Machine Learning for Healthcare Project 2 Report

Arda Arslan
aarslan@student.ethz.ch

Gökberk Özsoy
goezsoy@student.ethz.ch

Sebastian Müksch
smueksch@student.ethz.ch

1 INTRODUCTION

Randomized control trials (RCTs) are an invaluable source of medical evidence. As such, there is a vast literature dedicated to RCTs, which makes processing all of them a challenge in and of itself. However, gathering information for example on drug interactions in a timely matter is an undoubtedly vital procedure to ensure high quality treatments for patients. To this end, natural language processing (NLP) can be used to automate tasks such as text summarization or information retrieval. A first step for this can be the classification of sentences in a given abstract according to their purpose, for example background or results. We therefore present in this report a range of different classification techniques for the PubMed 200k RCT dataset [3]. We find that by fine-tuning a BioClinicalBert we can achieve a weighted F1-score of 0.874 on our test dataset.

2 DATASETS

We use the PubMed 200k RCT dataset for sequential sentence classification in medical abstracts [3]. PubMed 200k RCT consists of approx. 200k abstracts of randomized controlled trials, overall approx. 2.3 million sentences [3]. The sentences are classified into 5 different classes according to their role in the abstract: background, objective, method, result or conclusion, with some class imbalance. The dataset itself is divided into 3 splits: train, dev and test, where dev is used for hyperparameter tuning and model selection.

3 METHODS

3.1 Task 1: TF-IDF Embeddings

We map the corpus into term frequency–inverse document frequency (TF-IDF) features, then use these to train a classifier. Inverse document frequency is computed with respect to sentences, not abstracts.

Before we map the corpus, we apply basic preprocessing: lower-casing, punctuation and stop word removal. On top of that we investigate two other commonly applied preprocessing steps, lemmatization and normalizations. To prevent a combinatorial explosion of experiments with different preprocessing we conduct ours investigation in rounds. In every round, we apply a certain kind of preprocessing, map the corpus into TF-IDF features and train a LightGBM classifier [5] on 5 different, random 10% subsets (with replacement) from the train split. We then evaluate on the dev split.

Round 1 simply tests whether lemmatization is beneficial or not.

Round 2 consists of normalizing different token groups:

- Numbers;
- Paper IDs: token starting with “ntc”, “isrctn” or “ntr”, followed by a numbers, e.g. “nct00980850”
- Units: measurements such as “50mg” or “225min”;
- Technical Abbreviations: token with mixed alphabetic and numeric characters, e.g. “euro6” (regulation [1]), “12q14” (microdeletion syndrome [7]).

We match token to these groups using regular expressions and replace them with unified token that do not appear in the corpus, for example “paper-id-token“.

We motivate the chosen groups by their the frequencies across the 5 classes as shown in Figure 1. Baseline refers to the overall class distribution. As can be observed, paper IDs, for example, predominantly occur in the background sentences, whereas units in the result sentences. We hence hypothesize that being able to recognize these token groups increases classification performance.

Round 3 combines all normalizations of round 2 which lead, on average, to an improvement over the best weighted F1 score in round 1. Note that after round 1, we never apply lemmatization as this gave the better results. All results are summarized in Table 3 in Appendix A.

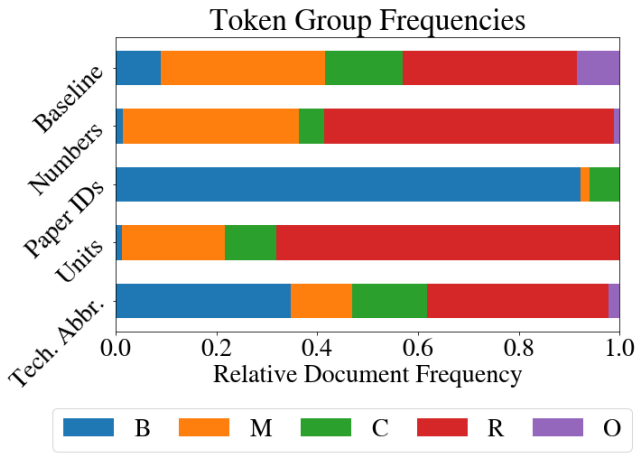

Figure 1: Token group frequencies across classes: (B)ackground, (M)ethods, (C)onclusions, (R)esults and (O)bjective.

Finally, we apply all preprocessing done in round 3 and map to TF-IDF features. Since the LightGBM classifier never selected more than 2k features based on feature importance, we elect the best 2k TF-IDF based on chi-squared statistics. This also prevents memory issues we encounter when using a large number of TF-IDF features. We then train a fully-connected neural network (FCNN) on the entire train split which yields the result in Table 1.

3.2 Task 2: Word Embeddings

The distributional hypothesis states that words in similar contexts should have similar meanings. As proposed in Word2Vec [8], this corresponds that frequent word-context embedding pairs having high inner product $\langle v_w, v_c \rangle$, while the irrelevant pairs have low.

Before training our Word2Vec model, we removed punctuation and stop words, then lower-cased and lemmatized the training corpus. Punctuation and stop words occur frequently, hence their word embeddings do not carry meaningful information. As the same word might occur in singular or plural forms and different tenses, lemmatization unifies its meaning across uses and reduces total number of pairings.

We used Gensim's Word2Vec implementation [8] with vector size 300, 5 epochs and 5 minimum word count. After obtaining word embeddings, we took the average of them per sentence to obtain sentence embeddings. For this, we did not include words that do not have an embedding vector due to the minimum word count constraint. After that, we used a FCNN with 2 hidden layers with dropouts before softmax. A comparison of different Word2Vec parameter settings can be found in Table 4 in Appendix A.

Using word embeddings in NLP tasks is both common and promising. For our task, apart from a word having meaning in the medical domain, which can be used for word analogies, see Section 5, each class is likely to have its distinct vocabulary. Hence, sentence embeddings might form a cluster per class, making the job of fully connected layers easier.

3.3 Task 3: Transformer-Based Language Models

In [10], the authors propose a neural architecture named *Transformer* which uses the so-called *attention mechanism* to weight importance of embeddings calculated in the previous layer. This architecture has proven successful in a wide variety of NLP-related tasks. When the dataset on hand is not very large, it is often beneficial to use a transformer network which was pre-trained on a dataset from a similar domain.

In our experiments, we used pre-trained *DistilBERT* [9] and *emilyalsentzer/BioClinicalBERT* [2] architectures. Due to limited computing resources, we used DistilBERT instead of BERT in our experiments. The former's size is 40% less than that of BERT [4], however, it has been shown that it maintains 97% of BERT's language understanding capacity [9].

BioClinicalBERT, on the other hand, was trained on clinical texts. Since our dataset is also from the medical domain, we expect BioClinicalBERT to provide better embeddings than DistilBERT and hence perform better on our classification task.

Each pre-trained network consists of two parts. The first part includes a word embedding layer and several transformer encoder blocks. The second part is a linear classifier. In our experiments, we only use the first part of these networks and concatenate an untrained linear classifier on top of it.

For each pre-trained network, we performed an ablation study along two dimensions: We first compared using the mean token embeddings at the last transformer layer and the special sentence embedding at the last transformer layer as input to the linear classifier. On top

we compared fine-tuning the last transformer encoder block and freezing all other weights of the transformer.

3.4 Bonus: Document Embeddings

Instead of training embeddings only for words, we can also train an embedding for each document, i.e. the sentences in the abstracts, called Doc2Vec, as proposed in [6]. The authors show that this approach provides better document embeddings compared to averaging embeddings of words in a given document, since averaging ignores word order. For the document embeddings, we again do an ablation study to explore three different parameters: document embedding sizes, window sizes and Doc2Vec training algorithm.

Weighted F1-Scores for different experiments which uses Doc2Vec document embeddings can be found in Table 5 in Appendix A. These scores are calculated on dev split.

4 EVALUATION

We present the results of our study in Table 1. For Task 3, *Fine-Tuning* refers to training the weights of the last transformer encoder block. *No Fine-Tuning* refers to freezing all transformer layers. *Mean* refers to using average of token embeddings at the last transformer layer as input to the linear classifier. *CLS* refers to using the special sentence embedding at the last transformer layer as input to the linear classifier.

For transformer and Doc2Vec experiments, we used weighted categorical cross entropy loss to deal with class imbalance. For our TF-IDF-based and Word2Vec experiments, we did not handle the class imbalance in any particular way as we found the methods to do so sufficiently by themselves.

Figure 2 presents our best model's confusion matrix across the different classes for test dataset. In Table 2, we provide the 5 most similar context words for each given word according to cosine similarity. These analogies are the natural by-product of our Word2Vec model trained for Task 2.

Appendix A provides Table 3, Table 4 Table 5 which summarize ablation studies and parameter investigations for our TF-IDF-based model preprocessing as well as Word2Vec and Doc2Vec models.

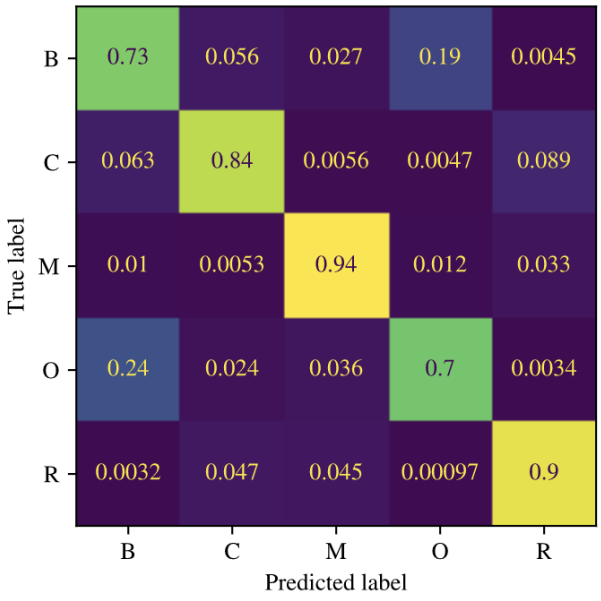

Figure 2: Confusion matrix for our best performing model's test predictions (BioClinicalBERT, Mean, Fine-Tuning). (B)ackground, (C)onclusions, (M)ethods, (O)bjective, and (R)esults.

5 DISCUSSION

As can be observed in Table 1, transformer-based models perform the best amongst our methods by a high margin, while the TF-IDF-based baseline performs the worst.

While TF-IDF yields a simple sentence embedding, a FCNN with our preprocessing still gives results close to word embeddings, with admittedly many more parameters. Nevertheless, normalizing specific token groups still appears promising given Figure 1. However, the normalizations we applied had to be based on regular expressions given our lack of domain-specific knowledge. Therefore they may be too crude. A look-up-based approach to properly categorize groups such as drugs, devices, etc. may yield better results.

In a small subset of the data, we observe that each class has a unique vocabulary conditioned by the class' role in the abstract. Hypothesizing that this holds for the entire dataset, this could be well-suited for word embeddings, as Word2Vec provides a statistical measure for word occurrences. Supporting this is that even a simple average for creating sentence embeddings seems to be enough for second place in Table 1.

Model Name		# Trainable Parameters	Weighted F1-Score
Task 1			
TF-IDF + FCNN		545 797	0.7879
Task 2			
Word2Vec + FCNN		110 597	0.7976
Task 3 - No Fine-Tuning			
BioClinicalBERT	CLS	594 437	0.8321
	Mean	594 437	0.8255
DistilBERT	CLS	594 437	0.8239
	Mean	594 437	0.8204
Task 3 - Fine-Tuning			
BioClinicalBERT	CLS	7 682 309	0.8714
	Mean	7 682 309	**0.8740**
DistilBERT	CLS	7 682 309	0.8697
	Mean	7 682 309	0.8728
Bonus			
Doc2Vec + FCNN		110 597	0.7691

Table 1: Performance comparison on test of models from all tasks. For TF-IDF, Word2Vec and Doc2Vec experiments, only number of parameters in FCNN is reported.

Word	Context
cancer	carcinoma(0.7), adenocarcinoma(0.7), carcinogenesis(0.6), carcinomas(0.6), neoplasm(0.6)
doctor	physician(0.8), gps(0.7), pediatrician(0.7), psychiatrist(0.6), practitioner(0.6)
bone	callus(0.5), ridge(0.5), bony(0.5), cartilage(0.5), osseous(0.5)
intestine	capsulorhexe(0.6), intestinal(0.6), caliber(0.6), jejunum(0.6), bore(0.5)
laboratory	hematology(0.6), chemistry(0.5), lab(0.5), haematology(0.5), urinalysis(0.5)
result	finding(0.6), lead(0.5), produce(0.5), spite(0.4), tendency(0.4)

Table 2: Most similar 5 context word (with similarity measures) for selected words.

In addition, we can analyze medical word analogies to verify that the inner working of our model is intuitive. Provided in Table 2, given the sufficiently large training set, we have highly meaningful word-context pairs, without requiring cherry-picking.

The confusion matrix demonstrates both the high performance of our best model as well as natural mistakes when attempting to differentiate the given classes. Objective and Background sentences are both general and explanatory and end up mutually confused for one another. Conclusions and Results sentences both provide summaries of findings, so the former is mistaken for the latter. On the other hand, Method sentences are full of clinical details and biological terms and the most distinctive among all 5 classes. This reflects in predictions with the highest recall.

Our transformer experiments show that using a transformer network (BioClinicalBERT) trained on a dataset which is from a similar domain our dataset yields better results than using a transformer network (DistilBERT) which is trained on a more generic data. We also observe that when fine-tuning, using mean token embeddings at the last transformer layer as input to a linear classifier yields better results than using the special sentence embedding.

6 CONCLUSION

In conclusion, we find that NLP-based methods are viable approaches to assist in processing the vast amount of RCTs literature available and incoming.

By far the best suited method in our study is the transformer-based model that is pre-trained on a similar medical domain, then fine-tuned. We find in particular its ability to recognize sentences belonging to the Methods section of the abstract promising. This could facilitate automatic information extraction and categorization of the quality of the methodology in a given RCT.

Future work entails a deep dive into further fine-tuning of transformers, for example continually unfreezing lower layers in the model once the higher stabilize, as well as addressing the noticeable confusion of Background and Objective.

REFERENCES

[1] [n. d.]. Commission Regulation (EU) No 459/2012 of 29 May 2012 amending Regulation (EC) No 715/2007 of the European Parliament and of the Council and Commission Regulation (EC) No 692/2008 as regards emissions from light passenger

and commercial vehicles (Euro 6) Text with EEA relevance. https://eur-lex.europa.eu/eli/reg/2012/459/oj. Accessed: 2022-04-25.

[2] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly Available Clinical BERT Embeddings. https://doi.org/10.48550/ARXIV.1904.03323

[3] 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).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[6] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. https://doi.org/10.48550/ARXIV.1405.4053

[7] Sally Ann Lynch, Nicola Foulds, Ann-Charlotte Thuresson, Amanda L Collins, Göran Annerén, Bernt-Oves Hedberg, Carol A Delaney, James Iremonger, Caroline M Murray, John A Crolla, et al. 2011. The 12q14 microdeletion syndrome: six new cases confirming the role of HMGA2 in growth. *European Journal of Human Genetics* 19, 5 (2011), 534–539.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. https://doi.org/10.48550/ARXIV.1301.3781

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. https://doi.org/10.48550/ARXIV.1910.01108

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. https://doi.org/10.48550/ARXIV.1706.03762

A APPENDIX

Preprocessing	Mean F1-Score ± Std.
Round 1: Lemmatization	
Lemmatization	0.7323 ± 0.0009
No Lemmatization	0.7412 ± 0.0016
Round 2: Normalizations	
Numbers	0.7445 ± 0.0006
Paper IDs	0.7430 ± 0.0014
Units	0.7409 ± 0.0013
Technical Abbreviations	0.7427 ± 0.0011
Round 3: Norm. Combination	
Numbers + Paper IDs + Tech. Abbr.	**0.7463 ± 0.0014**

Table 3: Performance of different preprocessing steps + TF-IDF embedding on dev. LightGBM trained on 5 different 10% train subsets. Best bold.

Vector Size	Window Size	Weighted F1-Score
Skip-Gram		
100	5	0.7830
	10	0.7811
300	5	0.7940
	10	**0.7958**
CBOW		
100	5	0.7785
	10	0.7751
300	5	0.7898
	10	0.7897

Table 4: Performance comparison of Word2Vec + MLP models for different settings. Scores are calculated on dev split.

Vector Size	Window Size	Weighted F1-Score
Distributed Bag of Words (DM = 0)		
100	5	0.7548
	10	0.7554
300	5	0.7629
	10	**0.7674**
Distributed Memory (DM = 1)		
100	5	0.6464
	10	0.6288
300	5	0.6775
	10	0.6558

Table 5: Performance comparison of Doc2Vec + MLP models for different settings. Scores are calculated on dev split. Best in bold.