Jae-Hong Min MD
W266
Final Project Submission

Using Root Word Tokenization to Improve Performance of Biomedical Natural Language Processing Tasks

The structure of biomedical language is very modular in the sense that root words are put together to try to explain a complex process. With this there is a lot of context in the words that could be lost via tokenization. To try to keep as much context as possible, a Latin and Greek root word tokenizer was created. However, tokenization with specific root words is superior to non-domain trained models, it does not seem to show an immediate benefit over domain specifically trained models, especially when paired with smaller training sets.

There has been great progress made in the field of natural language processing (NLP) with advancements such as Word2Vec [1], ELMo [2] and BERT[3]. However,the biomedical field has not had the same level of success with these new models.[4]
Many different groups have tried to overcome this problem by using domain specific corpora to pretrain BERT and they have had some degree of success such as BioBert[5] and PubMedBert[6]. However, this success comes at a large computational cost, as the BioBert model training time was computationally expensive (8x NVIDIA V100 (32GB) GPUs took 23 days to pretrain BERT with corpus).

However, there may be alternative approaches to improve BERT performances in the biomedical domain. Due to the complexity of biomedical concepts, most biomedical terms are usually a concatenation of multiple “root” words, usually from Greek or Latin roots.

For example:
“Myocardial Ischemia” which is a form of a heart attack has 4 root words:

My(o):	Muscle	Gives physiological information
Card(ial):	Related to the heart	Gives anatomical information
Ische-:	To suppress	Gives pathological information
(-)emia:	Related to blood	Gives pathophysiological information

Which combined together gives “Muscle related to heart gets suppression of blood” which will give significant contextual information.
However, if the same term is put through the tokenizer for BERT (and thus BioBERT) it gives:

```
'[CLS]', 'my', '##ocar', '##di', '##al', 'is', '##chem', '##ia', '[SEP]'
```

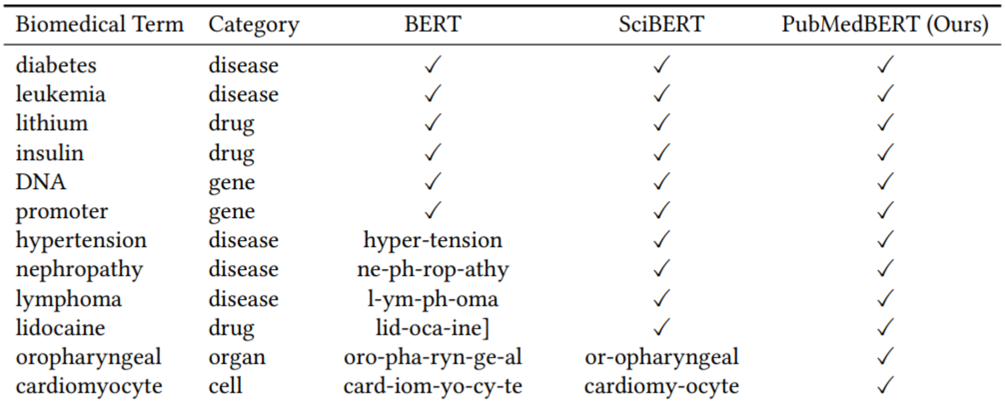
The issue with this is that some of the tokens above such as “my” and “is” are real words with separate meanings that are non-related and the subwords also may have alternative meanings and context. Although it can be argued that with a large enough biomedical corpora the context may be inferred by the model this seems less likely unless there is an exorbitantly large corpora. Biomedical terms and combinations cannot be part of the vocabulary.

This project aims to test the hypothesis that a Biomedical Domain Specific Tokenizer which emphasizes on tokens that are related to Greek and Latin Root words will improve the performance of biomedical Natural Language Processing Tasks when compared to BERT, and possibly other domain specific

Background

If one takes a look at the approach that the above models have taken, the similarity with all the models there is the same large corpus of biomedical information available (PubMED) and then run it through BERT either completely retraining or fine tuning BERT.

Microsoft's PubMED was successful in retraining the BERT model including the tokenizer to have more medical terms within the tokenizer vocabulary:



Biomedical Term	Category	BERT	SciBERT	PubMedBERT (Ours)
diabetes	disease	✓	✓	✓
leukemia	disease	✓	✓	✓
lithium	drug	✓	✓	✓
insulin	drug	✓	✓	✓
DNA	gene	✓	✓	✓
promoter	gene	✓	✓	✓
hypertension	disease	hyper-tension	✓	✓
nephropathy	disease	ne-ph-rop-athy	✓	✓
lymphoma	disease	l-ym-ph-oma	✓	✓
lidocaine	drug	lid-oca-ine]	✓	✓
oropharyngeal	organ	oro-pha-ryn-ge-al	or-opharyngeal	✓
cardiomyocyte	cell	card-iom-yo-cy-te	cardiomy-ocyte	✓

From *Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing Table 1[6]*

However, from a theoretical perspective, because biomedical terms are "modular" or that they are created from a combination of terms, tokenizing a medical term as a single word may actually lose information and context surrounding the term.

Zipf's law states that there is a few very common words, and after that it "decays" exponentially which means that a Tokenizer like Wordpiece[7] which uses frequency and absolute count, would only pick up very "common" words but may miss out as the frequency logarithmically decays. These "uncommon" words will thus be broken down into subwords which may have no contextual meaning with the original word.

For example, in our dataset example in table 1:

If the term "cardiac ischemia" is tokenized to just "cardiac" and "ischemia", it may capture those terms, but in if the term "cardiomyocyte" which is extremely closely related to cardiac ischemia will be tokenized by BERT as "**card**" - "iom" - "yo" "cy" "te", then trying to contextualize it may be difficult. So if the root "cardi" was just kept, then any word such as "**card**iac arrest", "**card**iomyocyte", "**card**iac sarcoma", "cardiac ischemia" words would immediately be recognized as similar. This is just one of many examples.

Method

Tokenizer:

To create a better root word tokenizer, it was initially decided to manually program the tokenizer (which proved to be extremely challenging given the huge possible corpus of roots), it was rather decided to do a “supervised” creation of the tokenizer.

WordPiece was used as the trainer which was due to its compatibility with BERT and act as a control, and that it allowed for Subwords which other tokenizer types such as BPE would not allow. The corpus was carefully constructed to increase the probability of a root word being chosen as the subword. One had to also overcome the fact that a word such as “card” could be a word, a suffix, a prefix or a word in the middle of 2 words(each which would be interpreted differently by the machine, even though they had the same meaning or context.

To overcome this the corpus was created by gathering a large corpus of root words which was a mixture of online dictionaries, Wikipedia’s page on root words, medical glossary pages, corpora for ICD-10 coding as well as freely available Medical Syntax corpora[8-13].

This corpus was painstakingly cleaned via the regex library, manual cleaned and also filtered and cleaned algorithmically.

From here, to account for the possibile positioning of the words and to prevent wordpiece from thinking a root word as a single word, a random matching algorithm generated 5 million randomized root word combinations following the common bridge letters such as “o” and “i” as seen in card-io-my-o-pathy. This was then fed into the tokenizer.Tokenize library to create a tokenizer.The tokenizer was then converted to an appropriate BERT compatible format to be used in the training of BERT.

Modelling

To try to test for objectivity, the following where in the experiment group:

1. RootBert:
   Tokenizer - Root Tokenizer
   Pre-trained with PubMed corpus
2. Control 1:
   Tokenizer - bert-base-cased
   Fine-tuned with medical corpus from PubMed
3. Control 2:
   Tokenizer - bert-base-cased
   Pretrained with medical corpus from PubMed

These trained models were then Fine Tuned by acting as the base model for the Bert
For the testing of biomedical language tasks, a transformer BESTclassification model was used and then fine tuned on the NCBI corpus used for Name Entity Recognition.

The experiment parameters had 3 different corpus sizes - Small, Medium and Large from the same PubMed corpus which were random biomedical article abstracts from the year 2021 consisting of 10000, 25000 and 75000 lines of biomedical literature.

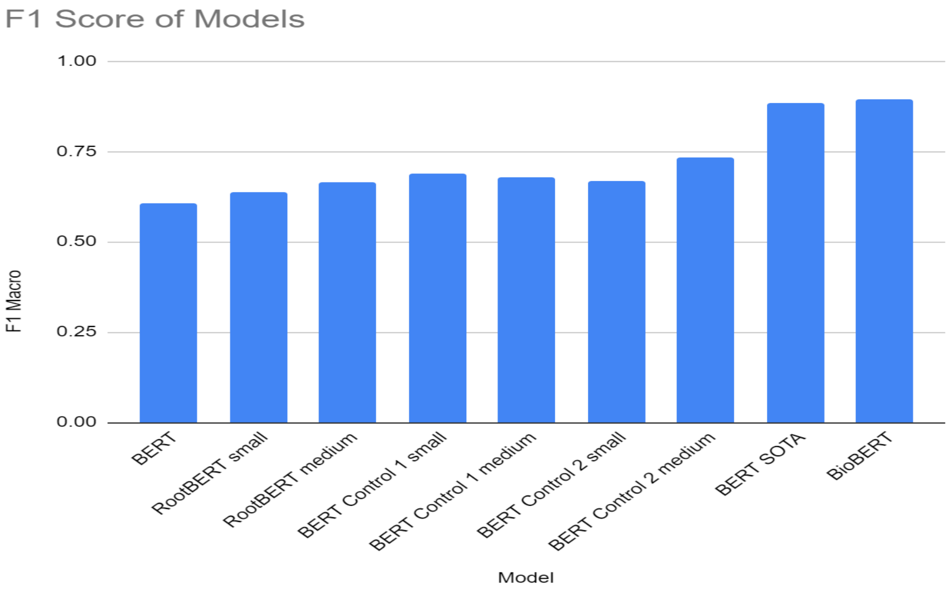
All the experiments had the same parameters including the same vocabulary size of the BERT original tokenizer and hyperparameters were also kept constant.

The final validation was done on a test dataset which consisted of using the F1 score on the NER using the "Macro" averaging on the sklearn.metrics.f1_score function. [14]

Results:
Tokenization:
There seems to be an improvement in the tokenization in terms of biomedical terms - as it would act similar to the higher performance models such as PubMedBERT where the medical terms are noted in the vocabulary while the BERT base would take the words apart.
And when there is a requirement to break it into root words, the Root tokenizer seems to perform better than BERT base and seems to do best for very complex compounded medical terms.

For example:

BERT base cased	RootBERT
`['or', '##op', '##har', '##yn', '##ge', '##al']`	`['oropharyngeal']`
`['l', '##ymph', '##oma']`	`['lymphoma']`
`['ne', '##ph', '##rop', '##athy']`	`['nephropathy']`
`['ta', '##chy', '##p', '##nea']`	`['tachyp', '##nea']`



ully.

However, when it came to the performance task of the NER, although it performed better than just a baseline BERT model, it did not perform better than the BERT models that were trained further with the same corpus RootBERT trained on.
There was a good increase in the F1 score from the different dataset sizes which may mean that it does require a larger dataset to train itself f

Discussion:
It is surprising that even with a relatively small corpus of information the performance beats the BERT base model given the large sizes of corpora that were used to train both sets.
And it is unlikely that the rootBert would be able to handle any day to day non-domain specific text or have better performance than BERT models that have been optimized
With the relatively high gain in the F1 score with the small training corpus and cheap computation it seems that it is likely that Root Word Specific Tokenizations may increase performance as it does perform better than its counterparts that were controlled for the training corpus as well as the fine tuning.

For future, the one area that I think would help increase the performance of RootBert is a large corpus for training. Currently, the corpus that it trained on was miniscule compared to the higher end models (25000 lines or 600,000 words vs 23 billion words or even against BERT (Wikipedia and Book Corpus). However, even with such a low count of training data set it did seem to still perform better than a baseline bert without any further training.

Conclusion
Although it is likely that with a large enough corpus that domain specific tokenization or root word tokenization would not be required, given the impracticality of this, a biomedical tokenizer using Latin and Greek root words as tokens may be a reasonable approach to approaching biomedical language tasks with performances that are within very good for the amount of computational time and the limited corpus of text.

References

1. Mikolov, Tomas, et al (2013) “Efficient Estimation of Word Representation in Vector Space”. arXiv:1301.3781 [cs.CL]
2. Peters, Matthew, et al (2018) “Deep contextualized word representations”. arXiv:1802.05365 [cs.CL]
3. Devlin, Jacob, et al (2019) “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. arXiv: 1810.04805v2 [cs.CL]
4. Habibi,M. et al. (2017) “Deep learning with word embeddings improves biomedical named entity recognition.” Bioinformatics, 33, i37–i48.
5. Lee, Jinhyuk, et al (2019) “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. Bioinformatics, 2019, 1-7.
6. Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. 1, 1, Article 1 (January 2021), 24 pages. https://doi.org/10.1145/3458754
7. Wu,Y. et al. (2016). “Google’s neural machine translation system: Bridging the gap between human and machine translation”. arXiv preprint arXiv: 1609.08144 [cs.CL]
8. https://www.nlm.nih.gov/databases/download/mesh.html
9. https://en.wikipedia.org/wiki/List_of_medical_roots,_suffixes_and_prefixes
10. https://en.wikipedia.org/wiki/Wikipedia:Database_download
11. https://pubmed.ncbi.nlm.nih.gov/18830537/
12. https://www.cob.cms.hhs.gov/Section111/assets/section111/icd10.dx.codes.htm
13. https://nlmpubs.nlm.nih.gov/projects/mesh/MESH_FILES/xmlmesh/
14. Dogan,R.I. et al. (2014) “NCBI disease corpus: a resource for disease name recognition and concept normalization”. J. Biomed. Inform., 47, 1–10.
15. Tsatsaronis,G. et al. (2015) “An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition”. BMC Bioinformatics, 16, 138.