Domain adaptation challenges of BERT in tokenization and sub-word representations of Out-of-Vocabulary words

Anmol Nayak, Hari P. Timmapathini, Karthikeyan Ponnalagu, Vijendran Venkoparao
ARiSE Labs at Bosch
{Anmol.Nayak, HariPrasad.Timmapathini, Karthikeyan.Ponnalagu, GopalanVijendran.Venkoparao}@in.bosch.com

Abstract

BERT model (Devlin et al., 2019) has achieved significant progress in several Natural Language Processing (NLP) tasks by leveraging the multi-head self-attention mechanism (Vaswani et al., 2017) in its architecture. However, it still has several research challenges which are not tackled well for domain specific corpus found in industries. In this paper, we have highlighted these problems through detailed experiments involving analysis of the attention scores and dynamic word embeddings with the BERT-Base-Uncased model. Our experiments have lead to interesting findings that showed: 1) Largest substring from the left that is found in the vocabulary (in-vocab) is always chosen at every sub-word unit that can lead to suboptimal tokenization choices, 2) Semantic meaning of a vocabulary word deteriorates when found as a substring in an Out-Of-Vocabulary (OOV) word, and 3) Minor misspellings in words are inadequately handled. We believe that if these challenges are tackled, it will significantly help the domain adaptation aspect of BERT.

1 Introduction

BERT is one of the prominent models used for a variety of NLP tasks. With the Masked Language Model (MLM) method, it has been successful at leveraging bidirectionality while training the language model. The BERT-Base-Uncased model has 12 encoder layers, with each layer consisting of 12 self-attention heads. The word representations are context-dependent 768 dimensional dynamic embeddings. In order to leverage the learnings of such pre-trained networks, fine tuning is commonly done while building NLP applications in industries.The BERT-Base-Uncased vocabulary has a size of 30522 with only 994 unused slots (in comparison, BERT-Base-Cased has only 101 unused slots). While the unused slots in the vocabulary can be used to include domain specific words, the representations of these will have to be fine tuned with domain specific corpus before they can be utilized. Hence, it is essential that the tokenization algorithm performs well to handle domain specific OOV words.

BERT relies on the WordPiece algorithm (Schuster and Nakajima, 2012) to create the vocabulary, that chooses those sub-word units for the vocabulary towards maximising the language model likelihood. However, the tokenization using this vocabulary is not done semantically. This leads to a poor tokenization that induces a semantic information loss in terms of dealing with OOV words for domain centric downstream tasks. While the largest substring tokenization problem can be alleviated to a large extent by integrating recent algorithms like BPE-Dropout (Provilkov et al., 2019) or SentencePiece (Kudo and Richardson, 2018), that use frequency and/or language model based tokenization, the remaining aforementioned challenges still persist. In the following section, we discuss these challenges with respect to two categories: Tokenization and Sub-word representations.

2 Experiments

The experiments we performed are using the pre-trained BERT-Base-Uncased model without any domain specific fine tuning as the examples were chosen with the purpose of highlighting the challenges with BERT across various domains. The 12^{th} encoder layer dynamic embeddings were used for all the analysis tasks. In case a word was OOV, the average of its sub-word units embeddings was considered as its embedding. Otherwise, the embedding for the word was considered as it is. In all cases, we ignore the [CLS] and [SEP] embeddings while computing the embedding of a par-

Proceedings of the First Workshop on Insights from Negative Results in NLP, pages 1–5

ISBN 978-1-952148-66-8

Input	Tokenized
deconstructed	[CLS], deco, ##nst, ##ru, ##cted, [SEP]
deactivated	[CLS], dea, ##ct, ##ivated, [SEP]
unequal	[CLS], une, ##qual, [SEP]
ccabbage	[CLS], cc, ##ab, ##bag, ##e, [SEP]
cababge	[CLS], cab, ##ab, ##ge, [SEP]
cabbagee	[CLS], cabbage, ##e, [SEP]']
unsaturated	[CLS], un, ##sat, ##ura, ##ted, [SEP]
saturated	[CLS], saturated, [SEP]
pork has saturated fat	[CLS], pork, has, saturated, fat, [SEP]
pork has ##sat ##ura ##ted fat	[CLS], pork, has, ##sat, ##ura, ##ted, fat, [SEP]

Table 1: BERT tokenized representations.

Cosine similarity	$\text{Count}_{\text{beg}}$	$\text{Count}_{\text{mid}}$	$\text{Count}_{\text{end}}$
0.0-0.1	0	1	0
0.1-0.2	6	1	1
0.2-0.3	39	44	19
0.3-0.4	160	232	113
0.4-0.5	368	562	232
0.5-0.6	264	442	213
0.6-0.7	96	145	179
0.7-0.8	15	60	145
0.8-0.9	0	0	23
0.9-1.0	0	0	1
Avg. similarity:	**0.474**	**0.49**	**0.552**

Table 2: Cosine similarity score count for words vs misspelled versions in the TOEFL-Spell corpus when the error occurs at the beginning, middle or end of the word.

ticular word. The ## counterpart of a word is ## prefixed to the word. For example, the ## counterpart of the word *active* is *##active*. We also made a subtle change to the tokenizer to leave untouched any word beginning with *##*.

2.1 Tokenization problems

BERT always picks the largest substring from the left that is in-vocab at every sub-word unit for the tokenized output. While this performs reasonably well for words where the root (or stem) are suffixed, prefixed words are vulnerable to a poor tokenization.

Taking *deconstructed*, *deactivated* and *unequal* as examples, even though the vocabulary had the prefixes *de* and *un* as well as the words *constructed*, *activated* and *equal*, the tokenizer chose the substrings *deco*, *dea* and *une* (see Table 1). In comparison since SentencePiece is a likelihood based tokenization algorithm, it has managed to generate better tokenizations (*deconstructed*: *_de, con, struct, ed*; *deactivated*: *_de, activated*; *unequal*: *_unequal*). We believe that if the BERT tokenizer correctly separates the prefixes while the model is being trained, it can help the model to learn better representations for the prefix as well as the sub-word units since the attention mechanism would understand the influence of the different categories of prefixes. Further it can be seen in Section 2.2 how a poor tokenization can lead to weaker semantic representations for the word.

Domain specific corpus often contain a large amount of jargons that can be misspelled frequently. Taking the in-vocab word *cabbage* as an example, *ccabbage*, *cababge* and *cabbagee* were chosen as the misspelled versions. The cosine similarities of *cabbage* with *ccabbage*, *cababge* and *cabbagee* were 0.33, 0.44 and 0.63 respectively. To verify that the low cosine similarity scores in the misspelled versions were not due to lack of surrounding context, we checked the cosine similarity score between *cabbage* and *onion* (in-vocab) and found it to be 0.88.

To analyze the extent of this problem and the im-

Cosine similarity	$\text{Count}_{L=4}$	$\text{Count}_{L=5}$	$\text{Count}_{L=6}$	$\text{Count'}_{L=4}$	$\text{Count'}_{L=5}$	$\text{Count'}_{L=6}$
0.0-0.1	0	0	0	0	0	0
0.1-0.2	6	8	13	0	0	1
0.2-0.3	25	86	134	4	0	2
0.3-0.4	198	404	556	5	2	2
0.4-0.5	573	975	1159	21	17	7
0.5-0.6	622	933	1133	8	3	14
0.6-0.7	176	340	398	172	54	23
0.7-0.8	4	17	32	127	58	28
0.8-0.9	0	0	1	51	23	8
0.9-1.0	0	0	0	0	0	0
Avg. similarity:	**0.496**	**0.487**	**0.484**	**0.665**	**0.669**	**0.652**

Table 3: Cosine similarity score count for word vs OOV ## counterpart ($\text{Count}_{L=4,5,6}$) and word vs in-vocab ## counterpart ($\text{Count'}_{L=4,5,6}$).

Figure 1: Inward Attention visualization for *fat* in Encoder layer 1 - Attention head 3.

pact of the position of the error in the word on the tokenization, we chose the TOEFL-Spell corpus that contains over 6000 common spelling errors.[1] We took the intersection of the common words between the TOEFL-Spell corpus and the BERT vocabulary words. The corpus was segregated depending on whether the spelling error occurred in the word within the starting 33% of the letters, in the middle or at the end. As we can see in Table 2, since the BERT tokenizer has the largest substring problem, the penalty of a spelling error earlier in the word is more harmful as it leads to subsequent sub-word tokenization choices to be suboptimal.

2.2 Semantic meaning deterioration from sub-word representations

For a model to handle OOV words well, it should learn strong representations of a words constituents. While OOV words that begin with an in-vocab root (or stem) will retain its semantic meaning when tokenized, they become vulnerable in other cases as the root (or stem) will be broken down into smaller constituent sub-word units.

[1] https://github.com/EducationalTestingService/TOEFL-Spell

To see how BERT handles this, we created two sets of words from the vocabulary of length 4,5 and 6 that were consisting of: 1) Words whose ## counterparts were OOV and 2) Words whose ## counterparts were in-vocab. We chose these particular words since a word with length less than 4 would be commonly be found as a sub-word across many words, while a word with length larger that 6 would be rarer to be found as a sub-word. The cosine similarity between a word and its ## counterpart was computed (see Table 3). This problem is not a concern when the ## counterpart is in-vocab as the average cosine similarity was around 0.66, which can be improved if supplied with a context in a sentence. However, when the ## counterpart is not part of the vocab, the average cosine similarity drops to a low value of 0.48, which makes it difficult for the network to recover from.

To further analyze this problem, we compared the embeddings of the words *unsaturated* (OOV but *un* and *saturated* are in-vocab) with *saturated*. The cosine similarity between *unsaturated* and *saturated* was only 0.30. In comparison, the cosine similarity between *un saturated* and *saturated* is 0.81. To verify that this low similarity was being

Layer	Influence	[CLS]	pork	has	fat	[SEP]
1	Outward	-0.00905915	0.06523646	0.06099863	**0.18786138**	0.03023722
	Inward	-0.01837084	0.04225173	0.01648326	**0.18937206**	0.01520266
2	Outward	-0.0004705	0.00843187	-0.00484527	0.02482779	0.01454217
	Inward	**0.17178166**	-0.01259612	-0.03369101	0.01195145	0.01798595
3	Outward	0.00184171	0.01289389	**0.12865019**	0.08927301	0.00345621
	Inward	0.03889459	0.01265209	-0.01004721	**0.15340301**	-0.00876677
4	Outward	-0.00047312	-0.00136977	0.03939556	0.03750715	-0.00168958
	Inward	0.04355699	0.00727928	0.04730521	0.04906496	0.01223874
5	Outward	0.0048245	-0.00982189	-0.02644702	0.02499489	-0.00251921
	Inward	0.00937152	-0.00695391	-0.01893955	0.05315585	**0.16019171**
6	Outward	-0.00265443	-0.01301921	0.0342003	0.01597476	-0.00048893
	Inward	-0.00409265	0.01658719	0.01285958	0.02900258	0.05893058
7	Outward	-0.00125295	0.00154607	-0.01132394	0.01854416	0.00031497
	Inward	0.00937736	-0.04564465	0.04480758	0.01291878	**0.11453587**
8	Outward	-0.00527599	-0.01248677	0.00545111	0.00576302	-0.00044466
	Inward	0.00128797	0.00911033	0.06117178	-0.02267864	0.06066525
9	Outward	-0.01058492	0.00061043	0.03902156	0.03468442	0.00406067
	Inward	0.0006195	0.01115873	0.03967107	-0.00104411	0.02020252
10	Outward	0.01764568	0.00554455	0.02936521	0.03989781	0.01815606
	Inward	0.03199076	0.03799912	0.01782591	-0.00910724	0.02136517
11	Outward	0.02042433	0.0195443	0.01784758	0.02018948	0.00963123
	Inward	-0.03223545	0.08976553	0.04230637	0.04836676	-0.10472655
12	Outward	0.02328578	0.00367201	0.00402358	0.03388398	0.00260962
	Inward	0.01822337	0.0257406	0.02513322	0.03337914	-0.0509001

Table 4: Difference in inward and outward attention scores between *saturated* and *##sat ##ura ##ted*.

Word	Nearest neighbours
saturated	bacon, nutrition, cereal, obesity, flour, tobacco, humidity, mustard, cigarettes, vitamin
##sat ##ura ##ted	destruction, egypt, erosion, malaria, morphology, concussion, organ, topography, aroused, sample

Table 5: Top 10 cosine similar nearest neighbours in the vocabulary for *saturated* and *##sat ##ura ##ted* as found in the sentences.

caused by the poor representation learning of the constituents of *saturated*, we compared the average embedding of *##sat*, *##ura*, *##ted* (since *unsaturated* was tokenized into sub-word units) with *saturated* and found their cosine similarity to be only 0.35.

Further, to rule out the possibility that it was being caused due to lack of surrounding context, we compared the average embedding of *##sat*, *##ura*, *##ted* with *saturated* as found in the following sentences: *pork has saturated fat* and *pork has ##sat ##ura ##ted fat*.

The cosine similarity even in this case was found to be only 0.57. For the above sentences, we wanted to see the impact of this problem by analyzing the attention scores in each encoder layer. The multi-head (12 heads) attention score matrix across the 12 encoder layers is of size 12 x 12 x 6 x 6 for the first sentence and 12 x 12 x 8 x 8 for the second sentence. Within each layer, we averaged the attention scores across the 12 heads. This resulted in 12 x 6 x 6 and 12 x 8 x 8 sized attention scores matrices for the two sentences respectively. We wanted to observe the inward influence of other words on *saturated* as well as outward influence of *saturated* towards the other words in both sentences. For the first sentence, the attention score matrix was hence reduced to a size of 12 x 5 x 5. In the second sentence, we averaged the inward and output influence for *##sat ##ura ##ted*, leading

to a reduced matrix of size 12 x 5 x 5. The two matrices were then subtracted to see the difference in the inward and outward influences for the word *saturated* (see Table 4).

Since the difference was taken, a positive value means *saturated* as found in the first sentence had a larger inwards or outwards attention influence compared to the second sentence. Clark et al. (2019) previously showed that a large number of attention heads in the early layers of BERT put >50% of their attention on previous and next tokens. As we can see in Table 4, the values in bold show a significant difference in attention scores, especially in the case for the neighbouring words of *saturated*, which we believe has caused the loss of semantic meaning between *saturated* and when tokenized to *##sat, ##ura, ##ted*. The inward attention visualization for *fat* in Encoder layer 1 - Attention head 3 generated using BertViz (Vig, 2019) can be seen in Figure 1. Further, we checked the top 10 cosine similar neighbours in the BERT-Base-Uncased vocabulary (using their dynamic embeddings) for the embeddings of *saturated* and *##sat ##ura ##ted* from the above sentences. We found that while *saturated* as found in the first sentence had semantically similar neighbours, its occurrence in the second sentence had neighbours which had a completely irrelevant semantic meaning (see Table 5). This confirmed that such a challenge can lead to cascading problems in the network.

3 Conclusion

In this paper we highlighted various challenges in the BERT model which if solved could significantly boost the models accuracy, especially in domain specific applications. These are mainly due to BERT lacking a semantic tokenization algorithm and its semantic information loss from sub-word representations in OOV scenarios.

References

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2019. Bpe-dropout: Simple and effective subword regularization. *arXiv preprint*, arXiv:1910.13267.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.