PROPOSTA DE SISTEMA DE RECOMENDAÇÃO COLD START PARA A PLATAFORMA REACLOUD UTILIZANDO WORD EMBEDDINGS

ULIAN GABRIEL ALFF RAMIRES[1], GUSTAVO HENRIQUE ROOS[2], ULISSES BRISOLARA CORRÊA[3], TIAGO THOMPSEN PRIMO[4]

[1] *Universidade Federal de Pelotas – ugaramires@inf.ufpel.edu.br*
[2] *Universidade Federal de Pelotas – ghroos@inf.ufpel.edu.br*
[3] *Universidade Federal de Pelotas – ub.correa@inf.ufpel.edu.br*
[4] *Universidade Federal de Pelotas – tiago.primo@inf.ufpel.edu.br*

1. INTRODUÇÃO

Atualmente, professores e outros profissionais da educação enfrentam desafios ao tentar encontrar, organizar e compartilhar materiais educativos que estejam espalhados em diversas plataformas e fontes, isso dificulta a agregação de material educacional de forma eficiente. O *ReaCloud* busca resolver o problema da dispersão e dificuldade de acesso a recursos educacionais em diferentes formatos e fontes; propondo centralizar esses recursos em um único repositório, facilitando a busca e recomendação de conteúdos educacionais, além de permitir a colaboração, possibilitando com que cada usuário possa publicar os seus recursos.

A ferramenta hoje encontra-se já em ambiente de produção, e prestes a ser divulgada para o público, porém, existe um problema; não existe ainda na plataforma uma massa de dados considerável, nem de usuários e nem de recursos. isso limita fortemente a nossa capacidade de gerar recomendações utilizando métodos tradicionais. Esse problema é conhecido na área de sistemas de recomendação por *cold start* (SCHEIN et al., 2002). Para resolver esse problema, propomos o uso de *word embeddings* gerados por um *encoder* baseado no modelo *BERT* (*Bidirectional Encoder Representations from Transformers*) (DEVLIN et al., 2019), com o objetivo de capturar a semântica dos dados textuais atrelados aos recursos da plataforma, e a partir disso, medir a similaridade destes embeddings utilizando similaridade de cosseno, podemos então, alcançar um método viável de recomendação no contexto do *ReaCloud*. O objetivo deste trabalho é propor um método de recomendação e ranqueamento de itens para uma plataforma de recursos educacionais em estágio inicial do seu ciclo de vida e investigar a viabilidade dos modelos de recomendação citados.

2. METODOLOGIA

A interface de usuário da plataforma *ReaCloud* foi projetada seguindo um guia de design e interface de usuário (NOGUEIRA, 2022), e na tela de visualização de recursos, que proporciona aos usuários uma visão de um material educacional específico, existe uma aba lateral dedicada a apresentar itens relacionados, permitindo ao usuário explorar recursos de provável interesse rapidamente. Como ainda não temos dados suficientes na plataforma para implementar um sistema de recomendação baseada em filtragem colaborativa (PRIMO, 2013), foi proposto um experimento utilizando um *dataset* de domínio relacionado para verificar a possibilidade de um motor de recomendação viável nesse contexto. Para conduzir esse

experimento, utilizamos o *Coursera Course Dataset* (SIDDHARTH, 2020), o *dataset* foi coletado a partir do site oficial da *Coursera* e contém informações sobre 890 cursos disponíveis na plataforma sobre diversos temas. O *dataset* possui colunas de título, organização, tipo de certificado, *rating*, dificuldade do curso e número de alunos matriculados. Como estamos interessados apenas no domínio do assunto e na similaridade entre os assuntos dos cursos, utilizamos somente a coluna de título.

Primeiramente utilizamos um método clássico para detectar similaridade entre dados textuais; O *TF-IDF (Term Frequency-Inverse Document Frequency)* é uma técnica de processamento de linguagem utilizada para medir a importância de um termo em um conjunto de corpus textual. A relevância de um termo é determinada por sua frequência em um documento (*TF*) e sua raridade em todo o corpus (*IDF*).

Utilizando *TfidfVectorizer*, da bilioteca *Scikit-learn*, transformamos os títulos dos cursos em uma matriz *TF-IDF*; cada linha da matriz corresponde a um título de curso, e cada coluna representa uma palavra única do vocabulário. O valor em cada célula indica a importância da palavra no título do curso. Ao buscar por um termo, são ranqueados em ordem crescente os textos que possuem maior valor de similaridade com o termo buscado. O problema dessa essa abordagem no contexto do *ReaCloud* é o caso onde a busca é feita por um termo fora dessa matriz; ao adicionar um novo material, as recomendações relacionadas a ele não funcionariam até que uma nova matriz *TF-IDF* seja criada, o que é impraticável pensando na escalabilidade do projeto e em grandes quantidades de recursos sendo adicionados em tempo real.

Pensando nesse problema, seguimos outra estratégia baseada em *embeddings* dos títulos dos cursos, utilizando o *SentenceTransformer* (REIMERS; GUREVYCH, 2019), especificamente a versão *distilbert-base-nli-mean-tokens*, permitindo que representemos o dado textual em um espaço vetorial, capturando não apenas as palavras de forma individual, mas também o contexto delas de forma mais profunda. O nível de relação item-item é calculado utilizando similaridade de cosseno, medindo o quão semelhante são dois vetores distintos.

A partir da codificação dos títulos dos cursos utilizando o *SentenceTransformer*, cada título é convertido em um vetor de *embeddings*. Essa representação vetorial é, então, usada para calcular a similaridade entre os cursos, empregando a métrica mencionada. Nessa estratégia, além das recomendações agora capturarem melhor a semântica dos termos buscados, qualquer termo que for transformado em *embedding* pode ser buscado e comparado aos demais títulos.

3. RESULTADOS

Analisando os quadros 1 e 2 uma comparação empírica dos modelos TF-IDF e *embeddings* para o termo "*Artificial Intelligence*", é possível identificar que os *embeddings* trazem recomendações relevantes mesmo quando o termo não é diretamente encontrado no título; em contrapartida, o sistema baseado em *TF-IDF* busca relações mais diretas com as palavras buscadas. Já no caso da palavra "*Programming*", como mostrado nos quadros 3 e 4, o sistema *TF-IDF* retornou recomendações adequadas, fazendo encontro direto com algum termo dos títulos recomendados, enquanto o método de *embeddings* capturou melhor a semântica do termo, resultando em recomendações mais variadas e relevantes.

ID	Curso
520	Introduction to TensorFlow for Artificial Intelligence
205	Data Warehousing for Business Intelligence
890	Financial Instruments for Private Investors.
293	Excel Skills for Business: Essentials
302	Feminism and Social Justice

Figura 1: TF-IDF: Busca por itens similares ao termo "Artificial Intelligence"

ID	Curso
855	Virtual Reality
563	Machine Learning
564	Machine Learning
469	Introduction to Artificial Intelligence (AI)
38	Algorithms
704	Robotics

Figura 2: BERT + Similaridade de Cosseno: Busca por itens similares ao termo "Artificial Intelligence"

ID	Título do Curso
510	Introduction to Programming in C
122	C for Everyone: Programming Fundamentals
673	Programming Fundamentals
675	Programming with Google Go
479	Introduction to Computer Programming

Figura 3: Recomendações TF-IDF com busca por termo: "Programming"

ID	Curso
673	Programming Fundamentals
691	R Programming
635	Object-Oriented Design
449	Interaction Design
38	Algorithms

Figura 4: Recomendações BERT + Similaridade Cosseno com busca por termo: "Programming"

No *ReaCloud*, temos outros tipos de dados textuais no item, como a descrição do recurso e instruções de uso, que podem ser concatenados com o título na etapa de tokenização, extraindo ainda mais da semântica do item e podendo alcançar melhores resultados. A etapa de implantação no sistema real virá logo em seguida, envolvendo o uso de um banco de vetores, como o *Qdrant* (Qdrant, 2021) para armazenamento desses *embeddings* a medida que a massa de dados cresce.

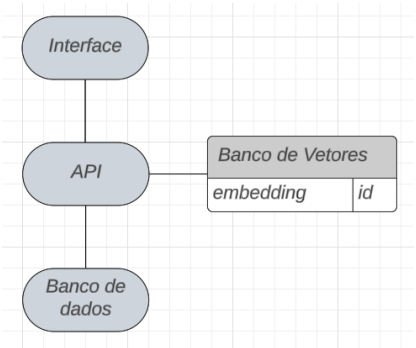

Figura 5: Diagrama simplificado da arquitetura da plataforma ReaCloud com a adição do banco de vetores

Quando um item é inserido, ele será enviado para a *API*, que gera o *embedding* do item. Em seguida, o *embedding* é armazenado no banco de vetores, e os detalhes do item são salvos no banco de dados. Para a consultar de itens similares, o usuário fará uma solicitação na interface, que consultará a *API*. A *API*, então, consultará o banco de vetores encontrando os vetores mais próximos usando similaridade de cosseno. Então, a *API* retorna os itens recomendados à interface.

4. CONCLUSÃO

Discutimos a implementação de um sistema de recomendação para a plataforma *ReaCloud*, utilizando como base um experimento com um corpus de dados de domínio semelhante, abordando as limitações do método *TF-IDF*, que embora eficiente para capturar a relevância de termos em documentos existentes, apresenta desafios na escalabilidade e na adaptação a novos conteúdos. Isso ocorre porque, sempre que um novo material é adicionado, é necessário recalcular a matriz *TF-IDF*, o que pode se tornar impraticável em ambientes dinâmicos, onde uma quantidade substancial de recursos é constantemente introduzida. Em contraste, a utilização de encoders de texto, como o *BERT*, combinado com a similaridade de cosseno, proporciona uma abordagem mais interessante nesse cenário. Essa metodologia permite que o sistema capture relações semânticas entre os dados de texto, possibilitando a recomendação de conteúdos relacionados mesmo quando novos materiais são incorporados à base de dados.

4. Referências

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2019.

NOGUEIRA, M. **Proposta de interface para o indexador de objetos de aprendizagem ReaCloud**. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2022.

PRIMO, T. T. **Método de representação de conhecimento baseado em ontologias para apoiar sistemas de recomendação educacionais**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013. Doutorado em Computação. Disponível em: <http://hdl.handle.net/10183/83654>.

Qdrant. **Qdrant: GitHub repository**. 2021. <https://github.com/qdrant/qdrant>. Accessed: 2024-10-09.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2019. Disponível em: <https://arxiv.org/abs/1908.10084>.

SCHEIN, A. I.; POPESCUL, A.; UNGAR, L. H.; PENNOCK, D. M. Methods and metrics for cold-start recommendations. In: **Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)**. New York, NY, USA: ACM, 2002. p. 253–260.

SIDDHARTH. **Coursera Course Dataset**. 2020. Accessed: 2024-10-09. Dataset scrapped from Coursera website for an intelligent course recommendation system. Disponível em: <https://github.com/Siddharth1698/Coursu>.