Nome: Rafael Claro Ito (R.A.: 118430)
Resumo do artigo: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Neste artigo é apresentado um novo modelo de representação de linguagem chamado BERT, criado pela Google. Sua principal característica em relação a outros modelos é que ele usa os contextos a esquerda e a direita conjuntamente durante o pré-treino. Enquanto o BERT usa *Transformer* bidirecional, outros modelos como o GPT da OpenAI usa também *Transformer* mas com uma abordagem unidirecional *left-to-right* e o modelo ELMo usa uma concatenação de LSTMs treinadas em contextos *left-to-right* e *right-to-left* independentes.

Existem duas estratégias para se usar uma representação de linguagem pré-treinada em tarefas: ***feature-based*** e ***fine-tuning***. Na abordagem *feature-based* as representações pré-treinadas são usadas como *features* adicionais em uma arquitetura específica para a tarefa em questão. Já na abordagem *fine-tuning*, evita-se parâmetros específicos e ajusta-se todos os parâmetros para cada tarefa em questão. Para o BERT são exploradas abordagens baseadas em *fine-tuning*, embora o resultado da seção 5.3 demonstre que este modelo performa bem em ambas abordagens.

A **arquitetura** do BERT é um ***Transformer*** *encoder* **bidirecional** de multi-camadas. Neste artigo são usados dois modelos: $\textbf{BERT}_{\textbf{BASE}}$ (12 blocos *Transformer*, tamanho da camada escondida H = 768, 12 cabeças de auto-atenção, 110M parâmetros) e $\textbf{BERT}_{\textbf{LARGE}}$ (24 blocos *Transformer*, tamanho da camada escondida H = 1024, 16 cabeças de auto-atenção, 340M parâmetros).

A arquitetura de entrada do modelo é de tal maneira a suportar tarefas que exigem uma sequência ou um par de sequências, sendo que neste último caso o *token* **[SEP]** é usado para separar as sequências. O primeiro *token* de entrada é sempre o **[CLS]**, sendo que o estado escondido correspondente é usado para tarefas de classificação. A entrada do modelo é composta pela soma de três ***embeddings***: *token*, posicional e de segmento. A única novidade aqui é o último, que corresponde ao *embedding* de sequência A ou B para tarefas de pares de sequências.

Uma característica relevante do BERT é a arquitetura unificada entre diversas tarefas. Pouco se altera entre a arquitetura de pré-treino e a arquitetura final usada nas tarefas. Outro ponto a se ressaltar é que a arquitetura do $\text{BERT}_{\text{BASE}}$ é quase idêntica ao OpenAI GPT, se diferenciando pelo fato de levar em conta a bidirecionalidade de contexto.

O **treinamento** é dividido em duas etapas. A primeira delas é a fase ***pre-training***. Aqui utiliza-se treinamento não-supervisionado em duas tarefas: MLM (*Masked LM*) e NSP (*Next Sentence Prediction*).

- **MLM**: Nesta tarefa alguns *tokens* são mascarados e o objetivo do modelo é prever o *token* mascarado. Aqui, são escolhidos 15% dos *tokens* de cada sequência randomicamente. Em 80% das vezes esse *token* é substituído pelo *token* **[MASK]**, 10% é substituído por um *token* qualquer e 10% o *token* original é mantido. Isso é feito para reduzir o descasamento entre as etapas de pré-treinamento e *fine-tuning*, visto que o *token* [MASK] nunca será visto no *fine-tuning*.
- **NSP**: Para algumas tarefas, como QA e NLI, faz-se necessário o entendimento da relação entre duas sequências. Como tal característica não é obtida diretamente através da modelagem da linguagem, treina-se aqui a predição de uma sequência B ser ou não a próxima da sequência A. Para o treino, 50% das vezes é utilizada a próxima sequência (label “IsNext”) e 50% uma sequência aleatória (label “NotNext”).

Por fim, o pré-treino é realizado nos corpus BookCorpus (800M palavras) e Wikipedia em inglês (2,500M palavras).

Seguido a etapa de pré-treinamento, tem a fase de ***fine-tuning***. As tarefas selecionadas para o *fine-tuning* são as tarefas do **GLUE** (o dataset WNLI foi excluído devido a um problema de construção), **SQuAD v1.1, SQuAD v2.0** e **SWAG**. Aqui o modelo é inicializado com os parâmetros do pré-treinamento, que são ajustados de acordo com o aprendizado supervisionado da tarefa específica em questão. Na saída, as representações dos *tokens* são usadas como entradas de uma camada final para tarefas a nível de palavras e o *token* [CLS] é usado também como entrada de uma camada final (K x H, onde K é o número de classes) para tarefas de classificação. Comparado ao pré-treinamento essa etapa é computacionalmente barata. Foram utilizados os seguintes **parâmetros de treinamento** para o *fine-tuning*: **GLUE** - 3 épocas, *learning rate* de melhor entre 5e-5/4e-5/3e-5/2e-5, *batch size* de 32; **SQuAD v1.1** - 3 épocas, *learning rate* de 5e-5, *batch size* de 32; **SQuAD v2.0** - 2 épocas, *learning rate* de 5e-5, *batch size* de 48; **SWAG** - 3 épocas, *learning rate* de 2e-5, *batch size* de 16.

Com o BERT obteve-se **resultados** *state-of-the-art* em 11 tarefas de NLP: **MNLI**, **QQP**, **QNLI**, **SST-2**, **CoLA**, **STS-B**, **MRPC**, **RTE**, **SQuAD v1.1**, **SQuAD v2.0** e **SWAG.** Na seção 5 do artigo são mostrados resultados de diversas variações do modelo a fim de se entender a contribuição e relevância de cada escolha. São realizados experimentos relacionados às tarefas de pré-treino, tamanho do modelo e aplicação do BERT baseados em *features* (*features-based*).

Podemos traçar um paralelo e fazer uma comparação de uso do BERT para tarefas de NLP com o ***transfer learning*** usado principalmente em visão computacional. Assim como é possível usar redes vencedoras de competições tipo ImageNet como extratores de *features* de imagens e adicionar apenas uma última camada (treinando-a individualmente ou fazendo *fine-tuning*), é possível usar o BERT e apenas uma única camada adicional de saída para empregá-lo em diversas tarefas de processamento de linguagem, como por exemplo as tarefas do GLUE. Isso faz com que o número de parâmetros a serem aprendidos do zero seja drasticamente reduzido além de incorporar uma representação da linguagem que foi aprendida com muitos dados durante a fase de pré-treinamento do BERT.