62,50	64,71	78,57	70,97
Representação	8,33	100,00	4,35	8,33
Decomposição	75,00	75,00	100,00	85,71
Abstração	91,25	60,62	65,00	62,67
Algoritmos	62,50	62,50	100,00	76,92
Automação	70,83	0,00	0,00	0,00
Paralelização	0,00	0,00	0,00	0,00
Simulação	58,33	20,00	50,00	28,57
Média	55,97	53,03	52,80	46,46

Figura 6: Resultados com Multinomial Naive Bayes.

Dentre os cinco modelos implementados com os novos dados, o Multinomial Naive Bayes foi o que obteve os melhores resultados. Apesar de um resultado muito próximo do XGBoost, o Naive Bayes se destaca pois foi capaz de classificar corretamente a competência "Algoritmos" com 100% de Cobertura e 76.92% de Medida F. Além disso, de todos os cinco modelos avaliados neste trabalho, o modelo com Multinomial Naive Bayes foi o único a conseguir classificar corretamente as questões da competência "Simulação", que é a competência menos presente nos dados de teste, com apenas 4 questões, conforme visto na Tabela 3.

Em comparação ao modelo de referência, que não conseguiu classificar seis competências, o novo modelo apenas não conseguiu classificar a competência "Automação", desconsiderando a competência "Paralelização", uma vez que os dados de treino não computavam a presença desta classe.

Os resultados obtidos utilizando *Multinomial Naive Bayes* são mostrados na Figura 6.

Como é possível observar na tabela 6, o modelo proposto utilizando os destaques feitos pela avaliação manual como uma forma de impulso aos dados, apresentou resultados superiores à implementação de referência que apenas utiliza o TF-IDF em conjunto com WMD, como características para o classificador. O resultado foi superior em todas as métricas, onde destaca-se o aumento percentual de 128.77% na precisão, o que indica que houve uma melhora significativa na predição positiva das classes, ou seja, o modelo **consegue identificar mais questões corretamente**.

Tabela 6: Resultados obtidos do modelo proposto utilizando *Naive Bayes*

Métrica	Modelo de Referência	Novo Modelo	Melhora em %
Acurácia	49.54%	55.97%	12.97%
Precisão	23.18%	53.03%	128.77%
Cobertura	26.46%	52.80%	99.54%
Medida F	24.32%	46.46%	91.03%

Analisando mais detalhadamente, na Figura 7, apresenta-se uma matriz de confusão, onde podemos obserar uma comparação específica da competência "Coleta de Dados". Nela, o eixo X representa os resultados reais das questões, enquanto que o eixo Y representa os resultados previstos pelo modelo. À esquerda, encontra-se os resultados do modelo de referência, enquanto que na direita estão os resultados do novo modelo. Nota-se que o modelo de referência não foi capaz de prever as questões desta competência, marcando todas as 22 questões verdadeiras como "Não". O novo modelo foi capaz de classificar 17 questões como verdadeiroPositivo. Essa característica do modelo em marcar mais questões corretamente como "Sim" pode ser observada na grande maioria das competências, o que confirma a melhora nos resultados.

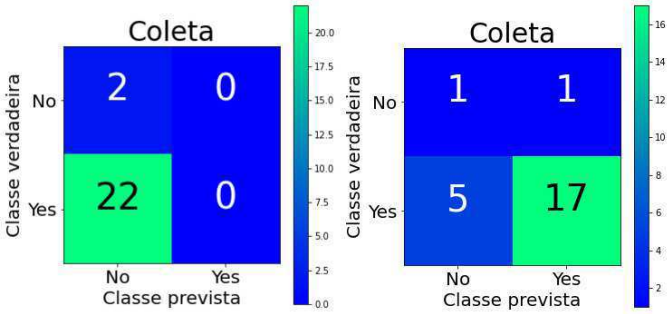

Figura 7: Distribuição dos tópicos encontrados.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado uma nova abordagem para o processo de classificação de questões de Matemática utilizando dados rotulados para treinar um classificador de modo a identificar as competências do Pensamento Computacional. A fim de prover modelos base para comparação, foram utilizados os dados, resultados e experimentos do estudo conduzido por Costa [4]. Nossa metodologia consistiu no uso dos trechos destacados das questões, atribuídas por especialistas através de uma avaliação manual.

O procedimento consistiu do uso destes destaques como características do modelo, com a adição de um pré processamento utilizando tópicos LDA, incrementando assim as características do modelo original. Tal abordagem permitiu um aumento significativo no desempenho dos cinco classificadores utilizados, onde notamos que a melhora dos resultados foi causada principalmente pelos dados utilizados nos modelos ao invés dos classificadores utilizados.

Os resultados obtidos nos indicam que o uso dos destaques é valioso para o problema e que produzir tais trechos cada vez mais completos e em maior quantidade por ser de grande valia para estudos na área. Os tópicos extraídos pelo LDA também se mostraram importantes para o classificador, uma vez que o algoritmo foi capaz de identificar importantes relações entre as palavras, indicando que cada competência está relacionada a um grupo de palavras (tópico) específico.

Para trabalhos futuros, busca-se aperfeiçoar o uso dos destaques, obtendo uma base de dados maior, através do mesmo procedimento de avaliação manual apresentado na Seção 3. Com isso, busca-se realizar novos experimentos com a presença da competência "Paralelização" que não foi explorada neste trabalho e realizar uma análise mais profunda a respeito dos destaques, estudando possíveis interseções entre as avaliações de diferentes avaliadores, além do uso de novas técnicas como *data augmentation* para incrementar o número de trechos. Por fim, pretende-se explorar ainda mais o uso de tópicos LDA, restringindo as classes de palavras utilizadas e utilizando um conjunto de *stop words* maior para que a modelagem seja feita apenas para palavras relevantes às competências, e com isso, gerar uma base de dados ainda mais fiel para cada competência, incrementando ainda mais a eficácia do classificador aqui demostrado.

6 AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Daniel e Edilene, por todo o amor, apoio e por tudo que fizeram e fazem por mim.

A minha namorada, Mariza Stefane, por sempre me apoiar, motivar e por ser essa pessoa incrível que tanto me inspira.

Aos meus amigos e colegas de curso, por todos os momentos e aprendizados que compartilhamos.

Ao professor Cláudio Campelo, pela orientação neste trabalho e por todas as oportunidades e ensinamentos ao longo destes anos.

A Erick Costa, pelo auxílio na revisão deste trabalho.

E agradeço à UFCG, por todos os diversos momentos que vivi e por ter sido minha segunda casa durante tantos anos.

REFERÊNCIAS

[1] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads* 2 (03 2011). https://doi.org/10.1145/1929887.1929905

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3, null (March 2003), 993–1022.

[3] Erick Costa, Livia Maria Rodrigues Campos, and Dalton Guerrero. 2017. Computational thinking in mathematics education: A joint approach to encourage problem-solving ability. 1–8. https://doi.org/10.1109/FIE.2017.8190655

[4] Erick John Fidelis Costa. 2021. *Um Arcabouço Metodológico para o Desenvolvimento do Pensamento Computacional na Matemática Apoiado por Técnicas de Aprendizado de Máquina e Recuperação da Informação.* PhD dissertation. Universidade Federal de Campina Grande. No prelo.

[5] Erick J. F. Costa, Cláudio E. C. Campelo, and Lívia M. R. Sampaio Campos. 2019. Automatic Classification of Computational Thinking Skills in Elementary School Math Questions. In *2019 IEEE Frontiers in Education Conference (FIE).* 1–9. https://doi.org/10.1109/FIE43999.2019.9028499

[6] Norma Suely Gomes Allevato Lourdes De La Rosa Onuchic. 2011. Pesquisa em resolução de problemas: caminhos, avanços e novas perspectivas. *Bolema-Mathematics Education Bulletin* (2011), 73–98.

[7] James Lu and George Fletcher. 2009. Thinking about Computational Thinking. *ACM Sigcse Bulletin* 41, 260–264. https://doi.org/10.1145/1539024.1508959

[8] Cambridge University Press. 2008. Stemming and lemmatization. https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

[9] Jeannette Wing. 2006. Computational Thinking. *Commun. ACM* 49 (03 2006), 33–35. https://doi.org/10.1145/1118178.1118215