MAC5725 – Linguística Computacional – 2020

EP 3: Geração Automática de Título para um Texto

Entrega: 14/12/2020

1 Motivação

Imagine que você tem um site onde são depositados avaliações de produtos, e o seu papel é gerar e propor um título informativo para a avaliação que acaba de ser submetida. Esta é a situação motivação para este exercício programa. No caso, vamos gerar títulos para as avaliações da B2W, os quais devem ser aprendidos a partir dos títulos fornecidos para as avaliações existentes no nosso córpus. No caso, vamos usar a coluna `review_text` como entrada e a coluna `review_title` como saída. Por exemplo, uma avaliação do córpus da B2W:

> **Entrada**: "MEU FILHO AMOU! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM!"
>
> **Saída**: "presente mais que desejado"

Neste exercício devemos comparar dois métodos de geração de texto. Primeiramente, vamos usar uma arquitetura encoder-decoder, baseada em rede recorrente do tipo LSTM/GRU, com bidirecionalidade opcional, e com uma camada de atenção.

Em seguida vamos gerar uma adaptação das redes tipo BERT para gerar textos curtos a partir de uma semente. No caso, a semente será a avaliação e o texto gerado será o título sugerido para esta avaliação.

Vamos também medir a qualidade da geração de texto e, para isso, vamos usar diversas métricas propostas para a atividade de tradução de textos, muito embora tanto a linguagem de origem quanto à linguagem de destino no nosso caso sejam as mesmas, e a nossa tarefa está mais próxima de uma sumarização do que de uma tradução. Note que neste caso a qualidade dos dados de treinamento é fundamental, pois nunca conseguiremos gerar títulos melhores do que os títulos fornecidos pelos próprios usuários para suas avaliações existentes no córpus.

2 Encoder-Decoder BiLSTM

O primeiro passo do trabalho consiste na implementação de uma arquitetura encoder-decoder (sequence-to-sequence) usando uma rede bidirecional LSTM (ou GRU, a escolha) como Encoder, uma rede LSTM (ou GRU) como Decoder, e uma camada de atenção relacionando os estados intermediários da entrada com a saída. O número de camadas da sua rede fica a sua escolha.

Esta arquitetura funciona de forma similar aos modelos de tradução que já conhecemos, onde a entrada consistirá no texto da avaliação do produto e a saída em um título expressivo para esta avaliação, ambos em português.

Você tem a opção de limitar o vocabulário de saída ao conjunto de palavras existentes nos títulos do córpus da B2W, mas isso não é obrigatório e nem seria usado em um sistema de recomendação de títulos em produção. Abrimos essa opção apenas para simplificar o EP caso necessário.

Você deve desenvolver a sua rede, prestando atenção no fenômeno de sobreajuste (overfitting), dividindo o Corpus em treinamento, validação e teste, e realizar as medidas finais da qualidade da tradução apenas sobre o córpus de teste.

3 Gerando um Título Usando BERT

A arquitetura de redes neurais BERT, assim como suas derivadas na BERTologia, é baseada na parte Encoder de um arquitetura Transformer. Como esta arquitetura não possui um componente decoder, ela pode ser usada para classificar sentenças inteiras ou partes de uma sentença, mas não tem embutida um gerador de texto como um módulo decoder. Sendo assim, é necessário adaptar arquitetura BERT para que ela seja capaz de gerar textos.

Neste exercício, vamos sugerir a utilização de um método auto-regressivo para a geração de texto. Esta sugestão não é obrigatória, você poderá usar qualquer outro método que quiser para adaptar o BERT para esta tarefa de geração de texto, ou até mesmo propor uma alteração deste método.

A característica deste método auto-regressivo é que iremos passar por várias iterações. A cada iteração, vamos gerar uma palavra da saída, a qual será incorporada a entrada na próxima iteração. Desta forma, vamos gerar a frase de saída da esquerda para a direita, uma palavra de cada vez, e o método de geração de palavras é uma iteração de vários passos em que o BERT é ativado uma vez por iteração.

Lembramos que o treinamento da arquitetura BERT possui duas fases:

- O treinamento auto-supervisionado bidirecional. Nesta etapa, que não iremos refazer, as palavras são recuperadas em qualquer posição de uma sentença, usando a informação anterior e posterior a marca de palavra mascarada, [MASK]. Ou seja, nesta fase, há uma aparente

incompatibilidade entre a geração da saída da esquerda para direita com a natureza direcional do treinamento auto-supervisionado.

- O treinamento por refinamento de uma tarefa de classificação. No nosso caso, vamos realizar essa etapa com uma classificação da sentença de entrada onde as possíveis classes serão as palavras de nosso vocabulário. Ou seja, a tarefa de classificação se confunde com a tarefa de geração de uma palavra. É o refinamento que irá ensinar o BERT a gerar as palavras da saída da esquerda para direita.

Durante o treinamento, cada frase do corpus irá gerar $n + 1$ exemplos de treinamento, onde n é o número de palavras presentes à saída. Vamos ilustrar este fato usando o exemplo mencionado anteriormente.

Entrada: "MEU FILHO AMOU! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM!"

Saída: "presente mais que desejado"

Este exemplo com 4 palavras na saída gera 5 frases de classificação:

Entrada: "[CLS] MEU FILHO AMOU ! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM ! [SEP] [MASK] [SEP]"

Classe: "presente"

Entrada: "[CLS] MEU FILHO AMOU ! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM ! [SEP] presente [MASK] [SEP]"

Classe: "mais"

Entrada: "[CLS] MEU FILHO AMOU ! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM ! [SEP] presente mais [MASK] [SEP]"

Classe: "que"

Entrada: "[CLS] MEU FILHO AMOU ! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM ! [SEP] presente mais que [MASK] [SEP]"

Classe: "desejado "

Entrada: "[CLS] MEU FILHO AMOU ! PARECE DE VERDADE COM TANTOS DETALHES QUE TÊM ! [SEP] presente mais que desejado [MASK] [SEP]"

Classe: "<EOS>"

Em cada caso, estamos fornecendo duas frases para a rede: "[CLS] frase 1 [SEP] frase 2 [SEP]". A primeira frase nunca se altera, e sempre contém o texto da avaliação. A segunda frase contém uma indicação de qual palavra deve ser gerada, indicada pelo rótulo [MASK]. Inicialmente a frase contém apenas esse rótulo. No último passo, estamos impondo a geração de um rótulo de fim de sentença, que aliás pode ser o próprio rótulo [SEP]. Desta forma "enganamos" o sistema, que acaba aprendendo a gerar as palavras da esquerda para direita. Note que o sistema pode aprender essa classificação mesmo que as sentenças de treinamentos sejam embaralhadas com outras sentenças do córpus de treinamento.

Teoricamente o vocabulário de saída é igual vocabulário total que estamos lidando. Por simplificação podemos assumir um vocabulário muito menor, construído a partir das palavras no campo de `review_title` do córpus. A classe de saída pode ser codificada como um vetor 1-hot dos elementos do vocabulário, e a saída da camada de classificação pode ser uma distribuição softmax sobre o vocabulário, de tal forma que a palavra que receber maior massa de probabilidade acaba sendo a palavra escolhida como classe de saída.

Na fase de execução, o processo de geração das palavras deve ser iterado. Inicialmente, a entrada será "[CLS] frase 1 [SEP] [MASK] [SEP]", e o seu programa deve prosseguir até gerar o carácter de final da sentença, ou então impor um limite no número de palavras do título, por exemplo, o título nunca deve ter mais do que 20 palavras.

4 Métricas da Qualidade do Texto Gerado

Existem várias formas de medir a qualidade da geração de um texto. Inicialmente, como estamos simulando o processo de geração do título como um processo de classificação, podemos tomar a acurácia média da classificação como uma medida de qualidade.

No entanto, essa não é uma medida muito justa. Por exemplo, se a resposta for " presente mais que desejado", e em vez disso o nosso programa gerar como título "um presente mais que desejado", a acurácia média será de zero, pois inseriu uma palavra nova logo no início e perdeu totalmente a sincronização, mas a impressão é de que este texto mereceria acurácia 100%.

Por esses motivos, a tarefa de tradução de texto já propõe diversas outras medidas de qualidade que devemos analisar nesse trabalho. Estas medidas levam em consideração trechos e palavras gerados, e comparam com uma ou mais possíveis alternativas consideradas como "referências".

Uma coisa que pode ser feita é olhar para cada palavra na frase de saída e atribuir a ela uma pontuação de 1 se ela aparecer em qualquer uma das frases de referência (no nosso caso sempre teremos apenas uma frase de referência)

e 0 se não aparecer. Então, para normalizar essa contagem para que fique sempre entre 0 e 1, você pode dividir o número de palavras que apareceram em alguma das traduções de referência pelo número total de palavras na frase de saída. Isso nos dá uma medida chamada precisão unigrama. No nosso caso, a saída tinha 5 palavra e 4 delas aparecem na referência, um score de 0,8.

Um problema de contar as palavras apenas é que a ordem em que elas aparecem na saída não importa. Podemos contornar isso contando, não palavras individuais, mas palavras que ocorrem próximas umas das outras. Esses conjuntos são chamados de n-gramas, onde n é o número de palavras por grupo. Unigramas, bigramas, trigramas e 4-gramas são compostos por blocos de uma, duas, três e quatro palavras, respectivamente. Esse é o princípio básico por trás da medida BLEU, que ainda incorpora a penalizações para saídas menores que a referência, pois sentenças menores aumentariam artificialmente o score.

A definição oficial pode ser encontrada no artigo que propos a medida, `https://www.aclweb.org/anthology/P02-1040.pdf`. Existem diversas implementações em Python para calcular esta medida. No entanto a medida BLEU não é perfeita e vem recebendo diversas críticas, pois sua correlação com a impressão humana é baixa, ela não leva em consideração o significado das palavras e mede apenas co-ocorrências de trechos.

Para este trabalho, cada um dos dois geradores de texto deverá ser avaliado pelas seguintes métricas:

- acurácia média,
- BLEU,
- NIST,
- METEOR.

Estas duas últimas medidas devem ser pesquisadas na internet.

Importante: não fique impressionado se os números forem ruins. É uma tarefa difícil, e as medidas não são muito intuitivas. O que importa aqui é comparar os dois modelos.

Instruções entrega

Entregar um zip contendo 3 arquivos:

1. Os programas fonte dos dois geradores de títulos.
2. Um arquivo README que indica como treinar, ativar e rodar as avaliações. Indicar onde devem ficar os datasets de treino, validação e teste. Indicar também se foi usado o córpus completo ou a versão abreviada na geração dos datasets.

3. Um pdf com os gráficos das medidas de qualidade da geração de textos. Insira neste pdf as informações que achar necessário para esclarecer sua solução.