Project 2 Natural Language Processing

Anton Schäfer
ETH Zurich
Switzerland
scanton@ethz.ch

Lasse F. Wolff Anthony
ETH Zurich
Switzerland
laanthony@ethz.ch

Nidhi Agrawal
ETH Zurich
Switzerland
agrawaln@ethz.ch

ABSTRACT

It is a daunting task to keep up with medical literature. There are massive amounts of medical information and literature already available and being produced every year. Analysis and processing of this data is crucial for research purposes as well as diagnostics and treatments. Natural language processing enables computers to understand, analyze and interpret human language. In this project work, we investigate automatic classification of sentences in abstracts from biomedical literature of the PubMed dataset using NLP. We develop deep learning as well as traditional ML models and evaluate their performance on this task. Finally, we show that BERT can further improve performance.

1 INTRODUCTION

A randomized controlled trial (RCT) is a study where participants from an eligible group are assigned randomly to an experimental group or a control group. The study tests the extent to which specific, planned impacts are being achieved. RCTs provide the most compelling medical evidence. Since there are massive amounts of RCT data available, automatic sequential sentence classification of this data will be very useful in literature reviews. NLP can also be used for automating other interesting use cases such as summarizing text, information extraction such as the scientific claim of an abstract or information retrieval such as effect of a drug on a disease.

In this work, we exploit NLP with traditional machine learning and deep learning and its capabilities to recognize complex data patterns, extract features automatically, and capture long range dependencies for PubMed 200k RCT dataset. We propose various machine learning, deep learning and transformer based models and evaluate their performance on this task and compare them.

2 MODELS AND METHODS

2.1 Dataset

We evaluate our models on PubMed 200k RCT dataset [5] consisting of abstracts of biomedical literature for sequential sentence classification. This dataset consists of around 200,000 abstracts of randomized controlled trials which is around 2.3 million sentences split into 98.7% (2.24M) train and 1.3% (29.4K) test samples. The sentences are labeled according to their contribution in the abstract. The labels are: background, objective, methods, results, or conclusions [Listing 1]. The dataset is slightly imbalanced with 33% sentences corresponding to 'methods' and 35% corresponding to 'results' labels. Figure 1 shows distribution of classes, distribution of tokens per sentence after preprocessing and distribution of characters per sentence.

```
###25787999
BACKGROUND  Emotional eating in children has been   [...]
OBJECTIVE   We evaluated whether emotional eating   [...]
METHODS     Forty-one parent-child dyads were       [...]
RESULTS     Children at ages 5-7 y who were exposed [...]
RESULTS     Parents who reported the use of more    [...]
CONCLUSIONS Parents who overly control children 's  [...]
CONCLUSIONS Additional research is needed to        [...]
CONCLUSIONS This trial was registered at            [...]
```

Listing 1: Example of one abstract from the PubMed 200k RCT dataset. Listing is inspired by Dernoncourt and Lee [5].

2.2 Models

We investigate several methods for obtaining sentence embeddings with varying complexity as well as explore the performance of different deep-learning and traditional ML classifiers. In the following we outline the models that we evaluate:

Baseline: As a baseline we employ a simple logistic regression classifier with term frequency-inverse document frequency (TF-IDF) features using both word-level unigrams and bigrams (Baseline and Baseline w. lemmatization). These models have the advantage of a low computation complexity while still showing reasonable performance on many NLP tasks and as such allows us to explore the trade-off between compute required and model performance. As a preprocessing step we lowercase our tokens, and remove stop-words and punctuation. We further introduce an optional lemmatization step [Listing 2].

One hot encoding: As a second, simpler approach, we propose a OneHot, where each word is one hot encoded and the sentences are encoded as fixed-size arrays of one hot encoded words. Next, word embeddings are learned by an embedding layer which is followed by a three-layer MLP. To reduce computational cost, we limit the vocabulary size to 10k.

Word2Vec: We apply the Word2Vec algorithm [11] to the given data after preprocessing using the same steps as in the baseline, to produce word embeddings. Sentence embeddings are then generated by averaging word vectors of all the words in the sentence. We train two classifiers on the sentence embeddings:
Word2Vec MLP: We implement four dense layers with alternating Dropout layers in order to avoid overfitting.
Word2Vec LR: We apply logistic regression to the sentence embeddings to compare with the baseline.

GloVe: As an alternative to Word2Vec, we use the GloVe embedding method which is able to incorporate global statistics [13]. We follow the approach described above for Word2Vec. However, we don't

Machine Learning for Healthcare, 2022, ETH Zurich, Switzerland
May 15, 2022.

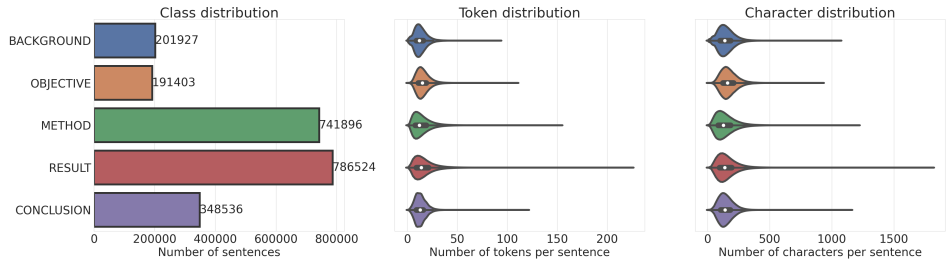

Figure 1: (Left) Distribution of classes. (Middle) Distribution of tokens per sentence after preprocessing sentences with lowercasing and removal of stop-words and punctuation. (Right) Distribution of characters per sentence.

train the GloVe embeddings on the given corpus but use Glove.6B provided by Pennington et al. [13].

BERT: BERT models pre-trained on large corpora achieve state-of-the-art results in many NLP tasks [6]. To explore whether they yield improvements in our setting, We use the Bio Clinical BERT model pre-trained on medical notes from the MIMIC III dataset by Alsentzer et al. [2, 8]. We investigate three different setups: BERT FINETUNE where we train the Bert model as well as a classification head and BERT FREEZE where we freeze the BERT model and only train the final classifier. Instead of a single layer, the BERT FREEZE classification head consists of a three-layer MLP which we find to improve performance when all previous layers are frozen. Further, to investigate whether the position/index of the sentence in the abstract

```
ORIGINAL SENTENCE:
Participants said they were not sleeping more , but
sleeping better , waking more refreshed , feeling less
distressed about insomnia , and better able to cope when
it occurred .

TOKENIZED (LOWERCASING, STOP-WORD & PUNCTUATION REMOVAL):
['participants', 'said', 'sleeping', 'sleeping', 'better',
'waking', 'refreshed', 'feeling', 'distressed', 'insomnia'
'better', 'able', 'cope', 'occurred']

TOKENIZED WITH LEMMATIZATION:
['participant', 'say', 'sleep', 'sleep', 'well',
'wake', 'refresh', 'feel', 'distressed', 'insomnia',
'well', 'able', 'cope', 'occur']
```

Listing 2: Example of a sentence before preprocessing (original), after tokenization with lowercasing and removal of stop-words and punctuation, and after tokenization with lemmatization.

3 EXPERIMENTS

3.1 Experimental Setup

All of our experiments were performed on the ETH Euler cluster[1] using 1 unknown GPU and 1 unknown CPU with 32 GB of reserved RAM.

The baseline models and the logistic regression classifier used in the word embedding model were developed using Scikit-learn [12]. Instead of employing a one-versus-rest scheme, we minimize the multinomial loss with L2 regularization for a maximum of

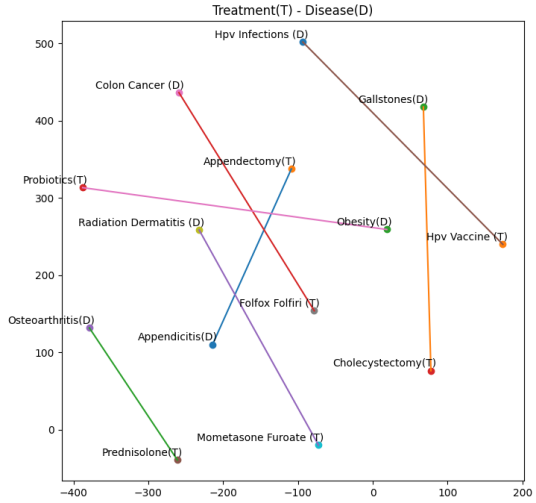

Figure 2: Possible Treatment(T) for Disease(D): Relationship visualization of learned Word2Vec embeddings using t-SNE.

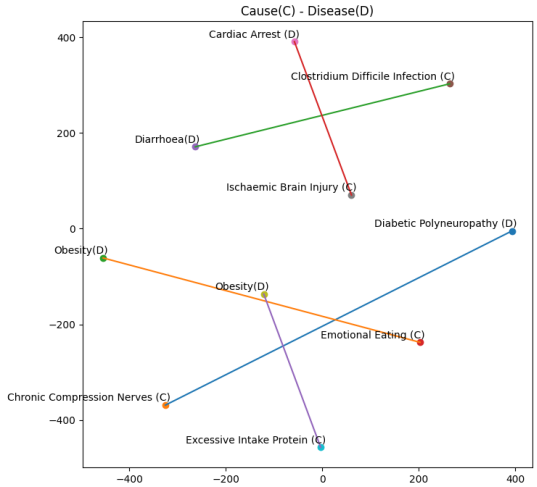

Figure 3: Possible Cause(C) of Disease(D): Relationship visualization of learned Word2Vec embeddings using t-SNE.

[1]https://scicomp.ethz.ch/wiki/Euler

Figure 4: Per-class metrics associated with our model performances on the test set. Only the seed achieving the best performance is shown. The class abbreviations are as follows: 'B' is background, 'O' is objective, 'M' is method, 'R' is result, and 'C' is conclusion.

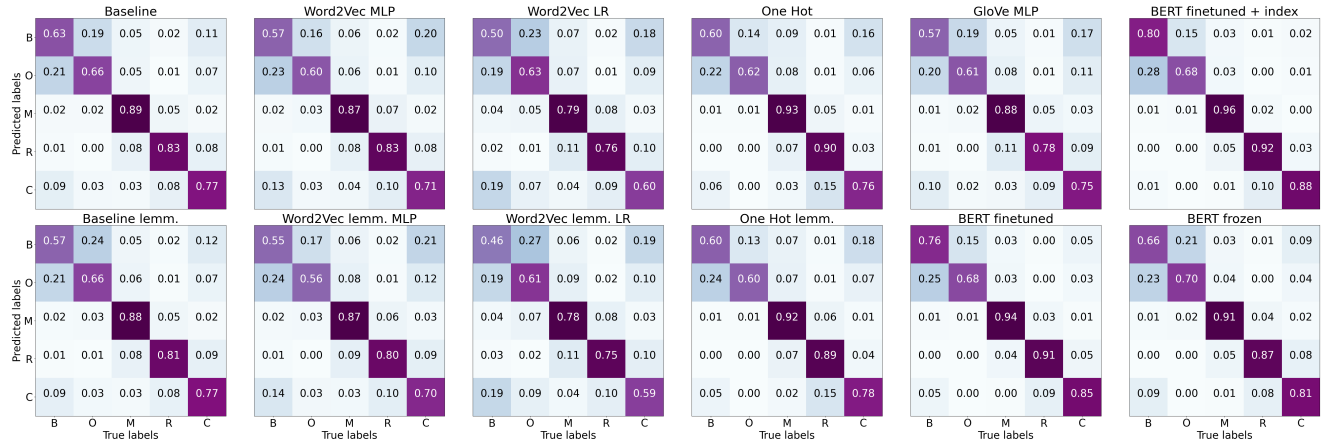

Figure 5: Confusion matrices associated with our model performances on the test set. Only the seed achieving best performance is shown. The class abbreviations are as in Figure 4

100 iterations using the SAGA optimization method [4] for faster convergence. Preprocessing was performed using spaCy[2] [7]. For the word embedding models we use the Gensim library [14] to train our word2vec model and Tensorflow [1] to develop the MLP trained on top of the word embeddings as well as the one-hot encoded model. We train the models with the Adam optimizer [9] for a maximum of 50 epochs using early stopping. Convergence was assumed when the weighted cross-entropy loss did not improve on the validation set for 3 consecutive epochs. The final models are those with the lowest validation loss.

Hyperparameters were tuned on a hold-out validation set using extensive random searches [3]. Optimal configurations can be found in the source code. Our final models were trained on the combined training and validation set to further increase performance.

The BERT-based models are trained with AdamW [10] for 20 epochs with a 12 hour timelimit. Due to the computational cost of BERT, we use common hyperparameters instead of tuning. For BERT Freeze we use a learning rate of 1e-3 while for finetuning BERT we use 3e-5.

3.2 Results and Discussion

Table 1 shows the weighted F1 scores of the proposed models. We generally observe increasing performance with increasing training time, suggesting a tradeoff between computational cost and performance. As can be seen in Figure 5 and Figure 4, the larger method and result classes are generally predicted more accurately while less frequent classes seem harder to predict and are much more frequently confused with one another.

[2]https://spacy.io/

The TFIDF Baseline performs relatively well, outperforming all word embedding methods. This is surprising given the simplicity of the method. The TFIDF features appear more useful than the more sophisticated word embeddings, especially considering that Word2Vec LR performs much worse than Baseline despite using the same logistic regression classifier. We hypothesize that the semantic information contained in the Word2Vec embeddings is not of high importance for the given task. Still, the word embedding methods produce useful results. Both, Word2Vec MLP and GloVe MLP yield F1 scores of 0.78. However, we would have expected a more significant performance difference between the domain specific Word2Vec embeddings and the general GloVe embeddings.

The learned Word2Vec embeddings seem to contain interesting relational information about concepts in the medical domain. While we observe no clear trend in Figure 3 for embeddings of diseases and their cause, Figure 2 suggests that the differences between diseases and their treatments in the vector space are generally very similar.

Compared to the unsupervised word embedding methods, the OneHot model with the directly learned embeddings performs much better, beating all non-BERT models. However, due to the larger number of parameters, it also takes longer to train. When the corpus is lemmatized the model seems to converge slightly faster and achieves similar performance. For all other methods, lemmatization only seems to hurt performance. Presumably, the word endings provide useful information for the given task.

As expected due to their good general language understanding capabilities, the BERT models outperform all other methods. BERT Frozen performs just slightly better than OneHot but the finetuning yields a significant performance improvement. BERT Finetuned + Idx achieves the best score among all models, however, due to the useful additional input, this comparison is not necessarily fair.

Model	Training time	F1
BERT Finetuned + Idx	12 h	**0.897**
BERT Finetuned	12 h	0.881
BERT Frozen	12 h	0.842
OneHot	2638 s	0.835
OneHot lemm.	1602 s	0.833
Baseline	825 s	0.812
Baseline w. lemmatization	783 s	0.798
Word2Vec MLP	248 s	0.784
GloVe MLP	283 s	0.780
Word2Vec MLP lemm.	197 s	0.771
Word2Vec LR	151 s	0.720
Word2Vec LR lemm.	**145 s**	0.706

Table 1: Model performance on the test set. The F1-score denotes the average weighted (by support) F1-score. Training time does not include training of embeddings.

4 CONCLUSION

In this work, we have investigated methods such as TF-IDF features, one-hot encodings, word embeddings based on word2vec and BERT for obtaining sentence embeddings for use in the task of sequential sentence classification. We combined these embeddings with several different classification architectures and explored their performance on the PubMed 200k RCT dataset. We showed that logistic regression using TF-IDF features presents a strong baseline performance. Unsupervised word embedding methods showed mediocre performance while a one-hot method learning embeddings directly produced very good results. Still, computationally more expensive BERT-based models were able to significantly outperform all other evaluated architectures.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323* (2019).

[3] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research* 13, 2 (2012).

[4] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems* 27 (2014).

[5] Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. *arXiv preprint arXiv:1710.06071* (2017).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[8] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[14] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.