UTILIZANDO BERT PARA A TAREFA DE PERGUNTA-RESPOSTA EM LÍNGUA PORTUGUESA

GUILHERME RAMISON[1]; LARISSA ASTROGILDO DE FREITAS[2]

[1]*Universidade Federal de Pelotas – gramison@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – larissa@inf.ufpel.edu.br*

1. INTRODUÇÃO

Com o constante avanço da tecnologia e um mercado cada vez mais automatizado, soluções envolvendo o Processamento da Língua Natural (PLN), sejam elas chatbots, atendentes virtuais ou outros softwares, estão ficando cada vez mais evidentes no nosso dia-a-dia. Segundo Gattis (2020), o uso de tecnologias envolvendo PLN cresceu 100% em torno do globo, tendo em vista a atual situação de isolamento social causado pelo COVID-19.

Existem vários segmentos envolvendo PLN, o que iremos abordar neste trabalho é o de Pergunta-Resposta (*Question-Answer*). Dado um contexto, essa tarefa auxilia na localização de respostas de acordo com uma pergunta feita ao modelo. Sendo assim, utilizaremos a arquitetura *Transformers*, mais especificamente o modelo BERT, um modelo de Aprendizado de Máquina (AM) que se destacou em 2018 por obter o melhor *score* na tarefa de *Question-Answer* (Google AI Blog, 2018).

2. METODOLOGIA

O BERT faz uso do *Transformer*, um mecanismo de atenção que aprende relações contextuais entre palavras (ou subpalavras) em um texto. Em sua forma original, o *Transformer* inclui dois mecanismos separados - um codificador que lê a entrada de texto e um decodificador que produz uma previsão para a tarefa. Uma vez que o objetivo do BERT é gerar um modelo de linguagem, apenas o mecanismo do codificador é necessário (HOREV, 2018).

Ao contrário dos modelos direcionais, que lêem a entrada de texto sequencialmente (da esquerda para a direita ou da direita para a esquerda), o codificador *Transformer* lê toda a sequência de palavras de uma vez. Portanto, é considerado bidirecional, embora seja mais preciso dizer que é não direcional. Esta característica permite que o modelo aprenda o contexto de uma palavra com base em todos os seus arredores (esquerdo e direito da palavra) (HOREV, 2018).

Com o BERT, podemos pré-treinar um modelo, processo que requer muitos dados na linguagem alvo, como artigos, notícias, livros, etc.

Também temos a opção de usar um modelo pré-treinado e fazer um *fine-tuning* para uma tarefa específica, usando nosso próprio *dataset* no processo. O *fine-tuning* visa melhorar os resultados obtidos pelo modelo na tarefa de interesse, no nosso caso, *Question-Answer*.

A sequência de entrada do *fine-tuning* para tarefa de Pergunta-Resposta tem duas partes: a Pergunta (tokenizada), seguida por um token especial [SEP] e o Contexto (tokenizado) (Figura 1). Para mais informações sobre o funcionamento do processo, veja Google AI Language (2019).

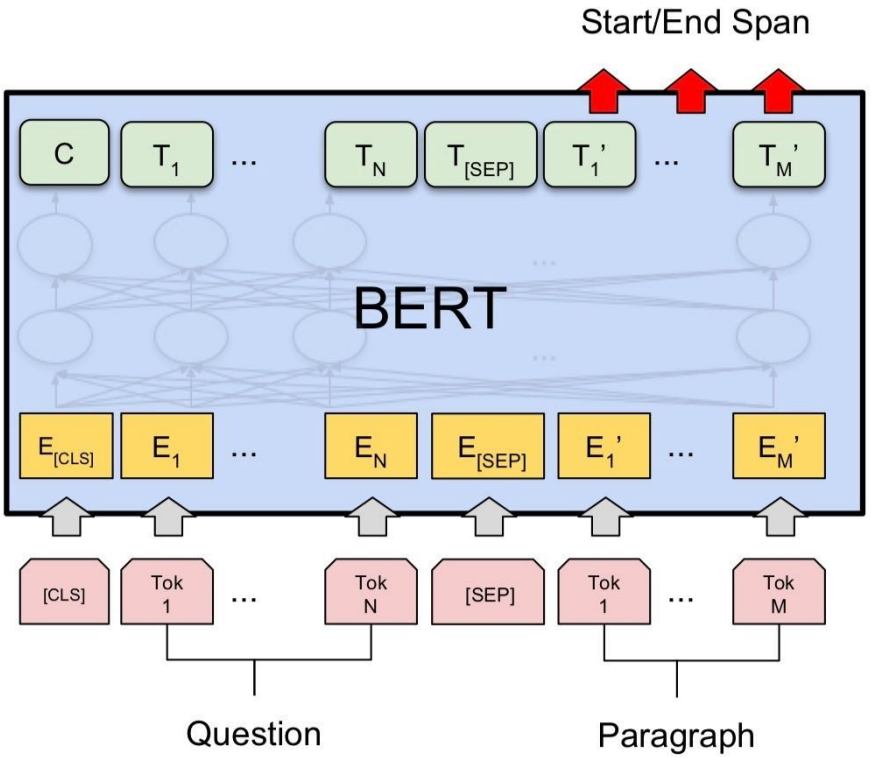

Figura 1. Arquitetura do *fine-tuning* para *Question-Answer*

O Google disponibiliza vários modelos pré-treinados do BERT. Neste trabalho vamos utilizar o BERT Multilingual (Google Research, 2018), o qual suporta 104 linguagens, dentre elas o português (https://github.com/google-research/bert).

Porém, para fazer o *fine-tuning*, nós precisamos juntar dados para o input do modelo. O formato dos dados mais utilizado para realizar a tarefa de Pergunta-Resposta é do SQuAD (https://rajpurkar.github.io/SQuAD-explorer/), que consiste em dar um contexto, uma lista de perguntas relacionadas ao contexto e suas respectivas respostas. O SQuAD versão 1.1 contém um conjunto com mais de 100 mil Perguntas-Respostas em inglês.

Dentre os *datasets* de Perguntas-Respostas disponíveis em língua portuguesa podemos citar: uma versão do SQuAD versão 1.1 com 60 mil Perguntas-Respostas traduzidas pela API do Google Translate (https://cloud.google.com/translate/?hl=pt-BR), uma versão do SQuAD versão 1.1 traduzida e revisada pela AI Lab (UnB, 2020) e partes da coleção CHAVE com 172 novos pares de Perguntas-Respostas. O CHAVE consiste nas edições completas dos anos de 1994 e 1995 dos jornais PÚBLICO (www.publico.pt) e Folha de São Paulo (www.folha.com.br), que foi compilada pela Linguateca (www.linguateca.pt) no quadro do CLEF (www.clef-campaign.org).

Como o dataset do CHAVE estava fora do formato SQuAD e tinha algumas informações que não podiam ser transformadas para o padrão, rodamos um *script* Python para colocar os dados no formato SQuAD.

Com os *datasets* padronizados, começamos juntando a versão do SQuAD com 60 mil Perguntas-Respostas com o CHAVE, em seguida rodamos o *script* para fazer o *fine-tuning* em uma TPU por meio do Google Colab. A execução demorou por volta de 20 minutos, um tempo muito pequeno em comparação com o treinamento de um modelo que pode durar horas.

Com a versão traduzida completa do SQuAD, repetimos o processo de *fine-tuning* no Google Colab. Desta vez, o tempo de execução foi em torno de 30 minutos, aumento causado pela diferença do tamanho entre os *datasets*.

Em ambos os processos foi observado falta ou perda de *matches* por conta da caixa das letras, pois o BERT leva em consideração se a letra está maiúscula ou minúscula. Por conta disso, pretendemos repetir os processos com os *datasets* formatados em minúsculo para uma possível melhora nos resultados.

3. RESULTADOS E DISCUSSÃO

Para fins de avaliação dos modelos, utilizaremos as seguintes medidas: Exact Match (EM) e o F1. Para encontrarmos essas medidas, precisamos comparar as predições feitas pelo modelo com um *dataset* de teste, onde se encontram perguntas, contextos e respostas diferentes do *dataset* usado para fazer o *fine-tuning*, e fazer uma matriz de confusão (Figura 2).

		Valor Predito	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Figura 2. Exemplo de matriz de confusão (Medium, 2019)

O F1 nos mostra o balanço entre a precisão e o *recall* do nosso modelo, calculado por 2 * (precisão * *recall*) / (precisão + *recall*), sendo a precisão: TP / (TP + FP) e sendo o recall: TP / (TP + FN).

O EM representa a proporção de respostas que bateram exatamente entre o conjunto de teste e as predições. Lembrando que um match pode variar entre 0 (*No Match*) e 1 (*Exact Match*).

Por meio do Google Colab, realizamos o *fine-tuning* com os dois *datasets* separadamente, obtendo os seguintes resultados:

Datasets	EM	F1
SQuADv1 60mil + CHAVE	50,73	68,98
SQuADv1 completo	57,58	74,16

Figura 3. Tabela dos resultados obtidos no *fine-tuning*

Como mencionado anteriormente, esperamos que se fizermos o *fine-tuning* com os contextos, perguntas e respostas formatados com as letras minúsculas, os resultados possam melhorar por causa da maior quantidade de matches entre a predição e o *dataset* de teste.

Comparando os resultados com modelos treinados em inglês, temos uma diferença considerável: a F1 do modelo *BERT Inglês base* (Google Research, 2018) é de 88,4. Já em modelos ajustados (BERT Multilingual) para outras línguas, como a Russa, a F1 é 83,09 e a Chinesa, a F1 é 85,00 (DeepPavlov, 2019).

4. CONCLUSÕES

A arquitetura BERT é consideravelmente fácil de usar, sendo bem documentada e tendo vários exemplos de uso na internet, deixando ainda mais acessível à comunidade de interesse.

A próxima etapa deste trabalho é utilizar o modelo pré-treinado para o português do BERT, BERTimbau (NeuralMind, 2020). Utilizando esse modelo, esperamos ter uma melhora nos resultados obtidos na tarefa de Pergunta-Resposta, por conta do modelo ser pré-treinado especialmente para o português.

5. REFERÊNCIAS BIBLIOGRÁFICAS

GATTIS, N. **Olhar Digital**. Acessado em 23 set. 2020. Disponível em: https://olhardigital.com.br/coronavirus/noticia/coronavirus-atendimentos-via-chatbots-aumentam-100-no-mundo/98554

Medium. **Métricas de Avaliação: acurácia, precisão, recall… quais as diferenças?**. Acessado em 27 set. 2020. Disponível em: https://medium.com/@vitorborbarodrigues/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c

Google Research. **BERT**. Acessado em 24 set. 2020. Disponível em: https://github.com/google-research/bert

HOREV, R. **BERT Explained: State of the art language model for NLP.** Acessado em 23 set. 2020. Disponível em: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

NeuralMind. **BERTimbau**. Acessado em 25 set. 2020. Disponível em: https://github.com/neuralmind-ai/portuguese-bert

Google AI Language. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.** Acessado em 25 set. 2020. Disponível em: https://arxiv.org/pdf/1810.04805.pdf

UnB. **Datasets em Português**. Acessado em 23 set. 2020. Disponível em: https://forum.ailab.unb.br/t/datasets-em-portugues/251