Report: Project 2

Machine Learning for Health Care

Remo Kellenberger, Michael Mazourik, Julian Neff

26.04.2022

1 Introduction

> Natural language processing (NLP) is a form of machine learning which enables the processing and analysis of unstructured text. When used with medical notes, it has the potential to aid in the prediction of patient outcomes, augment hospital triage systems, and generate diagnostic models that detect early-stage chronic diseases [Loc+21].

In this project, we used various different NLP techniques for multi-class classification such as sentence embeddings through term frequency inverse document frequency (TF-IDF), word embeddings with Word2Vec and transformer-based language model encoding, BERT. The repository with the code can be found here.

2 Dataset

PubMed 200k RCT is a dataset based on PubMed for sequential sentence classification. The dataset consists of approximately 200'000 abstracts of randomized controlled trials, totaling 2.3 million sentences. Each sentence of each abstract is labeled (unevenly) with their role in the abstract [DL17].

Labels	Occurrence
Objective	8.431%
Background	8.894%
Conclusions	15.35%
Result	32.68%
Methods	34.64%

Table 1: Occurrence of labels

We preprocessed the data by mapping various transformation. An essential transformation is to reduce all the sentences characters to lower case. Additionally, pre-procing by removing all punctuation and stopwords increased our predictions in all tests. The other filters we used in our tests were stemming and lemmatization. Stemming reduces infleced words to their word stem and lemmatization groups together inflected forms of a word.

In our testing we saw that lemmatization increases the prediction in most cases. This makes sense as stemming usually does not consider the context and just removes the last few characters. Lemmatization however maps words to a meaningful base form.

It also has to be noted that some samples had to be discarded because there were no words left after removing punctuation and stop-words. Regarding the transformer-based encodings, we chose to not apply any pre-processing on the sentences.

3 Baseline Model

As a baseline embedding model we made use of TF-IDF. It measures the originality of a word by comparing the number of times a certain word appears in a sample with the number of samples the word appears in.

$$TF\text{-}IDF = TF(t, d) \times IDF(t)$$

In our case, each phrase is first encoded to its corresponding TF-IDF vector. We then select the 2000 best features among all encodings. Our experiments have shown that the best results can be achieved by selecting the 2000 features according to the chi-squared test.

For our experiment, we use three different baseline models from the sklearn library. (1) **DT:** Decision Tree Classifier with a max depth of 25. (2) **KN:** K-Nearest Neighbors with 25 neighbors. (3) **MLP:** Multi layer perceptron with a single hidden layer of size 100.

Model	**Accuracy**
DT	57.6%
KN	61.7%
MLP	71.2%

Table 2: Accuracy of the baseline models

The results of the experiments are shown in table 2. The best results where achieved using the MLP classification model with an accuracy of 71.2%. As we can see in Figure 1, the model performs best for the categories Methods and Results. This could be expected as these two categories contain the most samples.

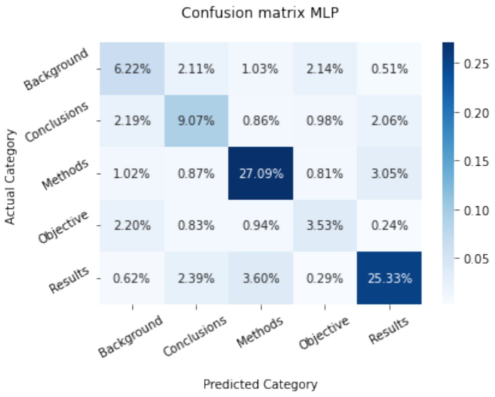

Figure 1: Confusion matrix of MLP classification

4 Word Embedding

For the word embedding task, we made use of Word2Vec [Mik+13a; Mik+13b]. It learns word associations from a large corpus of text and then tries to detect synonymous words or suggest additional words (for a partial sentence). The interesting thing about Word2Vec is that the word embeddings depend on the context that they occur. There are two model architectures: CBOW, which predicts words from a context of words and skip-gram, which uses words to predict a context. When embedding a sentence, we use the average of the word embeddings.

Model	**Accuracy**
Unigram	
DT	63%, 65%
KN	78%, 77%
MLP	80%, 80%
Bigram	
DT	65%, 64%
KN	77%, 75%
MLP	80%, 79%
Trigram	
DT	60%, 64%
KN	79%, 74%
MLP	82%, 79%

Table 3: Comparison: skip-gram (left), CBOW (right)

We embedded each phrase (without any preprocessing) to a vector of size 1000 by taking the average of the uni/bi/tri -grams.

As the focus of this task is on NLP, we use simple classifiers to also get a direct comparison with the previous task, model parameters were the standard from the sklearn library: **DT:** Decision Tree Classifier with a max depth of 25. **KN:** K-Nearest Neighbors with 25 neighbors. **MLP:** Multi layer perceptron with a single hidden layer of size 100.

After running various tests we see in Table 3 that we achieve the best results with a trigram encoding. However this is to be expected as we give the model the most context for each sample.

Lastly, we compare the different embedding models of word2vec and how they predict specific labels. We tested this with the MLP with the same settings as above and the trigram encoding. The results are shown in Table 4. We couldn't find any significant differences in the model architectures only that we overall got better results with skip-gram. One interesting insight from table 3 is that we achieved very high results (82%) when predicting ”Methods” even tough it appears only in 15.35% of samples. In comparison ”Conclusions” appears in ”34.64%” of samples but we are only able to predict it in 65% of cases. We assume that samples with the ”Methods” are more distinct from samples with ”Background”, ”Objective” or ”Conclusions” labels.

Labels	**Accuracy**
Background	60%, 58%
Objective	63%, 61%
Methods	82%, 82%
Results	86%, 85%
Conclusion	65%, 60%

Table 4: Accuracy for each class: skip-gram (left), CBOW (right)

5 Transformer Model

Language models have been introduced to utilise the vast amount of unsupervised text currently available to improve performance on tasks dealing with natural language processing. Specifically, BERT has been trained by predicting "deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context" [Dev+18]. Together with the the fact that training attention-based transformers is possible to train in a parallelizable manner has created an environment in which models achieve state of the art whilst reducing time for computation [Vas+17]. Furthermore, a specific instance of BERT called 'ClinicalBERT - Bio + Clinical BERT Model' which is a language model was trained on generic clinical text in the same self-supervised manner [Als+19]. In our attempt to classify the sentences found in the dataset, we employed a specific instance available through the open-source library hugging face [Wol+20].

Model	**Accuracy**
No-Fine Tuning	33%
Freeze Fine-tunning (6 epoch)	50%
Freeze Fine-tunning (10 epoch)	54%
Freeze Fine-tunning (15 epoch)	57%
No freeze fine-tunning (2 epoch)	35%

Table 5: BioClinical BERT: Training accuracy for the transformer-encoding based models

The fine-tuned models have two modes of training: 'Freeze' and 'No Freeze' on the language model weights. This creates a strong difference in terms of computational power as the amount of training weights changes from 3.4k to 108M for the 'Freeze' and 'No Freeze' modes respectively.

6 Appendix

References

[Als+19] E. Alsentzer, J. R. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann, and M. B. A. McDermott. "Publicly Available Clinical BERT Embeddings". *CoRR* abs/1904.03323 (2019). URL: http://arxiv.org/abs/1904.03323.

[Dev+18] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *CoRR* abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805.

[DL17] F. Dernoncourt and J. Y. Lee. "PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts". *CoRR* abs/1710.06071 (2017).

[Loc+21] S. Locke, A. Bashall, S. Al-Adely, J. Moore, A. Wilson, and G. B. Kitchen. "Natural language processing in medicine: A review". *Trends in Anaesthesia and Critical Care* 38 (2021), pp. 4–9.

[Mik+13a] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space" (2013). URL: https://arxiv.org/abs/1301.3781.

[Mik+13b] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. "Distributed representations of words and phrases and their compositionality". *Advances in Neural Information Processing Systems* (2013). URL: https://arxiv.org/abs/1310.4546.

[Vas+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention Is All You Need". *CoRR* abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762.

[Wol+20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. "Transformers: State-of-the-Art Natural Language Processing". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Online: Association for Computational Linguistics, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.