Relatório - EP3

Piero Conti Kauffmann (8940810)

1 Modelos utilizados

1.1 Encoder-Decoder BiLSTM

Pelas especificações da tarefa, o primeiro modelo proposto para o problema de geração automática de títulos deve ser uma rede neural encoder-decoder com um encoder LSTM bidirecional ($\overrightarrow{g_e}$ e $\overleftarrow{g_e}$) acoplado a um mecanismo de atenção (Figura 1).

Na componente do decoder do modelo escolhi usar duas LSTMs unidirecionais em sequência. A primeira LSTM recebe o embedding do último token escrito no título e é inicializada com os estados finais da rede bidirecional concatenados, portanto possuí o dobro de *hidden units* de $\overrightarrow{g_e}$ e $\overleftarrow{g_e}$. Além disso, é na primeira LSTM do decoder que é feito o cálculo dos vetores de contexto por meio do mecanismo de atenção do modelo, descrito adiante. Esses vetores são concatenados aos *hidden states* da LSTM e são passados como entrada para a LSTM final, que finaliza a decodificação do próximo token da sequência.

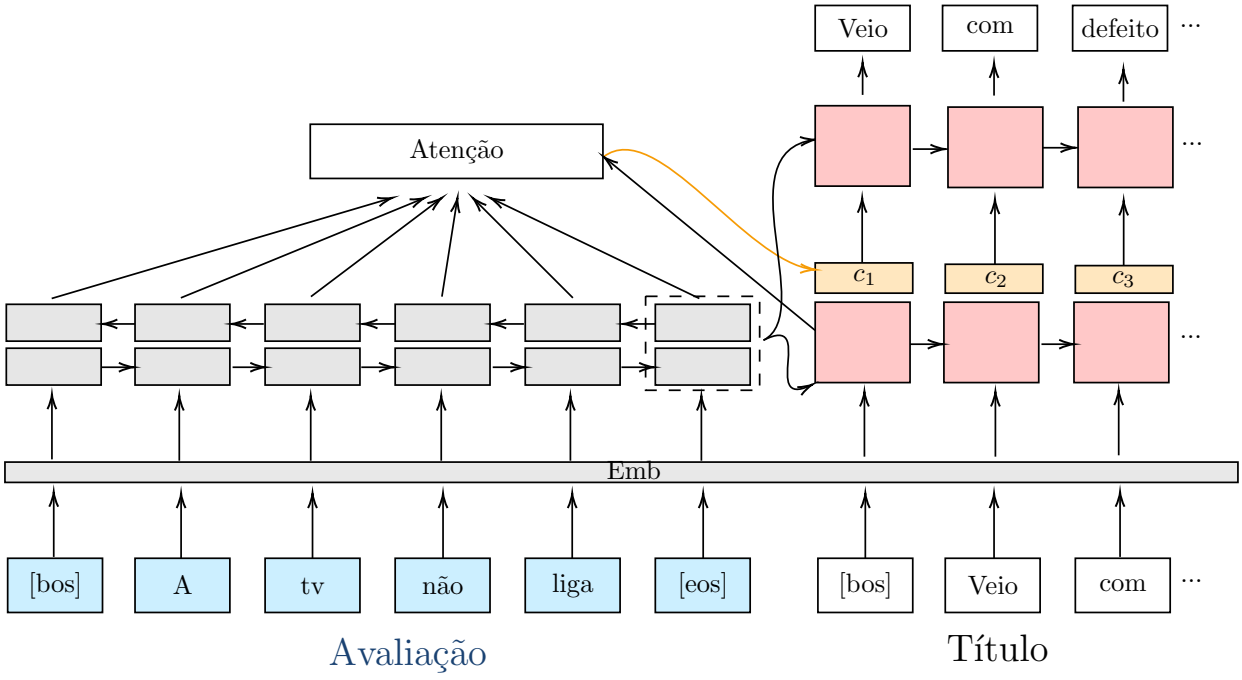

Figura 1: Diagrama do modelo encoder-decoder BiLSTM com camada de atenção escolhido. Os tokens especiais `[bos]` e `[eos]` delimitam respectivamente o início e fim das sequências de texto.

O mecanismos de atenção adotado para o modelo foi o mecanismo de atenção global proposto por Luong, Pham e Manning [3] com escores de atenção obtidos a partir do produto interno dos *hidden states*. Graças à existência da segunda LSTM do decoder, o modelo é também é capaz de incorporar a informação dos vetores de contexto dos tokens passados.

1.2 BERT-CLS e BERT-MASK

Neste trabalho iremos experimentar com duas variantes de soluções baseadas no BERT (representado na Figura 2) para geração de títulos de maneira autoregressiva. A primeira variante consiste em utilizar o campo de classificação (token `[CLS]`) de um modelo BERT pré-treinado para a língua portuguesa (Souza, Nogueira e Lotufo [4]) para prever o próximo token do título em sequência.

A camada final associada ao token `[CLS]` do BERT é pré-treinada com a tarefa de *Next Sentence Prediction* (NSP), que consiste em tentar adivinhar se a segunda sentença fornecida (separada pelo token `[SEP]`) vem depois da primeira sentença em um texto corrido. Para o problema em questão, podemos descartar a camada densa de classificação binária usada na tarefa de NSP e criar uma nova camada densa com a mesma dimensão final do vocabulário de saída, utilizando essa nova componente para prever o próximo token do título de maneira autoregressiva como se estivessemos resolvendo um problema de classificação supervisionada. Chamaremos esse modelo, descrito no enunciado do exercício, de BERT-CLS.

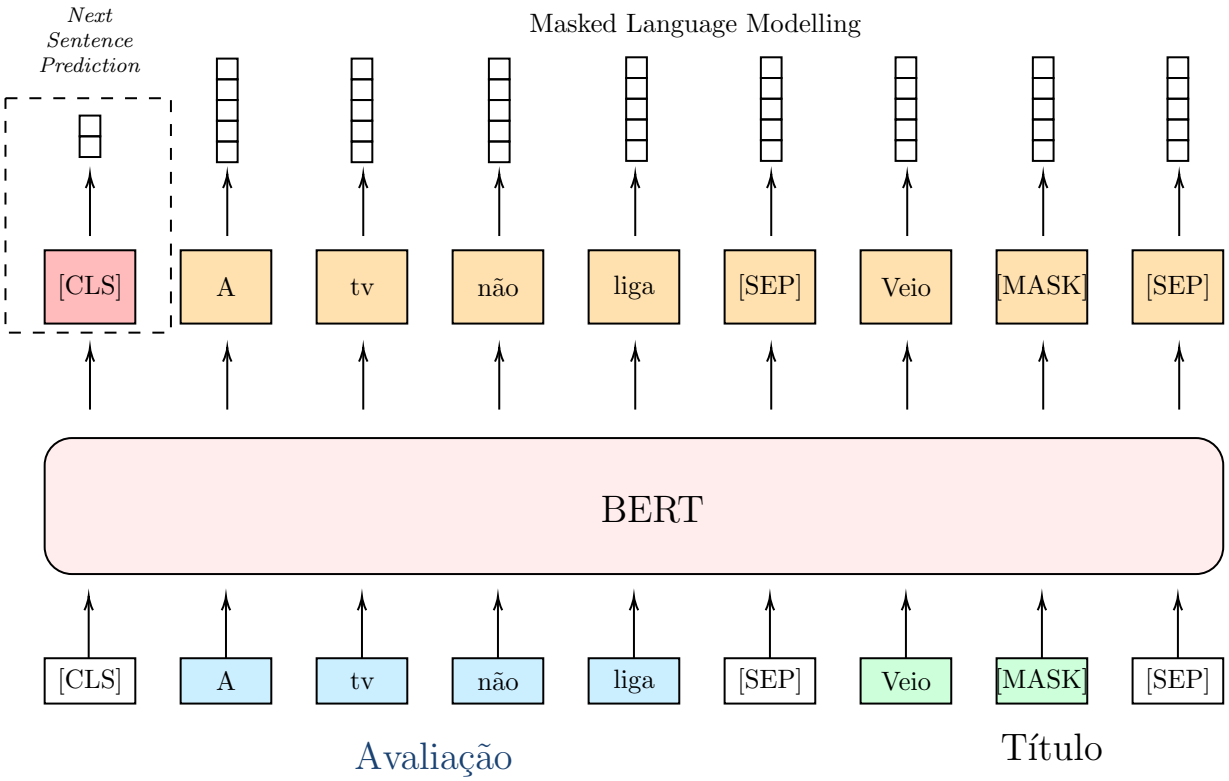

Figura 2: Diagrama ilustrativo do BERT para a task compartilhada de *Masked Language Modelling* e *Next Sentence Prediction*.

Alternativamente, podemos tentar aproveitar a semelhança do objetivo deste trabalho com a tarefa de *Masked Language Modelling* (MLM) do BERT pré-treinado, que tenta recuperar os tokens que são mascarados aleatóriamente nas sentenças passadas para o modelo. Apesar de parecer a melhor alternativa, existem algumas desvantagens aparentes: no objetivo de geração de texto deste trabalho, vamos sempre adicionar o token `[MASK]` no final da segunda sentença (título), que estará sempre seguido do token final `[SEP]`. Naturalmente, isto irá fazer o modelo original acreditar que o token mascarado é sempre um token próximo ao final de sentença (um caractere de ponto final, por exemplo) e irá prejudicar a performance do texto que iremos gerar de maneira iterativa. Ao fazermos o *fine tuning* segundo esta abordagem, estamos grosseiramente tentando converter um modelo treinado com a tarefa de *Masked Language Modelling* para um modelo causal de geração de títulos.

É difícil de prever de antemão se o BERT-MASK produzirá resultados melhores ou piores que o BERT-

CLS (proposto originalmente pelo enunciado da tarefa), então, a título de curiosidade, decidi avaliar a performance destas duas abordagens neste trabalho.

2 Treinamento

2.1 Encoder-Decoder BiLSTM

O treinamento da encoder-decoder BiLSTM é feito utilizando a técnica *teacher forcing* (Williams e Zipser [5]), que passa a sequência correta inteira de tokens para o decoder que tenta prever o próximo token relativamente a cada item da sequência passada. Essa técnica na prática acelera a convergência de modelos sequenciais e usualmente também facilita a implementação computacional destes modelos.

Treinamos o modelo com *early-stopping* em uma amostra de 72% do conjunto de dados da base completa da B2W, validando o modelo ao final de cada época em um conjunto de validação que corresponde a 8% dos dados. O treinamento é interrompido se o custo calculado no conjunto de validação não diminuir em 5 épocas consecutivas. Quando o treinamento é interrompido, o modelo com menor custo no conjunto de validação é salvo para a etapa de previsão.

Na Figura 3, é representado o mapa de atenção extraido do modelo para um exemplo do conjunto de validação. Verificamos que o modelo treinado foca a atenção em algumas palavras chaves úteis para gerar o título "Produto bom", como "não trava"e "Sem comentários".

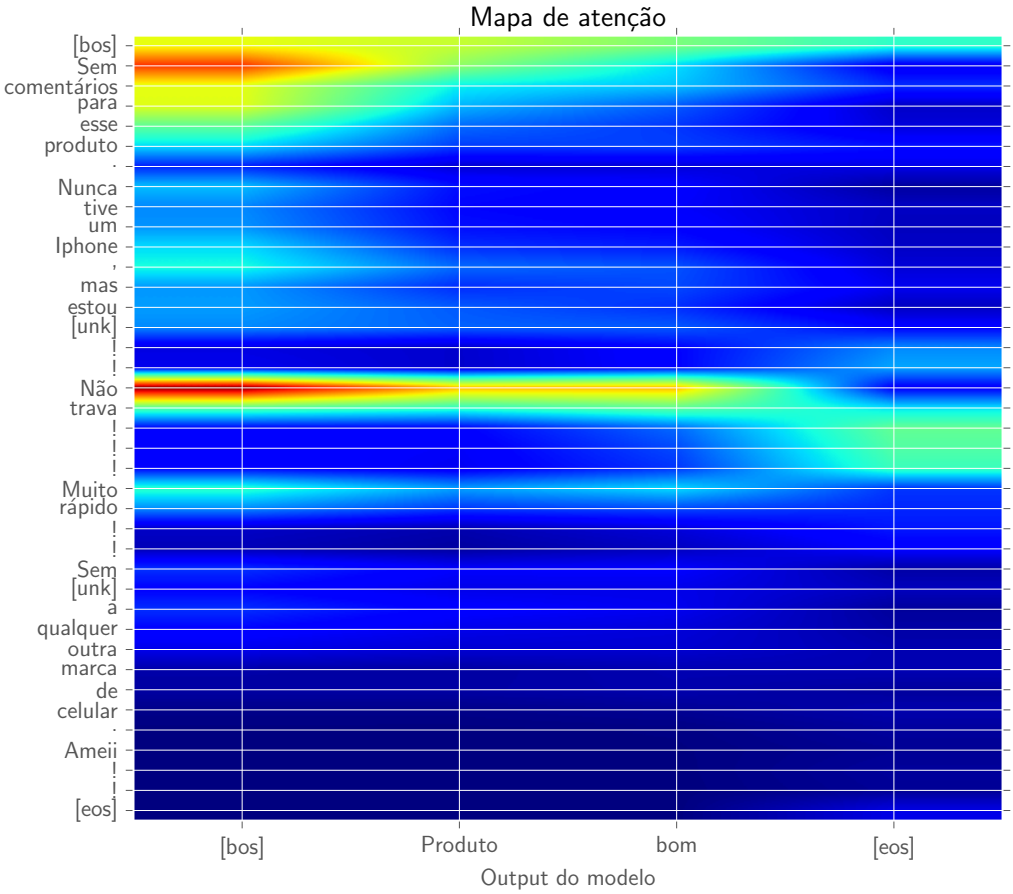

Figura 3: Mapa de atenção para uma avaliação do conjunto de validação com o título gerado pelo modelo recursivamente (eixo horizontal) e a avaliação submetida pelo cliente (eixo vertical).

2.2 BERT-CLS e BERT-MASK

Seguindo o procedimento de preparação dos dados descrito no enunciado do exercício, utilizamos a mesma amostra de 80% dos dados da base completa da B2W extraidos para o treinamento da LSTM. Após dividir os dados para a geração de texto token a token (conforme descrito no enunciado) o conjunto de treino final gerado possui cerca de 600 mil instâncias. Dada o elevado número de instâncias de treinamento e o alto nível de esforço computacional necessário no treinamento, os modelos são treinados durante apenas uma única época completa no conjunto de treinamento. Por sua vez, como a função de custo foi calculada apenas em dados que não foram vistos previamente pelo modelo, não foi necessário utilizar *early-stopping* para previnir *overfitting*.

O modelo BERT-CLS convergiu de maneira estável, sem precisar de um tamanho de *batch* muito elevado, finalizando após cerca de 10h de treinamento no *Google Colab*. O modelo BERT-MASK exigiu um esforço maior de treinamento e teve convergência difícil, o que é justificável, visto que neste modelo alteramos a tarefa base de *Masked Language Modelling* em que o modelo foi originalmente treinado.

Para garantir maior estabilidade no treinamento do BERT-MASK, foi necessário utilizar um tamanho de *batch* maior, e consequentemente um número maior de iterações de *gradient accumulation* para possibilitar o treinamento na infraestrutura disponível para o EP. O treinamento do modelo durou cerca de 20h na plataforma do *Google Colab*.

As especificações principais de treinamento de ambos modelos podem ser consultadas na Tabela 1 a seguir.

Modelo	lr	*batch*	GAcc	Total *batch* / iteração
BERT-CLS	5e-5	10	2	20
BERT-MASK	5e-6	12	4	48

Tabela 1: Hiperparâmetros de treinamento dos modelo BERT-CLS e BERT-MASK: taxa de aprendizado do otimizador (lr), tamanho do *batch* (*batch*), número de iterações de *gradient accumulation* (GAcc) e o tamanho total do *batch* por iteração.

3 Resultados experimentais

Os modelos foram avaliados na amostra de testes (que representa 20% do conjunto de dados completo da B2W) segundo as métricas:

- Acurácia
- BLEU-1 (unigramas)
- BLEU-2 (unigramas e bigramas)
- BLEU-3 (unigramas, bigramas e trigramas)
- BLEU-4 (unigramas, bigramas, trigramas e quadrigramas)
- Medida-Namorada
- METEOR

Como os títulos gerados pelos modelos são usualmente curtos, o que pode impossibilitar a geração de n-gramas, usarei o suavizador proposto por Lin e Och [2] para calcular as precisões da métrica BLEU:

$$P_n = \frac{|\text{Modelo}_n \cap \text{Ref}_n| + 1}{|\text{Modelo}_n| + 1}, \quad n \in \{2, 3, 4\} \tag{1}$$

onde Ref_n e Modelo_n são respectivamente os conjuntos dos n-gramas do texto de referência e do título gerado pelo modelo. Para um estudo comparativo entre diferentes técnicas de suavização, ver Chen e Cherry [1].

A métrica Medida-Namorada foi obtida a partir da avaliação manual de uma avaliadora externa que não participou do desenvolvimento deste trabalho. A avaliação consistiu em dar uma nota de 0 à 10 para os títulos gerados pelos três modelos a partir de uma amostra de 200 avaliações do conjunto de testes. No final da avaliação, a nota foi normalizada para uma escala de 0 à 1. A planilha de preenchimento com os resultados de avaliação pode ser consultada neste link.

Os resultados obtidos pelos três modelos são apresentados na Tabela 2.

Modelo	Acurácia	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	M-Namorada
BiLSTM	2,7%	16,5%	9,8%	6,7%	4,9%	10,8%	49,8%
BERT-CLS	3,5%	17,0%	10,4%	6,8%	4,7%	10,3%	61,3%
BERT-MASK	**4,9%**	**17,6%**	**11,1%**	**7,6%**	**5,4%**	**12,5%**	**65,5%**

Tabela 2: Métricas avaliadas no conjunto de testes para os três modelos.

Verificamos que nas 7 métricas usadas para comparar os modelos, o modelo BERT-MASK foi o que obteve o melhor resultado, enquanto a BiLSTM apresentou em média o pior resultado dos três modelos. Comparando os resultados obtidos na avaliação humana entre os modelos BERT-CLS e BERT-MASK, podemos hipotetizar que mesmo com grandes alterações na tarefa original, o modelo BERT-MASK pareceu aproveitar melhor o modelo pré-treinado. Porém, para testar efetivamente essa hipótese seria necessário um tratamento estatístico mais adequado e um número maior de experimentos.

É possível verificar também que na métrica medida via avaliação humana (M-Namorada), o gap entre os modelos baseados no BERT e a BiLSTM é muito maior do que nas demais métricas. Uma análise mais detalhada desta métrica revela que o modelo BiLSTM parece produzir títulos muito simplistas e às vezes é incapaz de identificar corretamente a situação que o cliente descreve, alguns exemplos destas situações são apresentados na Tabela 3.

Referências

[1] Boxing Chen e Colin Cherry. "A systematic comparison of smoothing techniques for sentence-level bleu". Em: *Proceedings of the Ninth Workshop on Statistical Machine Translation*. 2014, pp. 362–367.

[2] Chin-Yew Lin e Franz Josef Och. "Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics". Em: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. 2004, pp. 605–612.

[3] Minh-Thang Luong, Hieu Pham e Christopher D Manning. "Effective approaches to atib tention-based neural machine translation". Em: *arXiv preprint arXiv:1508.04025* (2015).

[4] Fábio Souza, Rodrigo Nogueira e Roberto Lotufo. "BERTimbau: pretrained BERT models for Brazilian Portuguese". Em: *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*. 2020.

[5] Ronald J Williams e David Zipser. "A learning algorithm for continually running fully recurrent neural networks". Em: *Neural computation* 1.2 (1989), pp. 270–280.

Review	BiLSTM	N	BERT-CSL	N	BERT-MASK	N
Ótima bicicleta , minha sobrinha amou... chegou na data prevista ... vendedor excelente	Excelente produto	7	Ótimo produdo	7	Ótima bicicleta	10
Estou aguardando a troca do produto, enviei pelo correio o produto com defeito e estou esperando a loja retornar com a resposta sobre a troca ou devolução do valor.	Não recebi o produto	0	Produto com defeito	7	Produto com defeito	7
Não gostei do produto péssima qualidade. Soltou todas os encaixe de metal..Solicito a troca do produto ou devolução do dinheiro. ... os encaixe de metal soltaram e não mas na piscina e uma lona final e simples....fiquei decepciona com produto fora o atraso na entrega. ....péssimo aguardo solução	Não recebi o produto	0	Péssimo produto	7	Não gostei do produto péssima qualidade. Soltou	10
Só imprime, nem cópia ela tira! Tão simples que poderia pelo menos vir com o cabo para o PC. Ela é branca, mas o fio de luz é preto. Impressora quebra galho, apenas compre como último recurso!	Bom	0	PÉSSIMA	6	Impressora quebra galho	9

Tabela 3: Exemplos selecionados da avaliação humana (Medida-Namorada) dos títulos gerados pelos três modelos e suas respectivas notas (0-10). Foram escolhidos exemplos em que o modelo BiLSTM não é capaz de identificar corretamente a situação ou de produzir um título com bom nível de detalhamento.