Machine Learning for Healthcare: Project 2

Mert Ertugrul, Johan Lokna, Nora Schneider

I. Introduction

Randomized controlled trials (RCTs) are considered to be the best source of medical evidence. Hence, there is a large amount of RCT publications, roughly half of them on PubMed. This amount requires researchers to efficiently parse previous literature. Structured abstracts facilitate this process, but only half of RCTs' abstracts are explicitly structured [2]. Sentence classification can efficiently solve the problem as it can predict the structure when it is not explicitly defined. In this report we evaluate different NLP approaches for sentences labeling in RCTs' abstracts.

Our experiments are based on the PubMed RCT dataset [2]. It consists of 190654 unique abstracts where each sentence is labeled with its role (”background”, ”objective”, ”methods”, ”results”, ”conclusions”). Further, the data is relative imbalanced; the largest class contains roughly four times more sentences than the smallest.

Furthermore, our models can be bundled together into three groups. First, we test different embeddings commonly used in NLP for creating a latent representation of unstructured text and devise classifiers based on these embeddings. Second, we extend our models by providing the classifiers with structural context. Last, we try to implicitly transfer structural awareness back into our simple TF-IDF model by means of knowledge distillation. Results are summarized in table I.

A. Methodology

The data is split into train, dev and test datasets. Weighted f1-score is used for model performance comparison and hyperparameter selection. Further, due to computational restraints we often use a subset for training during the model selection phase. Moreover, we also investigate the influence of the the dataset size by training models on both the full dataset as well as the 20k subset. Further extensions based on positional information and knowledge distillation are only tested on the 20k dataset due to computational restraints.

II. Models

A. Preprocessing

For task 1 and 2, we use different preprocessed versions of the data as an input for our sentence embedding models. First we apply lowercasing, tokenization, stop-words and punctuation removal. As an additional optional step we remove the number placeholder. Last, we optionally apply lemmatization or stemming.

B. Baseline: TF-IDF

We test different TF-IDF embeddings and classifiers (naive bayes classifier, logistic regression and random forest). Logistic regression consistently performs best; there is a significant performance gap between logistic regression and random forest. Due to resource limits, we restrict depth and number of decision trees, which we suspect causes the poor performance of random forest.

Additionally, we evaluate XGBoost on a subset of the TF-IDF features (due to resource limits). Still, it performs worse than logistic regression even though training is more complex. We hypothesize that the limited number of TF-IDF features is the reason for that.

Also, we compare the performance of logistic regression using different preprocessing options. Surprisingly, the classifier performs best using the lower-cased original sentences. When analysing the most important terms for classification (see below and II), we observe that stop-words are often included in the list of high-ranked words. This offers a potential explanation for a worse performance when removing stop-words during preprocessing.

Further, we test the effect of balancing the classes in the training data. The difference in f1-score is marginal with no balancing being slightly better. When balancing, the minority classes are more often correctly classified, but the majority ones are detected more poorly. Hence, the choice of balancing depends on the use case and the respective misclassification costs.

Final Model: The final model is a logistic regression trained on a TF-IDF embedding using the $50,000$ most relevant features. Further, the TF-IDF embedding includes 1-grams and 2-grams of the original sentences. We do not include any preprocessing besides lowercasing the sentences. This model is the best performer among our implemented single sentence classification models. Detailed results are in table I. The confusion matrix in Figure 1 shows that the majority classes are mostly correctly predicted. ”Background” and ”objective” are most often confused by the model and the model has the worst performance over these two classes .

In order to further interpret the classifier, we analyse low and high weights. If terms like ”further studies”,

”we recommend” and ”further research” are relevant and terms like ”we compared”, ”was assessed” and ”tested the” are irrelevant in a sentence, then the classifier predicts ”conclusion”. Similar, ”objective” is predicted when terms like ”compare”, ”to investigate” and ”evaluate” are relevant and terms like ”results indicate”, ”conclude that” and ”results suggest” are irrelevant. These results are intuitively sensible and logistic regression seems to learn meaningful correlations between the presence or absence of a certain words and the sentence label.

C. Task 2: Word2Vec

We use the skip-gram based version of Word2Vec. Medical text contains a variety of technical and rare terms that the model may need to efficiently deal with. Skip-gram showing better performance on a corpus with large vocabulary [7] and other studies based on medical corpus favoring skip-gram [8][10], are the reasons behind our model preference.

Semantic Relationships: Given that the embedding model aims to capture semantic information from the corpus, we expect word vectors to capture semantic relationships. Analysing arbitrarily selected sample words that have the smallest cosine distance to each other, we observe that the words are intuitively similar in meaning or context. Further, we compare our embedding to the one in a reference study trained on the same dataset [8]. Investigating words similar to ”aspirin”, we observe that a majority of the words in our top 10 most similar words also appear in their results with different ordering. However, our cosine similarity scores are not as high as theirs. Further preprocessing that handles medical words and phrases with special rule based methods could improve these results.

We then observe binary semantic relationships between word vectors that are referred to as analogies. We select word pairs that have an intuitive relationship, e.g. ”breakfast” and ”morning”, and obtain their vector difference in embeddings. Then, we analyse the word closest to the obtained embedding that we get when applying the relationship vector to a new word, e.g. ”dinner”. In this case the most similar word is ”evening”. However, this type of querying may not work for all relationships due to highly technical nature of the corpus. Attempts to replicate this for medical terms are not as successful, where (”fever”,”paracetemol”) mapping to (”schizophrenia”, new word 2) can only find a relevant treatment option ”clozapine” as the 5th most similar word, preceded by related words that are not treatment options.

Finally, we obtain average sentence vectors and observe the corresponding t-SNE plot to get an intuition about whether averaging leads to meaningful clusters already. We interpret the resulting Fig.3 as follows:

- ”Results” and ”methods” sentences are both a large portion of the data and form large clusters that are more visibly separable from each other compared to other sentence types.
- ”Background”, ”objective” and ”conclusion” sentences appear to be clustering in very similar regions.

These preliminary observations are consistent with the confusion matrix results of Task 2 in Figure 4 for classes ”methods”, ”background” and ”objective”. Performance on ”results” is surprisingly lower than ”methods” despite them being the largest two classes and ”conclusion” is classified more accurately than ”result”.

Non-Sequential Classifiers: First, we develop non-sequential models that are comparable to those for the TF-IDF baseline. We obtain sentence embeddings by averaging word embeddings, so we can avoid working with high dimensional data. We expect this to be a reasonable choice as argued above with the t-SNE projection.

Paralleling Task 1, we train logistic regression and random forest classifiers. Further model types such as SVM, K Nearest Neighbours and XGBoost take too much computation time and do not perform better in our trials. Surprisingly, Word2Vec based embeddings perform significantly worse (see table I) and do not demonstrate any significant improvement for different hyperparameter combinations. This indicates that either averaged embeddings lose relevant information or these models do not have enough capacity to leverage the embeddings.

Sequential Classifiers: Given the poor performance of aggregated embeddings, we explore using word embeddings in a sequential manner. Bidirectional LSTMs are commonly used in combination with Word2Ved embeddings [5]. We compared different layers to further process the LSTM hidden state output, namely: only full connected layers, one dimensional convolutions with attention inspired by previous research [5], or another Bidirectional LSTM and fully connected layers. Stacked Bidirectional LSTMs do not produce better convergence and are slower, hence they are discarded. The convolutional version does not improve on the the initial version with only fully connected layers. Therefore, Bidirectional LSTM followed by fully connected layers is the final best performing model, with both the test set f1 score and accuracy being 0.01 lower than the best TF-IDF model. Further model designs such as concatenating and padding word embeddings and applying 2D convolutons on them or using differences of neighboring average sentence vectors were either infeasible to train or did not converge.

We infer that sequential classifiers are a more suitable choice for the task than non-sequential ones, albeit not surpassing TF-IDF based models. We suspect that the relevant information for discriminating between classes is based primarily on word frequencies or specific words that appear in certain parts of a research paper abstract rather than semantic information that is derived from relationships between words. When we inspect the confusion matrices, performance distribution among classes is significantly different than TF-IDF based models. While TF-IDF most likely confuses ”background” and ”objective”, word2vec confuses ”background” and ”results”.

D. Task 3: BERT

BERT [3] is one of the most widely recognized transformer-based models [13], long considered state-of-the-art for text analysis tasks such as sequence classification. Following the model’s initial success, BERT-based models have been pre-trained on a wide variety of general-purpose as well as domain-specific corpora; many of whom have been made available through the Huggingface-framework [14]. As recent work indicates that using domain-specific models results in substantial gains over their general-domain counterparts, we decided to use the pre-trained model from Gu et al. which was specifically adapted to medical literature [4]. As a reference, we also consider the performance of the original general-purpose model from Devlin et al. [3].

Implementation: Our implementation follows a classical workflow for adapting a pre-trained BERT-based model to a sequence classification task. Specifically, we use the same pre-processing as the original authors; for both models this results in lower casing the sentences and applying the predefined token mapping. Furthermore, the pre-trained BERT models are used as sentence-wise feature extractors with a standard neural network classifier on top. As indicated by the task, we test both keeping the BERT-models unchanged as well fine-tuning them. Lastly, we optimize hyperparameters using the Optuna-framework [1] with respect to f1-score over the full dev dataset. However, due to computational restraints we use a sub-sampled training set during the model selection phase in order to explore a wider range of hyper parameters.

Results: We make several interesting observations from our results. Most importantly, our BERT-based models generally outperform any other model which only considers textual information. We suspect this is a result of BERT’s significant syntactic, semantic, and world knowledge [9]; this allows for informative embeddings from which the classifier can efficiently learn. Moreover, we observe a small but consistent increase in performance when using the full dataset compared to the 20k-subset. Still, as using a ten-fold more data only increases the f1-score by roughly 1%, it seems like model choice is more important than dataset size for this task. Even more significant is the impact of fine-tuning the embedding. Fine-tuning increases performance by close to 2% for the domain-specific model and more than 9% for the general-purpose one. This shows that although the pre-trained BERT-embedding is already quite informative fine-tuning and therefore adjusting to dataspecific properties further boosts performance. Lastly, in correspondence with previous results [4], we observe a significant performance increase when using the domain-specific model. Moreover, general purpose models seem to be require better fine-tuning than domain-specific models.

E. Integrating Structure

Relative position of a sentence: As a sentence’s function in an abstract is often strongly correlated with its internal position, we conjecture that providing structural information to the classifier might improve its performance. One method for giving structural context to a sentence is through its relative_linenumber.

$$\text{relative\_linenumber} = \frac{\text{index}}{\text{\# sentences in abstract}}$$

A logistic regression model exclusively relying on this feature is able to attain an f1-score of 0.68; significantly better than random guessing which has an f1-score of 0.18. These results indicate that our previous models might benefit notably if we include this feature.

Adding the line number in TF-IDF based logistic regression and different Word2Vec based models significantly boosts their performance (see I). Hence, adding structural information in a simple way already shows promising results, motivating further investigations of it. Note, that due to resource limits, we are not able to test incorporating relative_linenumber in BERT models.

Hierarchical abstract model: Following the significant improvements through the inclusion of structural information expressed through relative_linenumber, we try to capture even more complex relations by training a classifier on full abstracts. Similar to Jin and Szolovits we device a hierarchical model [6]. Here, we first create sentence-wise representations based on our previously trained models. For each abstract, the sequence of sentence embeddings is then forwarded to a bi-directional LSTM with a neural network classifier on top which directly predicts all labels within the abstract. Again, hyperparameters such as the width and depth of our model is optimized with the Optuna-framework [1].

We test this hierarchical approach using two different sentence representations. The first approach is to use the embedding from our top-performing, fine-tuned BERT-model. The second method is to create a sentence representation based on our learned Word2Vec-embedding. Furthermore, as Word2Vec operates on a token level, we use a bi-directional LSTM to create a representation of the full sentence.

Using this approach we obtain our two top-performing models. The BERT-based hierarchical model is our overall best performing classifier with an f1-score of 0.9140; this is a noteworthy improvement over the original model which disregards any structural context. We also observe a similar increase in performance for the Word2Vec-based model. Moreover, the hierarchical Word2Vec-model significantly outperforms its counterpart which only uses relative_linenumber for structural context. While abstracts often exhibit complex structural relationships, such as dependencies between the labels of neighbouring sentences, such correlations cannot always be captured through a simple parameter such as relative_linenumber. Hence, due to their high performance, we theorize that our hierarchical models are able to better leverage dependencies between neighbouring sentences in order to increase their overall performance.

F. *Knowledge distillation*

Inspired by our good results when providing the classifiers with explicit structural context, we test if these insights can be transferred to our original structurally unaware model based on TF-IDF by means of knowledge distillation (KD). This technique has previously been successfully used to increase the performance of simple NLP-models by transmitting knowledge from a more complex one [11, 12]. KD uses a well-performing teacher-model's predictions to regularize a student model during training; we employ the BERT-based hierarchical model as teacher. The new loss function is then a weighted average between the cross entropy loss and the KL-divergence between the student's and teacher's predictions.

Although we observe a slight increase of 0.5% in performance (see Ib), the change is not very significant. Therefore it seems that KD is to some degree able to act as a regularizer. However, we are not able to efficiently distil implicit structural awareness back into our original model. Therefore, in order to take advantage of the structured nature of an abstract, it seems like the structural context has to be explicitly provided.

G. *Discussion*

We test and evaluate different state of the art NLP methods for classifying sentences in RCTs' abstracts. Using a TF-IDF sentence embedding outperform any sequential and non-sequential classifier trained on a Word2Vec embedding. This high performance from TF-IDF is especially remarkable as the Word2Vec-embedder and corresponding classifiers are generally more complex. Fine-tuned BERT sentence embeddings achieve the highest performance for non-structured models. In the given classification task, more complex models and embeddings do not necessarily result in higher performance, which we attribute to the fact that the presence or absence of certain terms in a sentence seems sufficient to achieve a descent performance already.

Further, in the sentence classification task structure context carries valuable information. Including a simple positional feature boosts the performance already. Using a more complex hierarchical abstract model with BERT achieves highest performance among all evaluated models. This demonstrates the value of contextual information for interpreting and labeling sentences.

Last, knowledge distillation can improve simple models slightly. However, it seems that structural context has to be given explicitly for achieving the best performance.

Model	**F1-Score**
TF-IDF	0.8462
TF-IDF + line_number	**0.8931**
Word2Vec (sequential)	0.8365
Word2Vec + line_number (sequential)	0.8927
Word2Vec (non-sequential)	0.7508
Word2Vec + line_number (non-sequential)	0.8616
PubMed BERT no fine-tuning	0.8688
PubMed BERT with fine-tuning	0.8810

(a) Model Performances on full PubMed dataset

Model	**F1-Score**
TF-IDF	0.8176
TF-IDF + line_number	0.8591
TF-IDF + KD	0.8221
Word2Vec	0.7968
Word2Vec + line_number	0.8618
PubMed BERT no fine-tuning	0.8566
PubMed BERT with fine-tuning	0.8753
General BERT no fine-tuning	0.7426
General BERT with fine-tuning	0.8352
Hierarchical with BERT	**0.9140**
Hierarchical with Word2Vec	0.9047

(b) Model Performances on 20k-subset of PubMed dataset

TABLE I: Model Performances; the largest value for each column is highlighted.

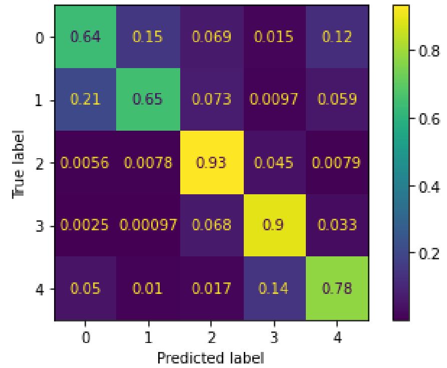


Fig. 1: Confusion matrix for TF-IDF + logistic regression (0: ”background”, 1: ”objective”, 2: ”methods”, 3: ”results”, 4: ”conclusions”)

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[2] Franck Dernoncourt and Ji Young Lee. Pubmed 200k RCT: a dataset for sequential sentence classification in medical abstracts. *CoRR*, abs/1710.06071, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[4] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing, 2020.

[5] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang, and Jong Wook Kim. Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, 10(17), 2020.

[6] Di Jin and Peter Szolovits. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts, 2018.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[8] José Antonio Miñarro-Giménez, Oscar Marín-Alonso, and Matthias Samwald. Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation. *CoRR*, abs/1502.03682, 2015.

[9] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works, 2020.

[10] Susan Sabra and Vian Sabeeh. A comparative study of n-gram and skip-gram for clinical concepts extraction. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 807–812, 2020.

[11] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019.

[12] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2019.

III. Appendix

Background - HW	’is’, ’are’, ’have’, ’has’, ’was to’, ’aimed to’, ’often’, ’study was’, ’we’, ’however’
Background - LW	’our data’, ’were’, ’we conclude’, ’was and’, ’we recommend’, ’results indicate’, ’study shows’, ’these data’, ’results suggest’, ’results support’
Objective - HW	’to compare’, ’compare’, ’to study’, ’to’, ’whether’, ’evaluate’, ’to evaluate’, ’to investigate’, ’to test’, ’to determine’
Objective - LW	’however’, ’improved’, ’can’, ’significantly’, ’showed’, ’reduces’, ’improves’, ’decreased’, ’may’, ’increased’
Methods - HW	’were compared’, ’provides class’, ’is registered’, ’were’, ’we used’, ’was defined’, ’study is’, ’less than’, ’was injected’, ’was compared’
Methods - LW	’however’, ’improved’, ’can’, ’significantly’, ’showed’, ’reduces’, ’improves’, ’decreased’, ’may’, ’increased’
Results - HW	’showed’, ’revealed’, ’was’, ’ci’, ’vs’, ’were and’, ’significantly’, ’were’, ’was and’, ’in of’
Results - LW	’is’, ’are’, ’is associated’, ’has’, ’compare’, ’were compared’, ’to compare’, ’may’, ’hypothesized that’, ’reduces’
Conclusions - HW	’further studies’, ’we recommend’, ’should’, ’these data’, ’improves’, ’improved’, ’we suggest’, ’studies are’, ’further research’, ’study suggests’
Conclusions - LW	’we compared’, ’was to’, ’compared the’, ’were compared’, ’was compared’, ’was assessed’, ’tested the’, ’to compare’, ’study compared’, ’evaluated the’

TABLE II: Terms corresponding to ten highest weights (HW) and ten lowest weights (LW) in logistic regression for each class.

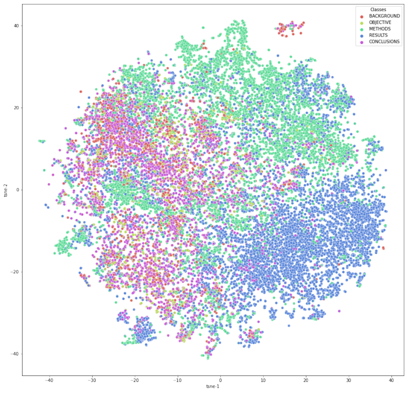


Fig. 2: Visualizing average sentence vectors with 2D t-SNE

1) *Word2Vec: Additional Visualisations:*

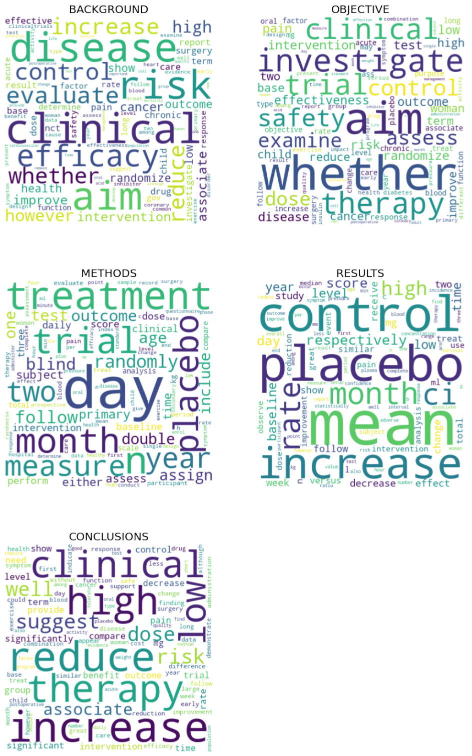
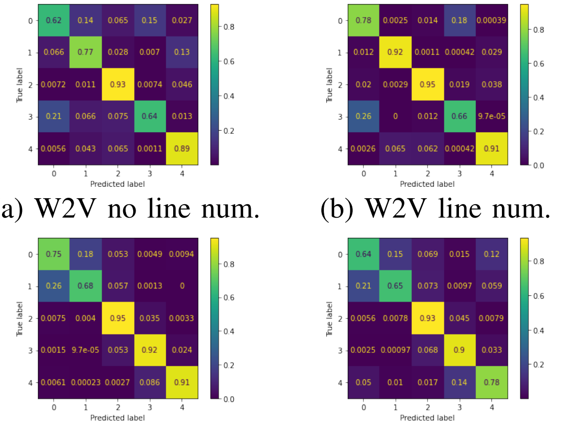


Fig. 3: Word clouds for each sentence type



(a) W2V no line num.

(b) W2V line num.

(c) TF-IDF no line num.

(d) TF-IDF line num.

Fig. 4: Confusion Matrices - increases from dark blue to yellow