CMPE493 Term Project

BIOCREATIVE – VII

Berkay Demirtaş
Sabri Gökberk Yılmaz
Alper Canberk Balcı

TABLE OF CONTENTS

Presentation Outline

PROJECT PURPOSE

Classifying COVID-19 related papers in BioCreative-VII challenge

Classes: Treatment, Diagnosis, Prevention, Mechanism, Transmission, Epidemic Forecasting, Case Report

TRAIN DATASET BEFORE PREPROCESSING

-Train dataset contains 24960 articles and there are 7 columns: pmid, journal, title, abstract, keywords, pub_type, authors, DOI, label

-Dev dataset contains 6238 articles.

-There are total 6168752 tokens in abstract and title parts of the articles before preprocessing.
This means the average number of tokens in an article (title+abstract) is 247.14 before preprocessing.

-There are 7 classes in total andtheir distribution is as follows:
Treatment : 8717, Diagnosis : 6193, Prevention : 11102, Mechanism : 4438,
Transmission : 1088, Epidemic Forecasting : 645, Case Report : 2063

PREPROCESSING

-When we checked the papers given in the Project document, we realized that almost all of the papers are using abstract and title parts of the articles. Therefore, we also used abstract and title parts.

-We merged these parts and tokenized the merged string.

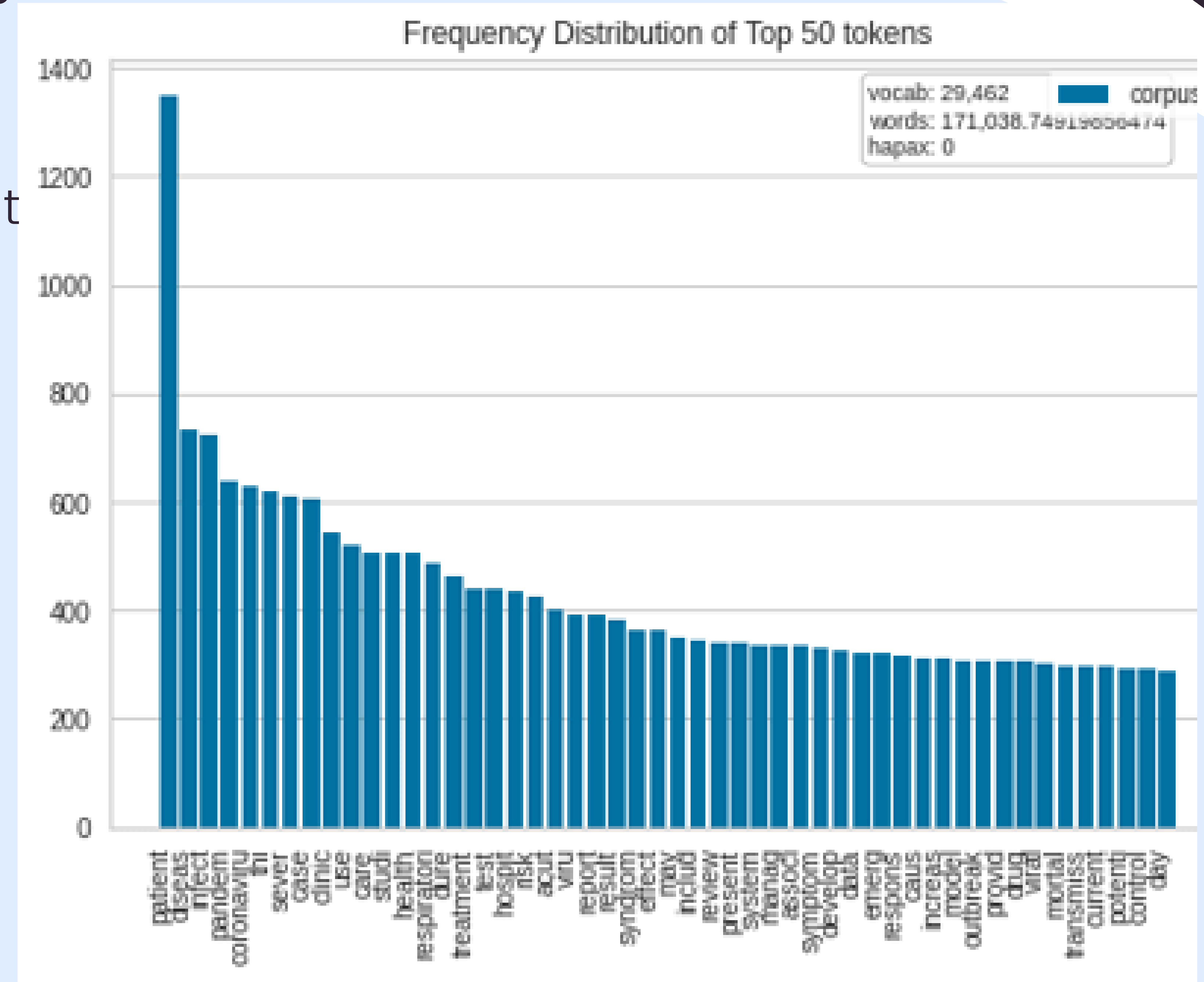
-After tokenization, we did case-folding and removed the tokens containing non-alphanumeric characters.

-We did stemming, lemmatization, stopword removal using nltk library.

TRAIN DATASET AFTER PREPROCESSING:

120.04 tokens per article, dataset almost halved in size after preprocessing(247.14 prior)

frequencies of the top 50 tokens



MULTINOMIAL NAIVE BAYES CLASSIFIER

-Model's scores for less frequent classes are not satisfying at all.

-More frequent classes have high precision and recall.

	precision	recall	f1-score	support
Treatment	0.8805	0.7345	0.8009	2207
Diagnosis	0.8921	0.5136	0.6519	1546
Prevention	0.9192	0.9145	0.9169	2750
Mechanism	0.9409	0.5638	0.7051	1073
Transmission	0.0000	0.0000	0.0000	256
Epidemic Forecasting	0.0000	0.0000	0.0000	192
Case Report	1.0000	0.0353	0.0681	482
micro avg	0.9062	0.6527	0.7588	8506
macro avg	0.6618	0.3945	0.4490	8506
weighted avg	0.8632	0.6527	0.7155	8506
samples avg	0.7582	0.6909	0.7088	8506

instance-based measures
mean precision 0.7582
mean recall 0.6909
f1 0.723

KNN CLASSIFIER

-KNN has a better performance compared to the NB classifier.

-Recall value for this model is way higher than the NB it means that the KNN finds true positives better than the NB classifier.

-Documents of less frequent classes have larger precision and recall values compared to the NB classifier.

-We can say that the KNN classifier is superior to the NB classifier for this task.

	precision	recall	f1-score	support
Treatment	0.7973	0.8197	0.8083	2207
Diagnosis	0.7062	0.7820	0.7422	1546
Prevention	0.8977	0.8647	0.8809	2750
Mechanism	0.8341	0.6980	0.7600	1073
Transmission	0.5535	0.3438	0.4241	256
Epidemic Forecasting	0.6725	0.5990	0.6336	192
Case Report	0.7718	0.3299	0.4622	482
micro avg	0.8069	0.7650	0.7854	8506
macro avg	0.7476	0.6339	0.6730	8506
weighted avg	0.8062	0.7650	0.7785	8506
samples avg	0.8096	0.7913	0.7835	8506

instance-based measures
mean precision 0.8096
mean recall 0.7913
f1 0.8003

RESEARCH

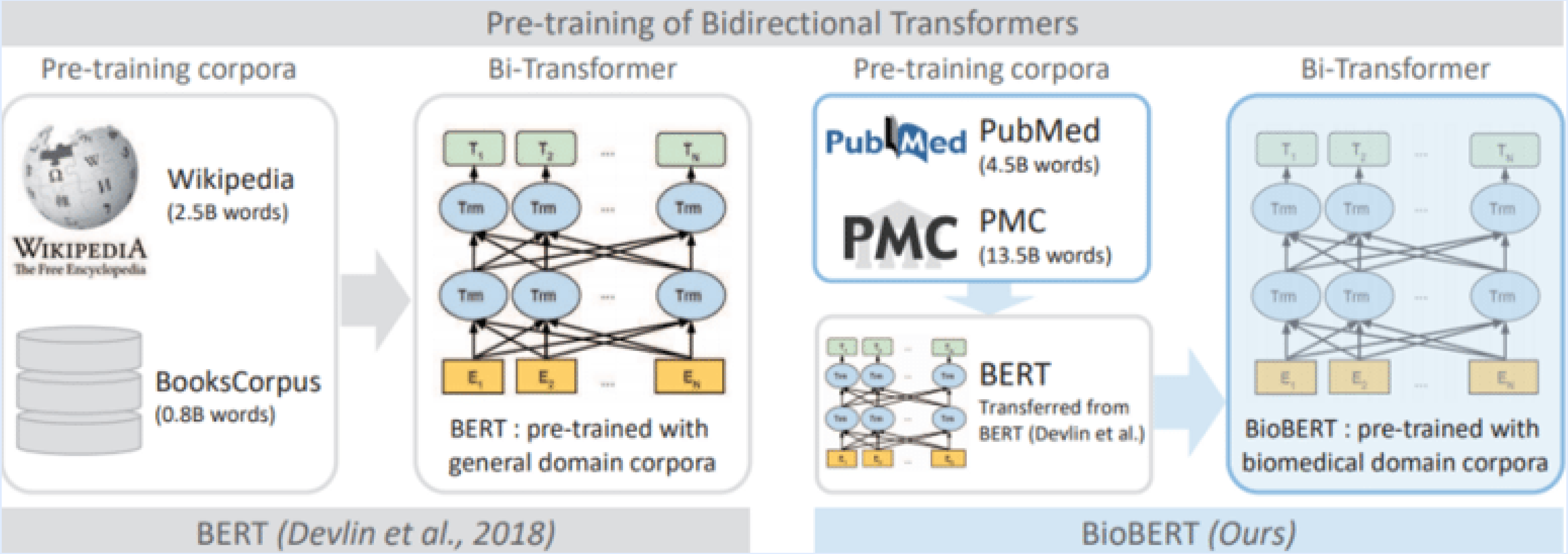
-SVM, CNN, LSTM seem to be used for text classification problems generally. Deep Learning algorithms seem to be on-trend.
-The initial plan was to create a word embedding by tf-idf or ppmi and feed a neural network, where kernels would be successive words. However, creating a suitable embedding list is a challenge.
-Hand-crafted embedding list and a CNN wouldn't be as powerful as Google's own BERT according to our research.
-BERT is a pre-trained transformer-based language model which employs deep learning. It is pre-trained by Google using a TPU for 4 days.

BIOBERT & BERT

-BERT is a transformer

-It yields compelling results in NLP

-BioBERT is domain-specific version of BERT for biomedical field.



How We Used It

-We tried alternative input types:

*Type1: Title + Abstract (we used 512 Tokens during training)

*Type2: Title only (we used 20 tokens during training)

-We used almost 20% of training data as development set.

-We tokenized the input and prepended the '[CLS]' token, which remarks it is classification, and other necessary tags like '[PAD]' to each document by using the BERT tokenizer.

-We are giving the output of the BERT model to a dropout layer with a frequency of 0.3

-We used Linear Classifier to obtain class scores.

```
self.bert_model = BertModel.from_pretrained('bert-base-uncased', return_dict=True)
self.dropout = torch.nn.Dropout(0.3)
self.linear = torch.nn.Linear(768, 7)
```

We tried various threshold values for BERT, 4 epochs

Threshold 0.1
Accuracy Score = 0.7042682926829268
F1 Score (Micro) = 0.8629856850715748
F1 Score (Macro) = 0.8187096457259796

Threshold 0.2
Accuracy Score = 0.7471116816431322
F1 Score (Micro) = 0.8780049427095036
F1 Score (Macro) = 0.8278082622179559

Threshold 0.23
Accuracy Score = 0.7546534017971759
F1 Score (Micro) = 0.8804889090086011
F1 Score (Macro) = 0.8296830553871614

Threshold 0.25
Accuracy Score = 0.7575417201540436
F1 Score (Micro) = 0.8815467728177423
F1 Score (Macro) = 0.8301670555856091

Threshold 0.27
Accuracy Score = 0.7588254172015404
F1 Score (Micro) = 0.8819293633558121
F1 Score (Macro) = 0.8296973003953034

Threshold 0.3
Accuracy Score = 0.7628369704749679
F1 Score (Micro) = 0.8828911779938999
F1 Score (Macro) = 0.8310589442939312

Threshold 0.33
Accuracy Score = 0.7650834403080873
F1 Score (Micro) = 0.8830988361994094
F1 Score (Macro) = 0.8281788187172757

Threshold 0.35
Accuracy Score = 0.7665275994865212
F1 Score (Micro) = 0.883274145659894
F1 Score (Macro) = 0.8280067411843323

BERT, title + abstract 2 epochs (30 min/epoch)

	precision	recall	f1-score	support
Treatment	0.8968	0.8675	0.8819	2204
Diagnosis	0.8403	0.8511	0.8457	1545
Prevention	0.9540	0.8992	0.9258	2747
Mechanism	0.8749	0.8618	0.8683	1071
Transmission	0.6492	0.6314	0.6402	255
Epidemic Forecasting	0.7529	0.6823	0.7158	192
Case Report	0.7856	0.9046	0.8409	482
micro avg	0.8834	0.8649	0.8740	8496
macro avg	0.8219	0.8140	0.8169	8496
weighted avg	0.8853	0.8649	0.8745	8496
samples avg	0.8975	0.8930	0.8810	8496

instance-based measures
mean precision 0.8975
mean recall 0.893
f1 0.8952

BERT, title only, 2 epochs (10 min/epoch)

	precision	recall	f1-score	support
Treatment	0.8294	0.7985	0.8137	2204
Diagnosis	0.8178	0.7178	0.7646	1545
Prevention	0.8873	0.8857	0.8865	2747
Mechanism	0.7961	0.7292	0.7612	1071
Transmission	0.7818	0.3373	0.4712	255
Epidemic Forecasting	0.8389	0.6510	0.7331	192
Case Report	0.7663	0.6598	0.7090	482
micro avg	0.8396	0.7782	0.8078	8496
macro avg	0.8168	0.6828	0.7342	8496
weighted avg	0.8370	0.7782	0.8036	8496
samples avg	0.8329	0.8129	0.8075	8496

instance-based measures
mean precision 0.8329
mean recall 0.8129
f1 0.8228

BioBERT, title + abstract only, 2 epochs

	precision	recall	f1-score	support
Treatment	0.9247	0.6906	0.7906	2204
Diagnosis	0.9019	0.7437	0.8152	1545
Prevention	0.8497	0.9614	0.9021	2747
Mechanism	0.8518	0.8049	0.8277	1071
Transmission	0.5593	0.3882	0.4583	255
Epidemic Forecasting	0.7683	0.3281	0.4599	192
Case Report	0.9393	0.5456	0.6903	482
micro avg	0.8707	0.7767	0.8210	8496
macro avg	0.8279	0.6375	0.7063	8496
weighted avg	0.8734	0.7767	0.8127	8496
samples avg	0.8587	0.8168	0.8200	8496

instance-based measures
mean precision 0.8587
mean recall 0.8168
f1 0.8372

BERT, title+abstract, 4 epochs (30 min/epoch)

	precision	recall	f1-score	support
Treatment	0.8972	0.8789	0.8879	2204
Diagnosis	0.8620	0.8447	0.8532	1545
Prevention	0.9337	0.9378	0.9357	2747
Mechanism	0.9094	0.8151	0.8597	1071
Transmission	0.8092	0.4824	0.6044	255
Epidemic Forecasting	0.8129	0.6562	0.7262	192
Case Report	0.8513	0.8672	0.8592	482
micro avg	0.8984	0.8661	0.8819	8496
macro avg	0.8679	0.7832	0.8181	8496
weighted avg	0.8970	0.8661	0.8797	8496
samples avg	0.9133	0.8970	0.8914	8496

```
instance-based measures
mean precision 0.9133
mean recall 0.897
f1 0.9051
```

BioBERT vs BERT

We expected BioBERT to have better scores than BERT due to it being specialized towards bio-medical topics. However, our results tell the opposite. BERT performed better with this data set.

Therefore, we trained BERT for 4 epochs.

Score Analysis for Classes

As we can see different classes have different precision and recall values. This occurrence is a result of unbalanced train set in terms of classes. The other reason is that some classes are correlated with each other. We came up with some guesses:
Epidemic Forecasting class has high precision and recall values although it has a low number of samples in the dataset, like Transmission class, which has low precision and recall values. The reason might be that Epidemic Forecasting class may be a class of single labelled outcomes, and Transmission may be a class of multi labelled outcomes, generally. Epidemic Forecasting is a class like Prevention so there might be multi labelled articles with these two. Also, higher frequency classes have higher scores.

There may be other important correlations among the classess as well, and there is a paper about BioBERT that includes a table for insight:

B. Label distribution

In addition, we count the distribution of labels in the training set as shown in Table I.

TABLE I. LABEL RELEVANCE DISTRIBUTION

class	Tre	Dia	Pre	Mec	Tra	Ep-Fore	Case-Re
Tre	1	0.34	0.07	0.39	0.01	0.00	0
Dia	0.48	1	0.11	0.12	0.04	0.00	0
Pre	0.06	0.06	1	0.01	0.05	0.04	0
Mec	0.77	0.16	0.04	1	0.05	0.00	0
Tra	0.12	0.26	0.56	0.23	1	0.06	0
Ep-Fore	0.01	0.01	0.65	0.01	0.09	1	0
Case-Re	0	0	0	0	0	0	1

We obtain the corresponding distribution matrix by counting the distribution of the labels in the training set. Specifically, count the total number of occurrences of the first label, M. Then count the total number of occurrences of both the first and the other label as N. N/M is the correlation value. Due to space limitations, we have abbreviated the names of each category.

Our guesses about Epidemic Forecast being an isolated class and being related to Prevention, if related to any, seem right. Case Report seems like an absolute isolated class. However its precision and recall scores are not 100%. The reason may be that the class sample size is relatively small and its vocabulary may not be unique.

Future Work

-Increasing number of epochs & further hyperparameter tuning such as learning rate.

-Creating binomial classifiers to make classification between highly correlated classes like epidemic forecasting and prevention.

-Lowering selection threshold for less frequent classes.

References

- https://github.com/yash-007/NLP-with-Deep-Learning/blob/master/BERT/Multi%20Label%20Text%20Classification%20using%20BERT%20PyTorch/bert_multilabel_pytorch_standard.ipynb
- https://towardsdatascience.com/tagging-genes-and-proteins-with-biobert-c7b04fc6eb4f
- https://colab.research.google.com/github/abhimishra91/transformers-tutorials/blob/master/transformers_multi_label_classification.ipynb#scrollTo=RhaFMQLPDqLl
- https://biocreative.bioinformatics.udel.edu/media/store/files/2021/TRACK5_pos_7_BC7_submission_143.pdf
- https://medium.com/@raghudeep/biobert-insights-b4c66fde8fa7
- https://towardsdatascience.com/text-classification-with-bert-in-pytorch-887965e5820f
- https://medium.com/technovators/machine-learning-based-multi-label-text-classification-9a0e17f88bb4
- https://scikit-learn.org/stable/modules/naive_bayes.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c
- https://biocreative.bioinformatics.udel.edu/media/store/files/2021/TRACK5_pos_7_BC7_submission_143.pdf
- https://arxiv.org/ftp/arxiv/papers/2111/2111.05808.pdf
- https://biocreative.bioinformatics.udel.edu/media/store/files/2021/TRACK5_pos_7_BC7_submission_143.pdf
- https://biocreative.bioinformatics.udel.edu/media/store/files/2021/TRACK5_pos_6_BC7_submission_140.pdf
- https://machinelearningmastery.com/best-practices-document-classification-deep-learning/
- https://monkeylearn.com/text-classification/