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

CAIO DAVI PEREIRA DA SILVA

UMA ABORDAGEM COM BUSCA SEMÂNTICA E MODELO DE LINGUAGEM PARA RESPONDER A PERGUNTAS DO CONTEXTO @CCC

CAMPINA GRANDE - PB

2023

CAIO DAVI PEREIRA DA SILVA

UMA ABORDAGEM COM BUSCA SEMÂNTICA E MODELO DE LINGUAGEM PARA RESPONDER A PERGUNTAS DO CONTEXTO @CCC

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 : Leandro Balby Marinho

CAMPINA GRANDE - PB

2023

CAIO DAVI PEREIRA DA SILVA

UMA ABORDAGEM COM BUSCA SEMÂNTICA E MODELO DE LINGUAGEM PARA RESPONDER A PERGUNTAS DO CONTEXTO @CCC

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:

Leandro Balby Marinho
Orientador – UASC/CEEI/UFCG

Cláudio de Souza Baptista
Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de JUNHO de 2023.

CAMPINA GRANDE - PB

RESUMO

A ciência de extrair dados a partir de informações textuais avança consideravelmente. Com a rápida evolução tecnológica da área de Processamento de Linguagem Natural (PLN), os modelos de linguagem são capazes de fornecer resultados completos e concisos a partir de entradas textuais. Atualmente, informações sobre a graduação em Ciência da Computação na UFCG se apresentam diluídas em diversos documentos, planilhas eletrônicas e links da web, podendo resultar em obstáculos para o aluno obter informações precisas referentes ao curso e seus respectivos aspectos, gerando assim uma potencial dificuldade para a resolução de dúvidas. Diante disso, este trabalho se propõe a utilizar técnicas da área de PLN e Machine Learning a fim de construir uma ferramenta capaz de gerar respostas automáticas para questionamentos no contexto da graduação em Ciência da Computação na Universidade Federal de Campina Grande.

A SEMANTIC SEARCH AND LANGUAGE MODEL APPROACH TO ANSWERING @CCC CONTEXT QUESTIONS

ABSTRACT

The science of extracting data from textual information has made significant advancements. With the rapid technological evolution in the field of Natural Language Processing (NLP), language models are able to provide complete and concise results from textual inputs. Currently, information about the undergraduate program in Computer Science at UFCG is scattered across various documents, spreadsheets, and web links, which can result in obstacles for the student to obtain accurate information about the course and its aspects. This situation can potentially create difficulties in the resolution of doubts. In light of this, this work proposes to utilize techniques from the field of NLP and Machine Learning to build a tool capable of generating automatic answers to questions in the context of the undergraduate Computer Science course at the Federal University of Campina Grande.

Uma abordagem com Busca Semântica e Modelo de Linguagem para responder a perguntas do contexto @CCC

Caio Davi Pereira da Silva
Universidade Federal de Campina Grande
caio.silva@ccc.ufcg.edu.br

Leandro Balby Marinho
Universidade Federal de Campina Grande
lbmarinho@computacao.ufcg.edu.br

RESUMO

A ciência de extrair dados a partir de informações textuais avança consideravelmente. Com a rápida evolução tecnológica da área de Processamento de Linguagem Natural (PLN), os modelos de linguagem são capazes de fornecer resultados completos e concisos a partir de entradas textuais. Atualmente, informações sobre a graduação em Ciência da Computação na UFCG se apresentam diluídas em diversos documentos, planilhas eletrônicas e links da *web*, podendo resultar em obstáculos para o aluno obter informações precisas referentes ao curso e seus respectivos aspectos, gerando assim uma potencial dificuldade para a resolução de dúvidas. Diante disso, este trabalho se propõe a utilizar técnicas da área de PLN e *Machine Learning* a fim de construir uma ferramenta capaz de gerar respostas automáticas para questionamentos no contexto da graduação em Ciência da Computação na Universidade Federal de Campina Grande.

Palavras-Chave

Processamento de Linguagem Natural, *instruction-tuned*, @ccc, busca semântica, *Machine Learning*, LLaMA.

1. INTRODUÇÃO

Grandes Modelos de Linguagem (GML) treinados em enormes corpos de texto, como o *Generative Pre-trained Transformer* 3 (GPT-3) e o *Large Language Model MetaAI* (LLaMA), demonstram ser eficazes em diversas atividades de PLN a partir de instruções textuais ou de alguns poucos exemplos [1]. Neste escopo, a tarefa de perguntas e respostas ganha destaque, na qual esses modelos conseguem atingir uma ótima performance na construção de textos em linguagem natural a partir de um prompt de entrada com instruções bem estabelecidas.

Com o avanço do desempenho desses modelos de linguagem surge a necessidade que se tem de ajustar o funcionamento do GML a instruções de contextos mais específicos. Uma alternativa amplamente utilizada é a técnica de *fine-tuning* baseada em instrução, ou apenas: *instruction-tuned*, permitindo que uma boa performance seja mantida para assuntos de domínios particulares. Somado a este processo de adaptação, a fim de reduzir o esforço humano na construção de novos dados para o ajuste, desenvolve-se um processo semi-automático de geração de novas instruções sobre o contexto em que se espera ter uma boa performance, utilizando um modelo de linguagem (ML) de larga escala [2].

Como tais modelos tem bilhões de parâmetros, para que um ajuste possa ser realizado sem a demanda de um grande custo computacional, se faz necessário reduzir a complexidade durante o *fine-tuning* com o uso de algumas técnicas disponibilizadas na biblioteca *Parameter-Efficient Fine-Tuning* (PEFT) [3], diminuindo o tempo e a necessidade de *hardwares* potentes para realizar um ajuste de modelo.

Com o objetivo de realizar o refinamento de um modelo de linguagem para que seja capaz de responder sobre perguntas do contexto de Ciência da Computação na UFCG, foi construído, de forma semi-automatizada, um conjunto de dados contendo informações sobre: disciplinas da graduação, períodos letivos, pré-requisitos, áreas de atuação de um profissional graduado, atividades complementares, regulamentos da instituição e informações gerais sobre matrícula.

A fim de garantir uma maior taxa de acerto nas respostas, é utilizado um mecanismo de recuperação de informações referentes ao contexto da pergunta inserida. Um modelo de dimensão menor baseado em *Bidirectional Encoder Representations from Transformers* (BERT) faz parte desse mecanismo, possibilitando que a semântica das palavras seja considerada durante a pesquisa, sendo realizada uma busca semântica. Caso não seja encontrada nenhuma informação, o GML irá responder com o conhecimento que adquiriu durante o *fine-tuning* baseado em instrução. O contexto retornado fará parte do *prompt* que será enviado como entrada ao modelo, a fim de que a resposta adequada seja obtida.

A partir do LLaMA, um Grande Modelo de Linguagem *open source* [1], foi proposta uma ferramenta de perguntas e respostas sobre o contexto da graduação de Ciência da Computação na UFCG. A construção desta ferramenta é baseada em ajustes de modelo de linguagem com instruções geradas de forma semi-automática fazendo uso de técnicas da biblioteca PEFT. Neste trabalho é abordada a metodologia utilizada para a geração das instruções, construção da ferramenta de busca semântica, treinamento do ML e a avaliação dos resultados.

2. MOTIVAÇÃO

As informações referentes à graduação em Ciência da Computação na UFCG se apresentam diluídas em diversos documentos, planilhas eletrônicas e links da *web*. Alunos relatam dúvidas sobre disciplinas, processo de matrícula, atividades complementares, entre outros temas do mesmo contexto. Além disso, com o novo Plano Pedagógico do Curso, surgiu a necessidade de traçar o perfil do estudante egresso baseado nas disciplinas optativas cursadas durante a graduação. Nesta seção são detalhadas as informações sobre a graduação, assim como a

importância de traçar o perfil profissional considerando as disciplinas do curso.

2.1 Disposição das informações sobre a graduação

Informações sobre regulamentos, procedimentos e disciplinas da graduação são do interesse de todos os alunos. Atualmente, as ementas dos componentes curriculares já cursadas estão dispostas no Controle Acadêmico, porém, os conteúdos abordados em todas as disciplinas estão disponíveis apenas no *site*[1] da graduação, assim como bibliografia, pré-requisitos e informações de cada disciplina.

Para se informar sobre como funcionam as Atividades Complementares durante a graduação, é necessário que o aluno acesse a planilha ou leia o documento que regulamenta estas atividades.

Instruções sobre como proceder diante de processos administrativos como Regime de Exercício Domiciliar, Antecipação de Estudos e Reativação de Vínculo estão disponíveis em regulamentos e páginas de *sites* diversos.

Mediante a esses diferentes locais que as informações estão hospedadas, agrupar esses conteúdos em uma aplicação que faz uso de um Modelo de Linguagem capaz de responder ao estudante uma dúvida, contribui com o âmbito acadêmico da graduação, tanto para discentes quanto para os docentes.

2.2 Perfil do estudante egresso

O curso de Ciência da Computação na UFCG dispõe de cerca de 50 disciplinas optativas, das quais o aluno deve escolher 11 ao longo da sua formação para cursar.

As ementas destes componentes curriculares do grupo de optativas abrangem conteúdos que contribuem para a especialização de diversas áreas de atuação do profissional graduado no curso.

Sendo assim, associar um conjunto de componentes curriculares a determinadas carreiras da Ciência da Computação contribui no sentido de guiar as escolhas dos discentes durante a graduação.

3. FUNDAMENTAÇÃO TEÓRICA

Esta seção irá discutir conceitos teóricos de Busca Semântica, Transformers, Sentence Embeddings, do modelo LLaMA e sobre *Instruction-Tuning*.

3.1 Busca Semântica

A Busca Semântica trata do processo de recuperação de informação, considerando: o significado, a relação entre os termos utilizados na construção da query e a intenção do usuário. Com isso, propõe uma melhora de precisão em relação a consultas realizadas apenas de forma sintática.

Uma abordagem amplamente utilizada para realizar a Busca Semântica em uma base de dados inicia-se com a indexação dos elementos já existentes, esses elementos que originalmente são textos, passam a ser representados por vetores, podendo ser Bag-Of-Words, TF-IDF (Term Frequency-Inverse Document Frequency), Word Embeddings e Sentence Embeddings, esta última será abordada na próxima subseção. A partir destas representações vetoriais indexadas em um espaço vetorial, com o objetivo de recuperar o texto que mais se aproxima semanticamente da entrada inserida pelo usuário, algumas métricas podem ser utilizadas a fim de obter os vetores que estão próximos à representação da query.

Podemos destacar duas formas para calcular as distâncias entre as representações de textos em um espaço vetorial, são elas: a distância L2, ou norma L2, e a Similaridade de Cosseno.

- Distância L2: Distância em linha reta entre 2 pontos de um espaço n-dimensional.

  $d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$, onde:

  p, q = Vetores do espaço;

  q_i, p_i = Componentes dos vetores do espaço vetorial;

  n = Dimensão do espaço;

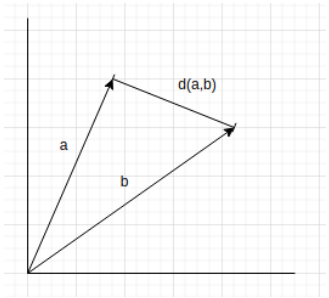

Figura 1: Representação da distância L2.

- Similaridade de Cosseno: Cosseno do ângulo existente entre os 2 vetores

  $cos(\theta) = \frac{A.B}{||A||.||B||}$, onde:

  A, B = Vetores do espaço vetorial;

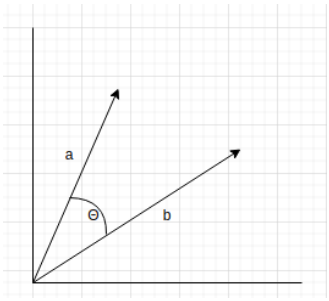

Figura 2: Representação da Similaridade de Cosseno.

3.2 Transformers

Introduzidos pelo artigo "*Attention is All You Need*" por Vaswani et al (2017) [4], os modelos Transformers são uma classe de modelos de aprendizado de máquina que se destacaram em várias tarefas de PLN, sendo amplamente utilizados, se tornando uma das arquiteturas mais populares desta área.

A principal ideia por trás dos modelos Transformers é a atenção, um mecanismo que permite aos modelos focalizar em diferentes partes da entrada durante o processamento, diferenciando-se das arquiteturas sequenciais tradicionais, permitindo que todos os

[1]https://www.computacao.ufcg.edu.br/

elementos de entrada sejam processados de forma simultânea, capturando as dependências entre eles.

Os Transformers consistem em uma pilha de camadas repetitivas chamadas de codificadores e decodificadores. Os codificadores mapeiam a entrada em uma representação intermediária e os decodificadores geram uma saída baseada nessa representação.

A camada de autoatenção permite que o modelo atribua pesos diferentes a cada elemento, considerando a relevância e as relações entre eles, capturando dependências de longo alcance e aprendendo representações mais ricas.

Os modelos Transformers têm sido aplicados em várias tarefas de PLN, como tradução automática, sumarização de texto, geração de texto, classificação de sentimentos e tarefas de Pergunta e Resposta. Com a evolução de hardware e de pesquisas na área, os Modelos de Linguagem com a arquitetura Transformer estão atingindo ótimos resultados, como é o caso do GPT-3 e do LLaMA.

3.3 Sentence Embeddings

Conhecido em português como Representações de Sentenças Distribuídas, o conceito de *Sentence Embeddings* permitiu um grande avanço na área de Processamento de Linguagem Natural, pois a possibilidade de capturar o significado e a semântica de frases e sentenças acarretou o desenvolvimento de técnicas para criar representações vetoriais densas para sentenças por completo.

Quoc et al. introduz o conceito de *Paragraph Vector*, no qual é criado um vetor que representa todo o parágrafo e concatenado com os vetores de cada palavra, com o objetivo de prever o próximo token, fornecendo mais contexto em relação ao uso de apenas as representações das palavras [5].

Ao estabelecer o modelo de linguagem BERT, Devlin et al. apresenta uma nova forma de representação de sentenças, introduzindo o símbolo especial [CLS] no início de cada *embedding* do texto de entrada. Este novo token após treinado agrega a representação da sentença inteira em relação ao contexto em que ela está inserida. [6]

Para obter o vetor da representação completo, cada token é uma soma do seu próprio *embedding* com o *segment embedding*, responsável por indicar a qual sentença aquele token faz parte e o *position embedding*, que codifica a posição relativa de cada palavra na sequência. Somando estes *embeddings*, temos a representação de cada palavra. O *Sentence Embedding* consiste na união destas representações com o símbolo especial CLS, a Figura 3 a seguir mostra a construção do vetor.

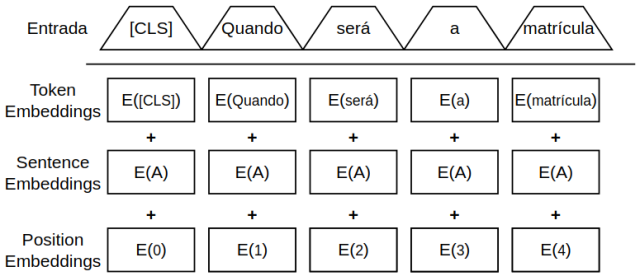

Figura 3: Representação da Entrada

Fonte: Adaptada de [4]

3.4 Large Language Model MetaAI (LLaMA)

Touvron et al. (2023) introduz LLaMA, uma coleção de modelos de linguagem que variam entre 7 e 65 bilhões de parâmetros, treinados em trilhões de tokens retirados de dados disponíveis de forma pública, sem o uso de *datasets* exclusivos e inacessíveis [1].

Ao demonstrar boas performances em responder instruções textuais, acreditou-se que os modelos de linguagem precisavam escalar em tamanho para melhorar seus desempenhos. O LLaMA segue uma abordagem contrária a esta ideia e prova que modelos menores treinados em mais dados demonstram ser eficazes com uma dimensão reduzida em relação aos já existentes. Além disso, GMLs treinados com uma quantidade menor de parâmetros possibilita a execução de inferências com um custo computacional bastante reduzido, democratizando o acesso a estes modelos, uma vez que podem ser executadas com configurações de apenas uma Unidade de Processamento Gráfica (GPU).

Durante o treino do LLaMA-7B foi observado uma melhoria da performance mesmo após 1 trilhão de tokens, enquanto trabalhos relacionados sugerem que para o treinamento de um GML de 10 bilhões de parâmetros deve ser utilizado apenas 200 bilhões. Conforme podemos analisar na figura 4, o *loss* durante o treino continua sendo reduzido, demonstrando que a abordagem de menores modelos com mais dados é eficaz.

Além de inserir mais dados durante o treino, também foram realizadas algumas otimizações na implementação do código, com o objetivo de diminuir o uso computacional e de memória.

Como prova de conceito, os modelos LLaMA foram avaliados em diversos *datasets* de tarefas distintas. Na maioria dos testes o LLaMA-13B superou o GPT-3, que tem uma dimensão 10 vezes maior, enquanto o LLaMA-65B compete com a performance de modelos com dimensão por volta de 500 bilhões de parâmetros.

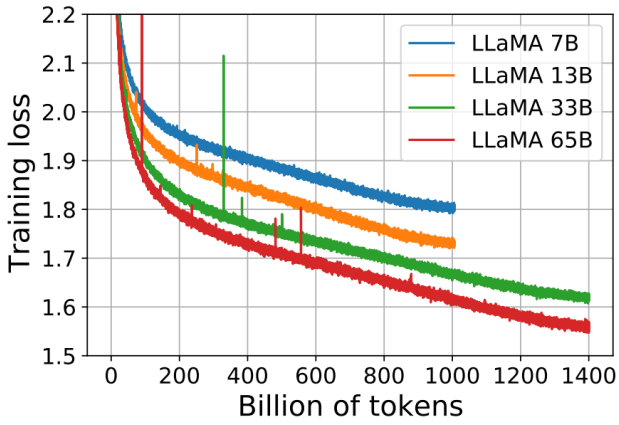

Figura 4: *Loss* durante o treino por tokens processados

Fonte: LLaMA: Open and Efficient Foundation Language Models. [1]

3.5 Instruction Tuning

Grandes Modelos de Linguagem têm demonstrado uma grande capacidade generativa e notável compreensão de texto em Linguagem Natural [7]. GMLs com a capacidade de responder a instruções desempenhando uma alta acurácia, como o modelo do

ChatGPT, ganham destaque e atenção, tanto na indústria como no meio acadêmico, pois seguindo instruções fornecidas pelo usuário, tais ferramentas são capazes de gerar respostas contextualizadas e úteis sobre diversos temas.

Com o objetivo de ajustar o modelo e torná-lo eficaz em responder aos comandos de usuários, é realizado o *Instruction Tuning*, uma abordagem de *fine-tuning* focada em instruções que vem sendo amplamente utilizada. Os dados de treino para este tipo de ajuste contém uma instrução, a resposta correta e, caso seja necessário, um texto que contextualiza a instrução para o Modelo de Linguagem.

Uma das barreiras deste ajuste seria a construção manual dos dados para adaptação dos GMLs. Com isso, Wang et al. (2022) [2] propõe uma abordagem que solicita a um modelo de linguagem a criação de novas instruções a partir de exemplos fornecidos, gerando os dados necessários de forma semi-automática, sendo estes diversos, reais e apropriados. Esta prática possibilita a redução do esforço humano e uma maior escalabilidade do desempenho desses modelos.

Uma outra barreira existente para o treinamento de Grandes Modelos de Linguagem é o grande custo computacional e a necessidade de hardwares potentes e caros. Como forma de contornar este impedimento, a biblioteca PEFT [3] permite que o ajuste seja feito de forma menos custosa e eficiente a partir do uso de pelo menos uma das 4 técnicas disponíveis: Low-Rank adaptation (LoRA) [8], Prefix-Tuning [9], Prompt Tuning [10] e P-Tuning[11].

Entre essas quatro, pode-se destacar a LORA, na qual congela-se os pesos do modelo original e executa-se uma decomposição de matrizes baseada no parâmetro *rank*, um valor inteiro que determinará como será realizada tal decomposição, reduzindo a quantidade de parâmetros treináveis durante *fine-tuning*, mas mantendo a acurácia do modelo original.

A partir da possibilidade de gerar dados de forma semi-automática e da redução da complexidade do treino dos Grandes Modelos de Linguagem, a prática do *Instruction Tuning* é uma boa abordagem para construir modelos capazes de responder a instruções do usuário.

4. METODOLOGIA

A fim de construir um Grande Modelo de Linguagem capaz de responder a perguntas do contexto @ccc da Universidade Federal de Campina Grande, foi proposta uma abordagem com algumas etapas.

Para a construção dos dados foi realizada uma formulação semi-automática de perguntas e respostas sobre o contexto e seus temas.

Com o objetivo de construir o mecanismo que auxilia o modelo entregando o contexto sobre a pergunta realizada foi necessária a implementação da estrutura de busca semântica.

Por fim, foi implementado e executado o código de fine-tuning do modelo assim como a avaliação dos resultados gerados pelo ML ajustado.

4.1 Perguntas e Respostas Sobre o Contexto

As perguntas e respostas formuladas de forma semi-automática são classificadas em assuntos de dúvidas dos estudantes da graduação de Ciência da Computação. Devido a diferentes disposições das informações, foram adotadas abordagens distintas para cada tópico.

4.1.1 Disciplinas

A fase inicial de coleta de informações foi realizada com o Plano Pedagógico do Curso disponibilizado pela coordenação da graduação. Nele estão dispostos dados sobre todos os componentes curriculares do grupo de obrigatórias e optativas do curso, foram utilizados: Nome, Ementa, Bibliografia, Carga Horária e Pré-Requisitos.

Cada componente curricular pode ser nomeado de maneira informal pelos estudantes de diversas formas. A fim de solucionar algumas possíveis dúvidas sobre esses apelidos, foram colhidas informações do projeto Glossário UFCG da OpenDevUFCG [12] e com alguns alunos com o objetivo de construir um conjunto de perguntas e respostas sobre siglas, alcunhas e abreviações de cada disciplina da grade curricular. Os padrões de Pergunta e Resposta foram gerados com a API da *OpenAI*, utilizando o Modelo de Linguagem text-davinci-003, com 10 exemplos para pergunta e resposta.

De forma semelhante, foram formuladas automaticamente questões para cada unidade curricular sobre seus respectivos períodos, informações gerais, ementas e pré-requisitos, sempre seguindo a abordagem de gerar diversos modelos de perguntas e respostas, garantindo diversidade aos dados de treino.

Os dados deste tópico foram divididos nos seguintes subtópicos:

- Sigla de disciplina;
- Significado de sigla de disciplina;
- Período;
- Ementa;
- Pré-requisito;
- Período de disciplina;

4.1.2 Regulamento

Foram considerados apenas 3 subtópicos no tema de Regulamento: Regime de Exercício Domiciliar, Antecipação de Estudos e Reativação de Vínculos e todos esses tiveram abordagem semelhante durante a construção dos dados.

As informações destes procedimentos foram coletadas no *site* da graduação [13] e nos respectivos documentos de cada portaria. Com tais informações, a API da *OpenAI* foi, mais uma vez, a geradora de perguntas e respostas, porém, de uma maneira diferente do que foi realizado anteriormente. Para esse tema são passados trechos dos textos colhidos e são solicitadas perguntas e respostas sobre o conteúdo dos trechos enviados no prompt.

4.1.3 Atividades Complementares

Para a construção dos dados acerca de Atividades Complementares, foi utilizada uma abordagem envolvendo ambas as maneiras descritas nas duas seções anteriores. Iniciando com a coleta de informações no *site* e no documento da portaria que regulamenta esse tema, segue-se a abordagem de requisição à API da *OpenAI* com trechos dos textos com o objetivo de obter perguntas e respostas sobre o conteúdo enviado no prompt.

A segunda etapa para a formulação de perguntas e respostas é semelhante à abordagem realizada para o tema de Disciplinas. Para esta etapa, inicia-se a coleta de informações da planilha para consulta das atividades e suas respectivas horas [14]. Cada atividade, forma de comprovação e aproveitamento de carga

horária mínima e máxima foram extraídas e combinadas com as perguntas modelo geradas pela API da *OpenAI*. Sendo assim, todas as informações contidas na planilha foram abrangidas e fazem parte dos dados de treino.

Os dados deste tópico foram divididos nos seguintes subtópicos:

- Atividades complementares;
- Estágio;

4.1.4 Perfil do Estudante Egresso

Os dados do tópico sobre o perfil do estudante egresso foram construídos a partir de uma consulta a graduados do curso que trabalham na área de Ciência da Computação. Foram colhidos os cargos de 160 profissionais formados na graduação da UFCG. Após remover repetições e agregar alguns cargos semelhantes, foram definidos 20 cargos que são utilizados para construir os dados necessários para este tema.

Para cada cargo foi realizada uma consulta à API da *OpenAI* com o objetivo de elencar 7 habilidades técnicas necessárias para desempenhar o papel designado para aquela função. Por competência de determinada carreira, foi enviada uma nova requisição com o propósito de associar a habilidade e a ementa de cada disciplina. Com isso, foi possível construir sugestões de componentes curriculares que auxiliam na carreira de cada cargo definido.

Tópico	Subtópico	Quantidade
Disciplina	Período de Disciplina	2232
	Ementa	1058
	Disciplina	936
	Pré-requisito	840
	Sigla de Disciplina	428
	Significado de Sigla de Disciplina	276
	Período	144
Atividades Complementares	Atividades Complementares	696
	Estágio	14
Regulamento	Regulamento	112
Matrícula	Matrícula	72
Trilha de Disciplinas	Trilha de Disciplinas	220
Total		7028

Tabela 1 - Distribuição dos dados nos tópicos e subtópicos

4.2 Busca Semântica

Uma estrutura de Busca Semântica foi implementada para recuperar, caso exista, o contexto da pergunta fornecida pelo usuário. A comparação é realizada entre a entrada inserida e as perguntas já existentes que compõem os dados de treino do Modelo de Linguagem.

Primeiramente, se faz necessário transformar todas as informações textuais em embeddings, para isso, foi utilizado a biblioteca *SentenceTransformer* [15] possibilitando importar um Modelo de Linguagem do *HuggingFace*[2], repositório de modelos pré-treinados. Foi escolhido o Legal-BERTimbau-sts-base, um modelo baseado no BERT, ajustado para uma melhor compreensão da língua portuguesa. As entradas inseridas pelos usuários terão suas *features* extraídas de forma semelhante.

Com o objetivo de construir uma estrutura de Busca Semântica ágil, foi utilizada a biblioteca FAISS [16]. Ela é responsável por criar índices planos para cada *sentence embedding* e fazer o cálculo da distância L2 de forma exata, exaustiva e otimizada entre o vetor de entrada e todos os vetores existentes, retornando os 5 resultados com a menor distância. Uma vez criados, os índices podem ser salvos em um arquivo, sem a necessidade da construção total novamente, caso seja preciso atualizar os vetores.

Foi definido um limiar máximo do valor da distância L2, com o objetivo de evitar o envio de contextos errados para o Modelo de Linguagem. Para a definição desse limiar, primeiramente foram calculadas as distâncias par-a-par de cada vetor com todos os outros vetores da base e inseridas em uma lista ordenada. Após isso, itera-se sobre a estrutura de dados até uma sequência de 5 elementos consecutivos que não pertencem à mesma categoria do item atual, pode-se considerar que resultados após essa sequência não fazem parte do contexto correto. O limiar corresponde a média dessas distâncias encontradas, caso não tenha um retorno abaixo, a pergunta será feita sem o contexto adicional.

Adicionalmente ao uso do limiar, os resultados retornados passam por uma etapa que obtém o ROUGE-1 entre a entrada e cada texto recuperado. Esta métrica mede a sobreposição de palavras de duas sentenças, comparando termo a termo e obtendo a precisão, revocação e *f1-score*, sendo os valores deste último utilizados para escolher o melhor contexto a ser enviado para o Modelo de Linguagem.

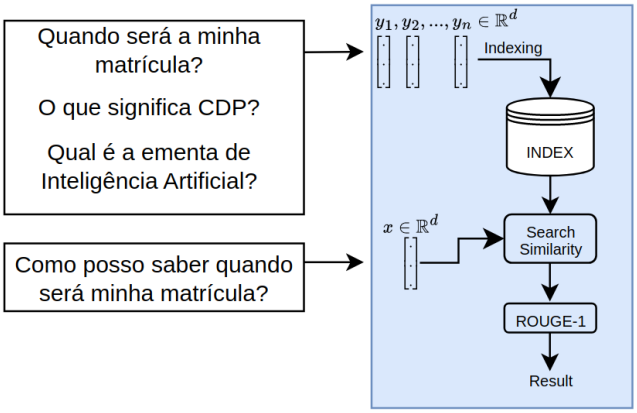

Figura 4 - Pipeline da Busca Semântica

4.3 Fine-Tuning

A tarefa de Pergunta e Resposta em um contexto específico, quando realizada por um Grande Modelo de Linguagem, pode não

[2] https://huggingface.co/

desempenhar um bom resultado devido à falta de conhecimento daquele domínio mais particular. Com o objetivo de contornar esse problema e alcançar uma boa performance, foi realizado o *fine-tuning* de um GML, mais especificamente um *instruct-tuning*, com o objetivo de ajustar o modelo a responder instruções do contexto da graduação de Ciência da Computação na Universidade Federal de Campina Grande.

1.1.1 O Modelo de Linguagem

Foi escolhido o LLaMA-13B como o modelo de linguagem a ser treinado e avaliado. São dois os principais motivos desta escolha: primeiramente, o fato dele ser *open source* e o MetaAI ter disponibilizado os pesos de forma pública para fins científicos possibilitou a execução desta pesquisa. Outro motivo é o desempenho do modelo que, com uma dimensão bem menor que outros, têm valores de performance semelhantes a abordagens com 10 vezes mais parâmetros. Devido a essa redução na quantidade de parâmetros, é permitido executar treino e inferências do modelo LLaMA com apenas uma placa de vídeo.

1.1.2 Ajuste do Modelo

A grande maioria de trabalhos atuais na área de Modelos de Linguagem utilizam funções, classes e modelos pré-treinados do *HuggingFace*. É nesta plataforma que estão disponíveis os arquivos com os pesos do GML, funções de pré-processamento e classes que executam o treino.

Durante o processo de *fine-tuning* de um modelo de linguagem é necessário que diversos cálculos sejam realizados com uma alta precisão, este fator pode acarretar em um treino que demande muito tempo. Com o objetivo de reduzir a duração total deste processo, a biblioteca *PyTorch* [17] é utilizada, sendo responsável por manipular tensores e matrizes e executar cálculos de forma paralela na GPU aumentando a velocidade de treino.

Os dados inseridos para o *instruction-tuning* são *prompts* no formato de instruções em Português que simulam futuras entradas que o modelo irá receber para gerar a resposta. Essas instruções contém as perguntas, contextos, para as perguntas que necessitam, e respostas que foram construídas conforme foi relatado na seção 4.1.

Para redução da complexidade do treino foi utilizada a abordagem LORA da biblioteca PEFT. O parâmetro *rank* com valor igual a 16 e *lora_alpha* igual a 32 permitem que o modelo realize o *fine-tuning* sem perder as informações do seu treino primário.

A função responsável por executar o treino faz parte da biblioteca *HuggingFace*. Com o módulo de treino da classe *Trainer*, apenas é necessário passar os atributos referentes aos dados de treino e teste, parâmetros numéricos como: taxa de aprendizado, *batch_size*, máximo de execuções, além das configurações do LORA. Com isso, o *fine-tuning* é executado, com cálculo de perda e de taxa de aprendizado.

1.1.1 Hardware

O experimento foi executado utilizando apenas um computador com um processador Intel Core i7-12700K, 128GB de Memória RAM e uma placa de vídeo NVIDIA GeForce RTX 3090.

4.4 Avaliação de Resultados

Para avaliar as respostas geradas, foram utilizadas duas abordagens: a métrica Bilingual Evaluation Understudy (BLEU) e a avaliação humana. As avaliações foram realizadas para 3 cenários diferentes:

1. Modelo de linguagem base, sem nenhum treino;
2. Modelo ajustado com os dados do contexto, sem auxílio da busca semântica;
3. Modelo ajustado com os dados do contexto, com o auxílio da busca semântica para fornecer mais informações sobre o tema da pergunta;

BLEU é uma métrica de avaliação automática geralmente usada para medir a qualidade de traduções geradas por modelos de tradução automática. Seu objetivo é fornecer uma pontuação numérica que correlacione a frase gerada por um modelo de linguagem e a frase de referência. Esta métrica utiliza contagem de n-gramas e suas ocorrências, precisão e também pondera os valores levando em consideração os tamanhos das sentenças. A pontuação BLEU varia de 0 a 1, e sua interpretação pode ser a seguinte:

- BLEU próximo de 0: Indica uma correspondência mínima ou nenhuma correspondência entre a tradução gerada e as traduções de referência;
- BLEU em torno de 0.3-0.4: Indica uma correspondência parcial ou baixa qualidade da tradução gerada;
- BLEU em torno de 0.5-0.6: Indica uma qualidade moderada da tradução gerada, mas com espaço para melhorias.
- BLEU acima de 0.6: Indica uma correspondência razoável ou alta qualidade da tradução gerada em relação às traduções de referência.

Para a avaliação humana participaram 5 alunos da graduação que avaliaram as respostas para os 3 cenários distintos. Cada texto gerado pelo modelo de linguagem foi julgado com uma das 3 seguintes classificações:

- 0: Resposta errada, não resolve a dúvida da pergunta inserida.
- 1: Resposta parcialmente correta, não responde completamente a pergunta inserida.
- 2: Resposta correta, responde por completo a pergunta inserida.

As perguntas que foram utilizadas para realizar a avaliação foram escritas por alunos da graduação através de um formulário Google.

5. RESULTADOS

O *instruction-tuning,* ajuste baseado em instruções, foi concluído em 3 horas e 5 minutos. Foram selecionadas 60 perguntas e respostas, 5 de cada subtópico, para serem avaliadas utilizando BLEU e também a avaliação humana.

Com a métrica BLEU, o modelo ajustado com o auxílio da busca semântica obteve o valor de: 0,30. A tabela 2 abaixo indica os valores para os 3 cenários de avaliação.

Modelo	Busca Semântica	BLEU
LLaMA-13B - Base	Não	0,17
LLaMA-13B - Ajustado	Não	0,22
LLaMA-13B - Ajustado	Sim	0,30

Tabela 2 - Métrica BLEU para cada cenário

Os valores obtidos na métrica BLEU mostram uma melhoria entre o modelo LLaMA sem ajuste e modelo de linguagem ajustado. Quando adicionado a busca semântica, fornecendo um contexto a mais ao modelo, pode-se perceber que o ganho é de cerca de 30% no valor, justificando o uso da estrutura de busca semântica em conjunto com o GML para responder às perguntas.

Durante a avaliação humana participaram 5 estudantes da graduação, que julgaram as perguntas com as classificações de 0 a 2. A tabela 3 indica a média dos valores atribuídos para cada modelo.

Modelo	Busca Semântica	Média da Classificação
LLaMA-13B - Base	Não	0,10
LLaMA-13B - Ajustado	Não	0,38
LLaMA-13B - Ajustado	Sim	1,05

Tabela 3 - Classificação humana para cada cenário

De acordo com os avaliadores, a média de respostas totalmente corretas, classificadas como 2, foi 26, mostrando que um valor próximo a metade do total de perguntas realizadas na validação foram respondidas de forma totalmente correta. A tabela 4 mostra a média da quantidade de erros e acertos para cada cenário de avaliação.

	LLaMA-13B - Base	LLaMA-13B - Ajustado	LLaMA-13B - Ajustado com Busca
0	55,0	43,8	24,6
1	4,7	10,4	8,6
2	0,3	5,8	26,8

Tabela 4 - Classificação absoluta para cada cenário de avaliação

Observando os valores da avaliação humana torna-se ainda mais perceptível a diferença da qualidade das respostas entre o modelo não ajustado e o modelo ajustado no contexto @ccc. Ainda assim, as respostas geradas pelo modelo sem o auxílio da estrutura de busca semântica prova que é necessário a utilização de um mecanismo capaz de fornecer mais informações sobre o tema da pergunta realizada para cenários mais específicos.

6. CONCLUSÃO E TRABALHOS FUTUROS

A partir dos resultados obtidos desta pesquisa, podemos observar que a prática de *instruction-tuned* é capaz de preparar Grandes Modelos de Linguagem para responder a instruções de contextos mais específicos, sem a necessidade de um grande custo computacional e com a geração de dados de forma semi-automática.

Abordagens que reduzem a complexidade do modelo e consequentemente custos de tempo e de *hardware* não afetam os resultados das gerações textuais, possibilitando que essa prática seja amplamente utilizada para diversas necessidades.

Associar informações recuperadas durante a busca semântica para realizar perguntas a um Grande Modelo de Linguagem acarreta em uma melhoria na eficácia das respostas, garantindo que o ML capture mais informações sobre o contexto do que se trata a pergunta e possíveis elementos que possam contribuir a fim de obter o melhor resultado.

Como trabalhos futuros pode-se considerar uma coleta de perguntas exaustiva, abrangendo todos os alunos da graduação. Com mais dados pode-se construir novas instruções, abrangendo mais tópicos e subtópicos do contexto @CCC. Também pretende-se aprimorar dados sobre o perfil do egresso do curso, levantando mais informações sobre os componentes curriculares e suas áreas de atuação.

Por fim, há a possibilidade da construção de um projeto piloto de uma API *Web* com um *chatbot* com a participação de todos os alunos da graduação, tendo em vista aumentar significativamente a quantidade de informações coletadas e melhorar ainda mais a acurácia do modelo de linguagem treinado, permitindo a existência de um mecanismo que seja capaz de resolver dúvidas de diversos temas do contexto de Ciência da Computação na Universidade Federal de Campina Grande.

REFERÊNCIAS

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, e Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL].

[2] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2022). Self-Instruct: Aligning Language Model with Self Generated Instructions. arXiv:2212.10560 [cs.CL].

[3] HuggingFace - PEFT Documentation. Disponível em: https://huggingface.co/docs/peft/index. Acesso em 8 de Junho de 2023.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. (2019). Attention Is All You Need. arXiv:1706.03762 [cs.CL].

[5] Le, Q. V., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. arXiv:1405.4053 [cs.CL].

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional

Transformers for Language Understanding. arXiv:1810.04805 [cs.CL].

[7] Renrui Zhang, Jiaming Hanm, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, Yu Qiao. (2023). LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. arXiv:2303.16199 [cs.CV].

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL].

[9] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, Jie Tang. (2022). P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. arXiv:2110.07602 [cs.CL]

[10] Brian Lester, Rami Al-Rfou, Noah Constant. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 [cs.CL]

[11] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang. (2021). GPT Understands, Too. arXiv:2103.10385 [cs.CL]

[12] Glossário UFCG. Disponível em: https://github.com/OpenDevUFCG/glossario-ufcg. Acesso em 9 de Junho de 2023.

[13] Computação UFCG - Regulamentação. Disponível em : https://www.computacao.ufcg.edu.br/graduacao/procedimentos-gradua%C3%A7%C3%A3o/regulamenta%C3%A7%C3%A3o. Acesso em 9 de Junho de 2023.

[14] Computação UFCG - Aproveitamento de Atividades Complementares. Disponível em: https://www.computacao.ufcg.edu.br/graduacao/procedimentos-gradua%C3%A7%C3%A3o/aproveitamento-de-atividades-complementares. Acesso em 9 de Junho de 2023.

[15] SBERT.net - Sentence Transformers Documentation. Disponível em: https://www.sbert.net/. Acesso em 9 de Junho de 2023.

[16] faiss.ai - Faiss Documentation. Disponível em: https://faiss.ai/. Acesso em 9 de Junho de 2023.

[17] pythorch - PyTorch. https://pytorch.org/. Acesso em 9 de Junho de 2023.