Joint Sentence Classification of Medical Paper Abstracts with Neural Networks and Syntactic Features

Carlos Sathler
School of Informatics, Computing, and Engineering
Indiana University at Bloomington
csathler@iu.edu

Abstract

A new dataset with Randomized Controlled Trial (RCT) research paper abstracts has been made available recently to Data Scientists. The dataset holds a collection of labeled abstract sentences. It can be used to train predictive models to help researchers more quickly peruse research literature. The creators of this dataset used state of the art models to establish a performance benchmark for joint sentence classification on the dataset. In my research I replicate their best result and attempt to improve it by adding syntactic information to the feature set used in their best model. The syntactic features I explore are part of speech (POS) and constituent parse tree. Experimentation with different recurrent and convolutional neural network architectures on the new features set did not boost performance of the classifier. The syntactic text features I researched were not useful in improving performance of state of the art models for sentence classification of RCT abstract sentences, in the researched dataset.

1 Introduction

Dernoncourt and Lee report that the number of RCT publications has increased significantly since 1980, but note that a larger percentage of paper abstracts reveal structured content (Dernoncourt and Lee, 2017, p. 1). Clearly, on the one hand searching the literature by accessing paper abstracts has become more challenging, since there are more publications. On the other hand, structured abstracts now allow researchers to narrow their focus on certain abstracts sections of interest, for example, sections describing research objectives, or results.

Researchers would benefit from predictive models that could classify RCT abstract sentences according to useful categorization. It would save them time to be able to search selectively for specific abstract sections.

With the intent of improving predictive models for sentence classification of RCT papers, Dernoncourt and Lee released the dataset "PubMed 200K RCT" (Dernoncourt and Lee, 2017). Accompanying the release of the dataset the authors published performance benchmarks for sentence classification using state of the art models with artificial neural networks.

The best models for RCT sentence classification take into account sentence context. It is reasonable to expect that distributional factors play a significant part in sentence classification of RCT abstract sentences. Particularly for structured documents, it seems natural that authors would place sentences addressing results and conclusions of their studies towards the end of abstracts, while sentences addressing background and hypothesis would be placed likely in the beginning. The use of context in sentence classification will be referenced in my research as "joint" sentence classification.

My research explores the impact of including syntactical information in the feature set used by the best performing model identified by (Dernoncourt and Lee, 2017). Using spaCy (Honnibal and Montani, 2017) and Stanford's CoreNLP (Manning et al., 2014) I extract part of speech (POS) and constituent parse tree tags for each abstract sentence.

The syntactic features are incorporated to abstract sentence representations obtained from word embeddings. Minimal modifications were made to the best model architecture referenced in (Dernoncourt and Lee, 2017).

Using accuracy and F1-score as evaluation metrics, results show that both POS and constituent parse tree features slightly degrade performance of models trained on word embeddings alone.

2 Related Work

In their paper "Complex Linguistic Features for Text Classification: a comprehensive study" (Moschitti and Basili, 2004), describe extensive research measuring the performance impact of complex NLP features to document classification. They include POS tag in their analysis and find no improvements to the performance of SVM models and others. (Kim et al., 2011) focus their efforts on sentence classification. They experiment with a number of different complex NLP feature extraction schemes on RCF models. They report some measure of success on sentence representation that includes POS tags.

The usefulness of syntactic features for text classification is confirmed by (Johansson and Moschitti, 2010) in subjectivity analysis, specifically detection of opinionated text.

Sentence classifiers that account for sentence context are discussed in (Kim et al., 2011) and in (Lee and Dernoncourt, 2016). These authors report improved performance in sentence classification tasks when previous sentence information is used. The approach is extended by (Dernoncourt et al., 2016) who account for the full sentence context in their models, i.e., both preceding and succeeding sentences. In "PubMed 200k RCT: a dataset for Sequential Sentence Classification in Medical Abstracts" (Dernoncourt and Lee, 2017) test a number of state of the art sentence classification models on the PubMed 200k RCT dataset. Their best model uses sentence representation based on character and word embeddings, bi-directional long short term memory networks and RCF.

3 Dataset

The PubMed 200k RCT dataset (Dernoncourt and Lee, 2017) is available in 2 versions. For my research I used the reduced version, which contains 20k abstracts, as opposed to 200k. The dataset is delivered in 3 partitions: train, dev (validation) and test. Dataset stats are provided below (Table 1). Note that sentences are grouped by abstract, and are available in sorted order per their sequence in the original abstract.

Data	**|V|**	**Train**	**Valid.**	**Test**
Abstract	68k	15k	2.5k	2.5k
Sentence		180k	30k	30k

Table 1: PubMed 20k RCT Stats.

Sentences are classified in 5 classes: background, objective, methods, results, and conclusion. Label distribution in the dataset is shown in Figure 1.

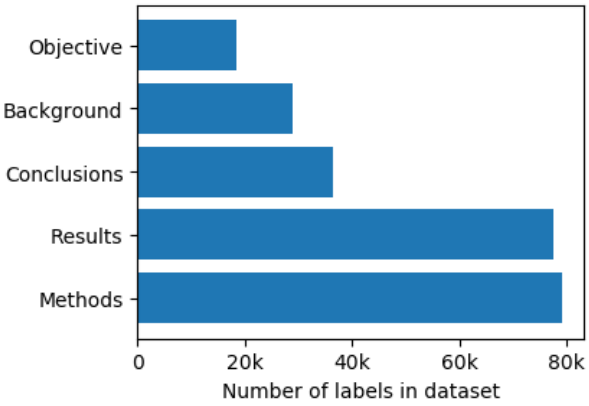

Figure 1: Label distribution.

Figure 2 shows the distribution of number of tokens per sentence. This distribution is important because sentence classification requires formatting sentences to match an arbitrarily chosen fixed length. Some sentences will be truncated if they are longer and some will be padded if they are shorter than the chosen length. For this dataset I used sentence length of 50 in my research.

Figure 3 shows the number of sentences per abstract. The distribution of sentence count is important because joint sentence classification requires formatting abstracts to match an arbitrarily chosen fixed length. In my research I selected length equal to 31, the maximum length found in the dataset. Therefore, no abstract was truncated.

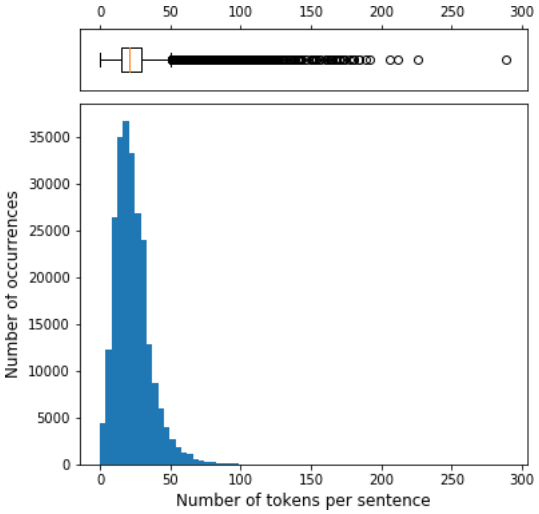

Figure 2: Tokens per sentence.

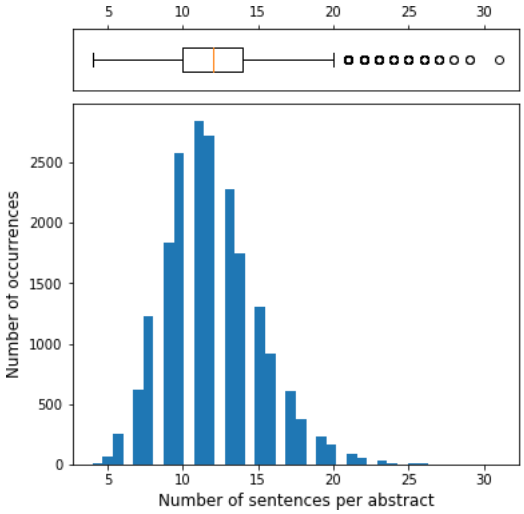

Figure 3: Sentences per abstract.

4 Approach

With the release of PubMed 200k RCT, (Dernoncourt and Lee, 2017) established a number of performance benchmarks for prediction on the PubMed 200k dataset. The model with the best performance on the dataset is described by (Dernoncourt et al., 2016). This model, called "bi-ANN" by the authors, consists of an artificial neural network with "a token embedding layer (bi-LSTM), a sentence label prediction layer (bi-LSTM), and a label sequence optimization layer (CRF)" (Dernoncourt and Lee, 2017, p. 4). All the models I developed in my research are adaptations of bi-ANN.

4.1 Benchmarking

My benchmark model is a simplified version of bi-ANN, in that it does not employ a conditional random field (CRF) step for sequence optimization at the end. The benchmark model can be summarized as follows:

1. Perform sentence classification (Figure 4) using word embeddings on LSTM followed by MLP neural net.

2. Perform prediction using Step 1 model and use output from hidden layer to create sentence representations.

3. Perform joint sentence classification (Figure 5) using sentence representations obtained from Step 2 on bi-LSTM neural net.

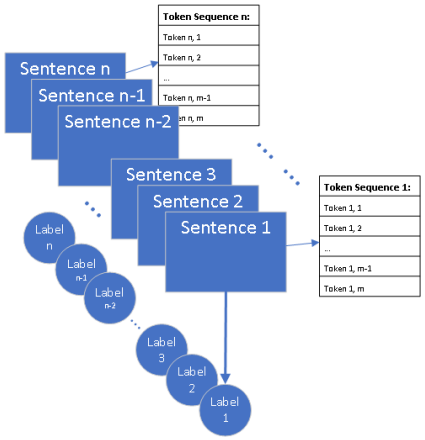

Figure 4: Sentence classification.

For Step 1 I used Glove (Pennington et al., 2014) word embeddings of dimension 200, trained on Twitter. Sentence representations obtained in Step 2 have dimension 100. The model for Step 1 was LSTM, rather than bi-LSTM. Model for Step 3 was bi-LSTM. **Benchmark: *F1-Score = 0.9097***

4.2 Model Extensions

Model extensions consisted mainly in adding NLP syntactical features, specifically POS tags and constituent parse tree tags in bracketed format, to the sentence representations fed to the joint sentence classifier. The general architecture of the benchmark

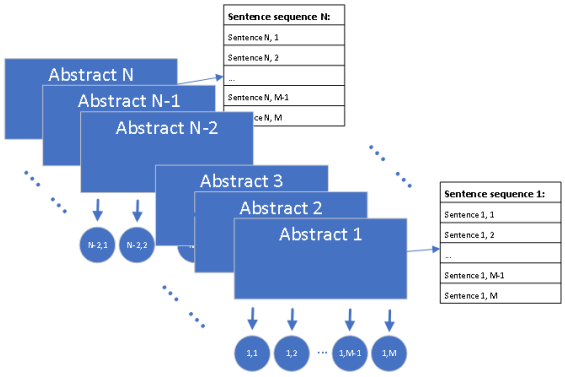

Figure 5: Joint sentence classification.

model was preserved, to allow like-to-like comparison of performance results with and without NLP features.

The following summarizes the approach:

1. Perform sentence classification using word embeddings on LSTM followed by MLP neural net. (Same as benchmark).

2. Perform sentence classification using one-hot-encoded NLP syntactic features extracted from text.

3. Perform predictions for Steps 1 and 2 and use outputs from hidden layers to create combined sentence representations (Figure 6)

4. Perform joint sentence classification using combined sentence representations obtained from Step 3.

4.3 NLP Feature extraction

For extraction of POS tag features I used spaCy (Honnibal and Montani, 2017). For extraction of constituent parse tree tags I used Stanford CoreNLP (Manning et al., 2014). The parse tree was flattened and expressed in bracketed format. Tokens were removed, only leaving parentheses and POS tags. When referring in this report to constituent parse tree ”tags”, I mean both POS tags and parentheses. Both POS tags and bracketed constituent parse trees were ”one-hot-encoded” for input to the neural networks.

5 Experiments

For both sentence and joint sentence classification I experimented with long short term memory recurrent networks (LSTM), bi-directional LSTM (bi-LSTM) and 1 dimension convolutional networks (CNN-1D). I quickly was able to replicate the F1-score reported by (Dernoncourt et al., 2016) on PubMed 20k RCT (even without CRF).

After that, my experiments focused on the different ways to produce a combined sentence representation. As stated above, Step 3 in the extended models combines sentence representations produced in Step 1 (semantic information) and Step 2 (syntactic information). Both steps produce sentence representations of dimension 100. I experimented with combined representations obtained through stacking (horizontally), and weighted averages. My best model used a horizontally stacked combined representation of dimension 200.

I experimented with different network depths, number of nodes, batch sizes, regularization parameter values, and dropout percentages. Variations did not results in significant delta performance of extended model relative to benchmark.

Loss function used during training was Keras’ (Chollet and others, 2015) ”binary-crossentropy”. Optimizer was Adam. Activation functions were ”relu” for perceptron layers, ”tanh” and ”hard sigmoid” for LSTM layers and ”softmax” for output layer.

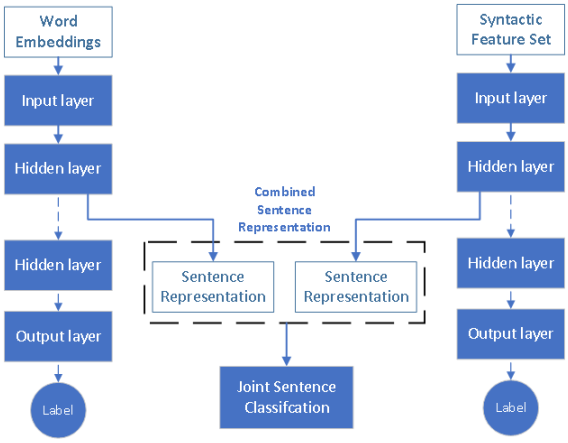

Figure 6: Joint sentence Representation.

6 Results

Table 2 below shows the joint classification results on the *validation* partition. Accuracy is the performance metric used to express the results (last column).

	Model	Feature Set	Acc.
1	bi-LSTM, MLP	Word embeddings	0.8400
2	LSTM, MLP	Word embeddings	0.8394
3	bi-LSTM, MLP	POS tags	0.6712
4	CNN 1D	Const. tree tags	0.7631
5	CNN 1D	POS tags	0.6902
6	bi-LSTM, MLP	Const. tree tags	0.7533
7	bi-LSTM, MLP	POS tags	0.6690

Table 2: Sentence Classification Results

Table 3 below shows the joint sentence classification results on the *test* partition. F1-Score is the performance metric used to express the results (last column).

The benchmark F1-score (bold) is the best score. This score is slightly superior to the F1-score = 0.8990 reported by (Dernoncourt et al., 2016) on the same dataset.

Model	Feature Set	F1
bi-LSTM, MLP	Word embeddings	0.9032
LSTM, MLP		**0.9097**
LSTM, MLP, CNN 1D, and LSTM (joint classification)	Word embeddings + POS tags	0.9010
	Word embeddings + Const. tree tags	0.9062

Table 3: Joint Sentence Classification Results

7 Conclusion

In my research I combined syntactic features extracted from PubMed 20k RCT abstract sentences with semantic features obtained from word embeddings. The syntactic features I explored, POS tags and constituent parse tree, were found to have some predictive power of their own. For example, constituent tree tags yielded accuracy = 0.7631 for sentence classification on the validation partition of the dataset, with CNN 1D. However, this predictive power did not boost performance attained with word embeddings alone. In fact, when combined with sentence representations produced from word embeddings, syntactic features were found to slightly degrade the performance of the joint classifier.

The best F1-score obtained with combined features on the test dataset was 0.9062 vs. 0.9097 for the benchmark. POS tag yielded worse performance than constituent parse tree for both sentence and joint sentence classifications. Given the intensive processing required to extract constituent parse tree tags, and given these results, I would not recommend further research to assess the impact of adding either of these features to sentence classification on PubMed 20k RCT or PubMed 200k RCT datasets.

References

Chollet, François and others. 2015. *Keras*. `https://keras.io`.

Jeffrey Pennington and Richard Socher and Christopher D. Manning. 2014. *GloVe: Global Vectors for Word Representation*. Empirical Methods in Natural Language Processing (EMNLP), 1532–1543.

Lee, Ji Young and Dernoncourt, Franck. 2016. *Sequential short-text classification with recurrent and convolutional neural networks*. arXiv preprint arXiv:1603.0382.

Kim, Su Nam and Martinez, David and Cavedon, Lawrence and Yencken, Lars. 2011. *Automatic classification of sentences to support evidence based medicine* in BMC bioinformatics (Vol. 12, No. 2, p. S5). BioMed Central.

Honnibal, Matthew AND Montani, Ines. 2017. *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. To appear.

Johansson, Richard and Moschitti, Alessandro. 2010. *Syntactic and semantic structure for opinion expression detection*. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning (pp. 67-76). Association for Computational Linguistics.

Moschitti, Alessandro and Basili, Roberto. 2004. *Complex linguistic features for text classification: A comprehensive study*. European Conference on Information Retrieval (pp. 181-196). Springer.

Dernoncourt, Franck and Lee, Ji Young and Szolovits, Peter. 2016. *Neural Networks for Joint Sentence Classification in Medical Paper Abstracts*. arXiv preprint arXiv:1612.05251.

Manning, Christopher D. and Surdeanu, Mihai and Bauer, John and Finkel, Jenny and Bethard, Steven J. and McClosky, David. 2014. *The Stanford*

CoreNLP Natural Language Processing Toolkit. In Association for Computational Linguistics (ACL) System Demonstrations (pp. 55-60). http://www.aclweb.org/anthology/P/P14/P14-5010.

Dernoncourt, Franck and Lee, Ji Young. 2017. *PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts*. arXiv preprint arXiv:1710.06071.

Evidence Based Medicine Working Group and others 1992. Evidence based medicine. A new approach to teaching the practice of medicine. *Jama, 268(17), 2420.*.