Hierarchical Neural Networks for Sequential Sentence Classification in Medical Scientific Abstracts

Di Jin
MIT
jindi15@mit.edu

Peter Szolovits
MIT
psz@mit.edu

Abstract

Prevalent models based on artificial neural network (ANN) for sentence classification often classify sentences in isolation without considering the context in which sentences appear. This hampers the traditional sentence classification approaches to the problem of sequential sentence classification, where structured prediction is needed for better overall classification performance. In this work, we present a hierarchical sequential labeling network to make use of the contextual information within surrounding sentences to help classify the current sentence. Our model outperforms the state-of-the-art results by 2%-3% on two benchmarking datasets for sequential sentence classification in medical scientific abstracts.

1 Introduction

Since 1665, over 50 million scholarly research articles have been published (Jinha, 2010), with approximately 2.5 million new scientific papers coming out each year (Ware and Mabe, 2015). While this enormous corpus provides us with the ability to conclusively accept or reject hypotheses and yields insight into promising research directions, it is getting harder and harder to extract useful information from the literature in an efficient and timely manner due to its sheer amount. Therefore, an automatic and intelligent tool to help users locate the information of interest quickly and comprehensively is highly desired.

When searching for relevant literature for a certain field, investigators first check the abstracts of scientific papers to see whether they match the criterion of interest. This process can be expedited if the abstracts are structured; that is, if the rhetorical structural elements of scientific abstracts such as *purpose, methods, results, and conclusions* (American National Standards Institute, 1979) are explicitly stated. However, even today, a significant portion of scientific abstracts is still unstructured, which causes great difficulty in information retrieval. In this paper, we develop a machine-learning based approach to automatically categorize sentences in scientific abstracts into rhetorical sections so that the desired information can be efficiently retrieved.

In a scientific abstract, each sentence can be assigned to a rhetorical structural element sequentially. This rhetorical structure profiling process can be formulated as a sequential sentence classification task, as the element assignment of any single sentence is greatly associated with the assignments of the surrounding sentences. This is in contrast to the general sentence classification problem, where each sentence is classified individually and no contextual information can be used. Previous state-of-the-art methods relied on Conditional Random Fields (CRFs) to take into account the inter-dependence between subsequent labels, which improved joint sentence classification performance by considering the label sequence information. In this work, we add a bi-directional long short-term memory (bi-LSTM) layer over the representations of individual sentences so that it can encode the contextual content and semantics from preceding and succeeding sentences for better categorical inference of the current one.

In this work, we present a hierarchical neural network model for the sequential sentence classification task, which we call a hierarchical sequential labeling network (HSLN). Our model first uses a RNN or CNN layer to individually encode the sentence representation from the sequence of word embeddings, then uses another bi-LSTM layer to take as input the individual sentence representation and output the contextualized sentence representation, subsequently uses a single-hidden-layer feed-forward network to transform the sentence

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3100–3109
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

representation to the probability vector, and finally optimizes the predicted label sequence jointly via a CRF layer. We evaluate our model on two benchmarking datasets, PubMed RCT (Dernoncourt and Lee, 2017) and NICTA-PIBOSO (Kim et al., 2011), which were both generated from the PubMed database[1]. Our key contributions are summarized as follows:

1. Based on the previous best performing architecture for sequential sentence classification (Dernoncourt et al., 2016), we add one more layer to extract contextual information from surrounding sentences for more accurate prediction of the current one. Together with the CRF algorithm, this allows us to make use of not only the preceding labels' information but also the content and semantics of adjacent sentences to infer the label of the target sentence.

2. We remove the need for a character-based word embedding component without sacrificing performance. For individual sentence encoding, we propose the use of a CNN module as an alternative to RNN for small datasets, suffering less from over-fitting as evidenced by our experiments. Moreover, we incorporate attention-based pooling in both RNN and CNN models to further improve the performance.

3. We adopt dropout with expectation-linear regularization instead of the standard one to reduce the performance gap between training and test phases.

4. We obtain state-of-the-art results on two datasets for sequential sentence classification in medical abstracts, outperforming the previous best models by at least 2% in terms of F1 scores.

2 Related Work

Previous systems for sequential sentence classification concentrate on the rhetorical structure analysis of biomedical abstracts. They are mainly based on naive Bayes (Ruch et al., 2007), support vector machine (SVM) (McKnight and Srinivasan, 2003; Yamamoto and Takagi, 2005; Liu et al., 2013), Hidden Markov Model (HMM) (Lin et al., 2006), and CRF (Kim et al., 2011; Hassanzadeh et al., 2014; Hirohata et al., 2008; Chung, 2009). All these methods heavily rely on numerous carefully hand-engineered features such as lexical (bag-of-words (BOW)), semantic (hypernyms, synonyms), structural (part of speech (POS) tags, lemmas, orthographic shapes, headings), statistical (statistical distributions of token types) and sequential (sentence position, surrounding features, predicted labels) features.

In contrast, current emerging artificial neural network (ANN) based models have removed the need for manually selected features; instead, features are self-learned from the token and/or character embeddings. These deep learning models have revolutionized the natural language processing (NLP) field with state-of-the-art results achieved in various tasks, including the most relevant text classification task (Kim, 2014; Zhang et al., 2016; Conneau et al., 2017; Lai et al., 2015; Joulin et al., 2016; Ma et al., 2015). Most of these models are built upon deep CNNs or RNNs as well as combinations of them, where CNN is good at extracting local n-gram features while RNN is suitable for sequence modeling.

The above-mentioned works for short-text classification do not consider any context of sentence semantics in the models, making them underperform in the sequential sentence classification scenario, where surrounding sentences can play a big role in inferring the label of the current sentence. Recent works that apply deep neural networks to the sequential sentence classification problem include the system proposed by Lee et al. (Lee and Dernoncourt, 2016), where the preceding utterances were used to help classify the current utterance in a dialog into the corresponding dialogue act. Most recent work from Dernoncourt et al. (Dernoncourt et al., 2016) used a CRF layer to optimize the predicted label sequence, where the preceding labels have influence on determining the current label. This model outperformed the state-of-the-art results on two datasets PubMed RCT and NICTA-PIBOSO for sentence classification in medical abstracts.

3 Proposed Model

Notation We denote scalars in italic lowercase (e.g., k), vectors in bold italic lowercase (e.g., $\boldsymbol{s}$) and matrices in italic uppercase (e.g., W). Colon notations $x_{i:j}$ and $\boldsymbol{s}_{i:j}$ are used to denote the se-

[1] https://www.ncbi.nlm.nih.gov/pubmed/

quence of scalars $(x_i, x_{i+1}, ..., x_j)$ and vectors $(\boldsymbol{s}_i, \boldsymbol{s}_{i+1}, ..., \boldsymbol{s}_j)$.

Our model is composed of four components: the word embedding layer, the sentence encoding layer, the context enriching layer, and the label sequence optimization layer. In the following sections they will be discussed in detail.

3.1 Word Embedding Layer

Given a sentence $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}$ comprising N words, this layer maps each word to a real-valued vector as its lexical-semantic representation. Word representations are encoded by the column vector in the embedding matrix $W^{word} \in \mathbb{R}^{d^w \times |V|}$, where d^w is the dimension of the word vector and V is the vocabulary of the dataset. Each column $W_i^{word} \in \mathbb{R}^{d^w}$ is the word embedding vector for the i^{th} word in the vocabulary. The word embeddings W^{word} can be pre-trained on large unlabeled datasets using unsupervised algorithms such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2016).

3.2 Sentence Encoding Layer

This layer takes as input the embedding vector of each token in a sentence from the word embedding layer and produces a vector **s** to encode this sentence. The sequence of embedding vectors is first processed by a bi-directional RNN (bi-RNN) or CNN layer, similar to the ones used in the text classification before (Kim, 2014; Lee and Dernoncourt, 2016; Liu et al., 2016). This layer outputs a sequence of hidden states $\boldsymbol{h}_{1:N}$ ($\boldsymbol{h} \in \mathbb{R}^{d^{hs}}$) for a sentence of N words with each hidden state corresponding to a word. To form the final representation vector $\boldsymbol{s}$ of this sentence, attention-based pooling is used, which can be described using the following equations:

$$A = \text{softmax}(U_s \tanh(W_s H + \boldsymbol{b}_s)), \quad (1)$$

$$S = AH^T, \quad (2)$$

where $H = \begin{bmatrix} h_1 & h_2 & \cdots & h_N \end{bmatrix} \in \mathbb{R}^{d^{hs} \times N}$, $W_s \in \mathbb{R}^{d^a \times d^{hs}}$ is the transformation matrix for soft alignment, $\boldsymbol{b}_s \in \mathbb{R}^{d^a}$ is the bias vector, $U_s \in \mathbb{R}^{r \times d^a}$ is the token level context matrix used to measure the relevance or importance of each token with respect to the whole sentence, softmax is performed along the second dimension of its input matrix, and $A \in \mathbb{R}^{r \times N}$ is the attention matrix.

Here each row of U_s is a context vector $\boldsymbol{u}_s \in \mathbb{R}^{d^a}$ and it is expected to reflect an aspect or component of the semantics of a sentence. To represent the overall semantics of the sentence, we use multiple context vectors to focus on different parts of this sentence.

Finally, the sentence encoding vector $\boldsymbol{s} \in \mathbb{R}^{rd^{hs}}$ is obtained by reshaping the matrix S into a vector.

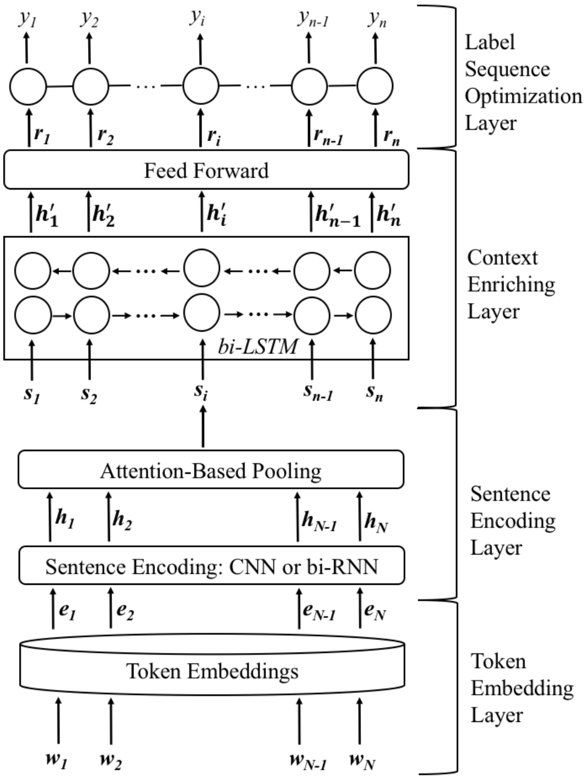
y_1 y_2 y_i y_{n-1} y_n
r_1 r_2 r_i r_{n-1} r_n
Feed Forward
h'_1 h'_2 h'_i h'_{n-1} h'_n
bi-LSTM
s_1 s_2 s_i s_{n-1} s_n
Attention-Based Pooling
h_1 h_2 h_{N-1} h_N
Sentence Encoding: CNN or bi-RNN
e_1 e_2 e_{N-1} e_N
Token Embeddings
w_1 w_2 w_{N-1} w_N
Label Sequence Optimization Layer
Context Enriching Layer
Sentence Encoding Layer
Token Embedding Layer

Figure 1: Model architecture. $\boldsymbol{w}$: original word; $\boldsymbol{e}$: word embedding vector; $\boldsymbol{h}$: sentence-level hidden state output by the bi-RNN or CNN layer; $\boldsymbol{s}$: sentence representation vector; $\boldsymbol{h'}$: abstract-level hidden state output by the bi-LSTM layer; $\boldsymbol{r}$: sentence label probability vector; y: predicted sentence label.

3.3 Context Enriching Layer

This layer takes as input the sequence of individual sentence encoding vectors in a given abstract of n sentences obtained from the last sentence encoding layer, with each vector corresponding to a sentence. It outputs a new sequence of contextualized sentence encoding vectors, which are enriched with the contextual information from surrounding sentences. Specifically, the sequence of individual sentence encoding vectors is input into a bi-LSTM layer, which produces a sequence of hidden state vectors $\boldsymbol{h'}_{1:n}$ ($\boldsymbol{h'} \in \mathbb{R}^{d^{hd}}$) with each corresponding to a sentence. Each of these vectors is subsequently input to a feed-forward neural network with only one hidden layer to get the corresponding probability vector $\boldsymbol{r} \in \mathbb{R}^{l}$, which represents the probability that this sentence belongs to each label, where l is the number of labels.

3.4 Label Sequence Optimization Layer

Within the abstract, the sequence of sentence categories implicitly follows some patterns. For example, the category *Results* is always followed by *Conclusion*, and the category *Methods* is certainly after the *Background*. Making use of such patterns can boost the classification performance via the CRF algorithm (Lample et al., 2016). Given the sequence of probability vectors $\boldsymbol{r}_{1:n}$ from the last context enriching layer for an abstract of n sentences, this layer outputs a sequence of labels $y_{1:n}$, where y_i represents the predicted label assigned to the i^{th} sentence.

In the CRF algorithm, in order to model dependencies between subsequent labels, we incorporate a matrix T that contains the transition probabilities between two subsequent labels; we define $T[i,j]$ as the probability that a token with label i is followed by a token with the label j. The score of a label sequence $y_{1:n}$ is defined as the sum of the probabilities of individual labels and the transition probabilities:

$$s(y_{1:n}) = \sum_{i=1}^{n} \boldsymbol{r}_i(y_i) + \sum_{i=2}^{n} T[y_{i-1}, y_i]. \qquad (3)$$

The score in the above equation can be transformed into the probability of a certain label sequence by taking a softmax operation over all possible label sequences:

$$p(y_{1:n}) = \frac{e^{s(y_{1:n})}}{\sum_{\hat{y}_{1:n} \in Y} e^{s(\hat{y}_{1:n})}}, \qquad (4)$$

where Y denotes the set of all possible label sequences. During the training phase, the objective is to maximize the probability of the gold label sequence. In the testing phase, given an input sequence, the corresponding sequence of predicted labels is chosen as the one that maximizes the score, computed via the Viterbi algorithm (Forney, 1973).

4 Experiments

4.1 Datasets

We evaluate our model on two sources of benchmarking datasets on medical scientific abstracts, where each sentence of the abstract is annotated with one label that is associated with the rhetorical structure. Table 1 summarizes the statistics of the two datasets.

NICTA-PIBOSO This dataset[2] was shared from the ALTA 2012 Shared Task (Amini et al., 2012), the goal of which is to build automatic sentence classifiers that can map the sentences from biomedical abstracts into a set of pre-defined categories for Evidence-Based Medicine (EBM).

PubMed RCT This new dataset was curated by (Dernoncourt and Lee, 2017)[3] and is currently the largest dataset for sequential sentence classification. It is based on the PubMed database of biomedical literature and each sentence of each abstract is labeled with its role in the abstract using one of the following classes: *background, objective, method, result, and conclusion*. Table 2 presents an example abstract comprising structured sentences with their annotated labels.

4.2 Training Settings

For both datasets, test performance is assessed on the training epoch with best validation performance and F1 scores (weighted average by support (the number of true instances for each label)) are reported as the results.

The token embeddings were pre-trained on a large corpus combining Wikipedia, PubMed, and PMC texts (Moen and Ananiadou, 2013) using the word2vec tool[4] (denoted as "Word2vec-wiki+P.M."). They are fixed during the training phase to avoid over-fitting. We also tried other types of word embeddings, such as the word2vec embeddings pre-trained on the Google News dataset[5] (denoted as "Word2vec-News"), word2vec embeddings pre-trained on the Wikipedia corpus[6] (denoted as "Word2vec-wiki"), GloVe embeddings pre-trained on the cor-

[2] This dataset can be found online at https://www.kaggle.com/c/alta-nicta-challenge2

[3] This dataset can be downloaded from https://github.com/Franck-Dernoncourt/pubmed-rct

[4] The word vectors can be downloaded at http://bio.nlplab.org/

[5] https://code.google.com/archive/p/word2vec/

[6] https://github.com/jind11/word2vec-on-wikipedia

Dataset	$|C|$	$|V|$	Train	Validation	Test
NICTA-PIBOSO	6	17k	720 (7.7k)	80 (0.9k)	200 (2.2k)
PubMed 20k	5	68k	15k (180k)	2.5k (30k)	2.5k (30k)
PubMed 200k	5	331k	190k (2.2M)	2.5k (29k)	2.5k (29k)

Table 1: Datasets statistics. $|C|$ denotes the number of labels, $|V|$ represents the vocabulary size. For the train, validation, and test sets, we indicate the number of abstracts followed by the number of sentences in parentheses.

Category	**Sentences**
BACKGROUND	Emotional eating is associated with overeating and the development of obesity. [...]
OBJECTIVES	The aim of this study was to test if attention bias for food moderates the effect of self-reported emotional eating during sad mood (vs neutral mood) on actual food intake. [...]
METHODS	Participants (N = 85) were randomly assigned to one of the two experimental mood induction conditions (sad/neutral). [...]
RESULTS	[...] Yet, attention maintenance on food cues was significantly related to increased intake specifically in the neutral condition, but not in the sad mood condition.
CONCLUSIONS	The current findings show that self-reported emotional eating (based on the DEBQ) might not validly predict who overeats when sad, at least not in a laboratory setting with healthy women. [...]

Table 2: A typical abstract example with structured sentences and their corresponding annotated labels. The PMID of this abstract is 24854809.

pus of Wikipedia 2014 + Gigaword 5[7] (denoted as "Glove-wiki"), fastText embeddings pre-trained on Wikipedia[8] (denoted as "FastText-wiki"), and fastText embeddings initialized with the standard GloVe Common Crawl embeddings and then fine-tuned on PubMed abstracts plus MIMIC-III notes (denoted as "FastText-P.M.+MIMIC"). The comparison results are summarized in the next section.

The model is trained using the Adam optimization method (Kingma and Ba, 2014). The learning rate is initially set as 0.003 and decayed by 0.9 after each epoch. For regularization, dropout (Srivastava et al., 2014) is applied to each layer. For the version of dropout used in practice (e.g., the dropout function implemented in the TensorFlow and Pytorch libraries), the model ensemble generated by dropout in the training phase is approximated by a single model with scaled weights in the inference phase, resulting in a gap between training and inference. To reduce this gap, we adopted the dropout with expectation-linear regularization introduced by Ma et al. (2016) to explicitly control the inference gap and thus improve the generalization performance.

Hyperparameters were optimized via grid search based on the validation set and the best configuration is shown in Table 3. The window sizes of the CNN encoder in the sentence encoding layer are 2, 3, 4 and 5. The RNN encoder in the sentence encoding layer is set as LSTM for the PubMed datasets and gated recurrent unit (GRU) for the NICTA-PIBOSO dataset. Code for this work is available online[9].

5 Results and Discussion

Table 4 compares our model against the best performing models in the literature (Dernoncourt et al., 2016; Liu et al., 2013). There are two variants of our model in terms of different implementations of the sentence encoding layer: the model that uses bi-RNN to encode the sentence is called HSLN-RNN; while the model that uses the CNN module is named HSLN-CNN. We have evaluated both model variants on all datasets. And as evidenced by Table 4, our best model can improve the F1 scores by 2%-3% in absolute number compared with the previous best published results for all

[7] http://nlp.stanford.edu/data/glove.6B.zip

[8] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[9] https://github.com/jind11/HSLN-Joint-Sentence-Classification

Parameter	PubMed		NICTA	
	RNN	CNN	RNN	CNN
d^{hs}	200	-	200	-
d^{hd}	200	200	200	300
d^{a}	200	100	250	75
d^{c}	-	200	-	150
r	15	1	5	4
β	0.01	0.001	0.01	0.01
dr	0.5	0.5	0.6	0.6

Table 3: Hyperparameter settings. d^{hs}: hidden size of the sentence-level RNN layer (single direction); d^{hd}: hidden size of the abstract-level bi-LSTM layer (single direction); d^{a}: dimension of the context vector $\boldsymbol{u}_s$; r: number of context vectors; β: coefficient of the dropout regularization added to the total loss; dr: dropout.

datasets. For the PubMed 20k and 200k datasets, our HSLN-RNN model achieves better results; however, for the NICTA dataset, the HSLN-CNN model performs better. This makes sense because the CNN sentence encoder has fewer parameters to be optimized, thus the HSLN-CNN model is less likely to over-fit in a smaller dataset such as NICTA. With sufficient data, however, the increased capacity of the HSLN-RNN model offers performance benefits. To be noted, this performance gap between RNN and CNN sentence encoder gets larger as the dataset size increases from 20k to 200k for the PubMed dataset.

Model	PubMed		NICTA
	20k	200k	
Best Published			
Marco Lui (Lui, 2012)	-	-	82.0
bi-ANN (Dernoncourt et al., 2016)	90.0	91.6	82.7
Our Models			
HSLN-CNN	92.2	92.8	**84.7**
HSLN-RNN	**92.6**	**93.9**	84.3

Table 4: Comparison of F1 scores (weighted average by support (the number of true instances for each label)) between our model and the best published methods. The presented results of our model are evaluated on the test set of the run with the highest F1 score on the validation set.

Table 5 presents the ablation analysis of our model (on the PubMed 20k dataset), where we remove one component at a time and quantify the performance drop (reported on F1 scores). As can be seen from Table 5, our HSLN-CNN model uniformly suffers a little more from the component removal than the HSLN-RNN model, indicating that the HSLN-RNN model is more robust. When the context enriching layer is removed, both models experience the most significant performance drop and can only be on par with the previous state-of-the-art results, strongly demonstrating that this proposed component is the key to the performance improvement of our model. Furthermore, even without the label sequence optimization layer, our model still significantly outperforms the best published methods that are empowered by this layer, indicating that the context enriching layer we propose can help optimize the label sequence by considering the context information from the surrounding sentences. Last but not the least, the dropout regularization and attention-based pooling components we add to our system can help further improve the model in a limited extent.

Model	HSLN-RNN	HSLN-CNN
Full Model	**92.6**	**92.2**
– context	90.0	89.0
– seq. opt.	92.3	91.8
– dropout reg.	92.4	91.9
– attention	92.4	91.7

Table 5: Ablation analysis. F1 scores are reported. "– context" is our model without the context enriching layer. "– seq. opt." is our model without the label sequence optimization layer. "– dropout reg.' is our model using the standard dropout strategy without the expectation-linearization regularization. "– attention" refers to the model without attention-based pooling, i.e., in the sentence encoding layer, the final hidden state is used for the HSLN-RNN model while max-pooling is used for the HSLN-CNN model.

Table 6 and 7 detail the results of classification for each label in terms of performance scores (precision, recall and F1) and confusion matrix, respectively (for our HSLN-RNN model trained on the PubMed 20k dataset). These show that the classifier is very good at predicting the labels *Methods*, *Results* and *Conclusions*, whereas the greatest difficulty the classifier has is in distinguishing *Background* sections from *Objectives* sections. One fifth of *Background* sentences are incorrectly classified as *Objectives*, while around one forth of *Objectives* sentences are wrongly assigned to the label of *Background*. We conjecture this difficulty mainly comes from the fact that the difference between *Background* and *Ob-*

jectives sentences in terms of writing style is less obvious compared with the other sections of the abstract. Moreover, our model has some difficulty in telling *Methods* sentences apart from *Results* sentences.

Label	P	R	F1	Support
Background	78.5	80.0	79.2	3077
Objectives	74.2	69.9	72.0	2333
Methods	95.0	97.7	96.3	9884
Results	96.8	95.3	96.0	9713
Conclusions	97.6	96.5	97.1	4571
Total	92.6	92.7	92.6	29578

Table 6: Results (presented in percentage) in terms of precision (P), recall (R) and F-measure (F1) on the test set for each label obtained by our HSLN-RNN model on the PubMed 20k dataset.

	B	**C**	**M**	**O**	**R**
B	2460	4	69	537	7
C	4	4413	11	1	142
M	37	11	9657	27	152
O	632	0	68	1630	3
R	2	95	362	1	9253

Table 7: Confusion matrix obtained by our model on the PubMed 20k dataset. Rows correspond to predicted labels, and columns correspond to true labels. B represents background, O represents objectives, M represents methods, R represents results, and C represents conclusions.

Table 8 presents a few examples of prediction errors that are produced by our HSLN-RNN model trained on the PubMed 20k dataset. This error analysis suggests that one of the biggest model error sources could be from the debatable gold standard labels of the dataset. For example, the sentence “Depressive disorders are one of the leading components of the global burden of disease with a prevalence of up to 14% in the general population.” is indeed introducing the background of the problem (depressive disorders) on which this article is going to focus; however, the gold label classifies it into the *Objective* category. For another instance, the sentence “A post hoc analysis was conducted with the use of data from the evaluation study of congestive heart failure and pulmonary artery catheterization effectiveness (escape).” belongs to the *Result* label according to the gold standard, but it makes more sense that it should be classified as a *Method* label.

Figure 2 presents an example of the transition matrix after the HSLN-RNN model has been trained on the PubMed 20k dataset, which encodes the transition probability between two subsequent labels. It effectively reflects what label is the most likely one that follows the current one. For example, by comparing the transition scores in the *Result* row in Figure 2, we can conclude that a sentence pertaining to the *Result* is typically followed by a sentence pertaining to the *Conclusion* and is unlikely to be followed by a sentence in the *Background* category (transition scores of 2.48 vs -5.46), which makes sense. From this transition matrix, we can figure out the most probable label sequence: $Background \rightarrow Objective \rightarrow Method \rightarrow Result \rightarrow Conclusion$, which is also consistent with our expectations.

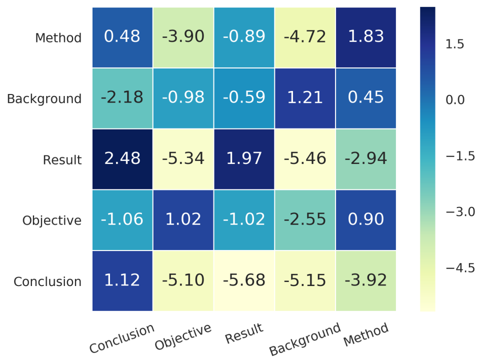

Figure 2: Transition matrix of label sequence after the HSLN-RNN model has been trained on the PubMed 20k dataset. The rows represent the label of the previous sentence, while the columns represent the label of the current sentence.

In order to test the importance of pretrained word embeddings, we performed experiments with different sets of publicly published word embeddings, as well as our locally curated word embeddings, to initialize our model. Table 9 gives the performance of six different word embeddings for our HSLN-RNN model trained on the PubMed 20k dataset. According to Table 9, the training methods that create the word embeddings do not have a strong influence on model performance, but the corpus they are trained on does. The combination of Wikipedia and PubMed abstracts as the corpus for unsupervised word embedding training yields the best result, and the individual use of either the Wikipedia corpus or the PubMed abstracts performs much worse. Although the dataset we

Sentence	Predicted	Gold
Depressive disorders are one of the leading components of the global burden of disease with a prevalence of up to 14% in the general population. [25829103]	Background	Objective
This study assessed whether diets with different fat quality and supplementation with coenzyme Q10 (CoQ) affect the metabolomic profile in urine. [24986061]	Objective	Background
A post hoc analysis was conducted with the use of data from the evaluation study of congestive heart failure. [24845963]	Method	Result
Hence, 47 secondary schools from all 12 districts of the city [...] are participating in the study. [25150368]	Result	Method
This study investigated whether oxytocin can affect attentional bias in social anxiety. [25552432]	Objective	Method
We hypothesize that BMC+Phone and BMC+Home will produce greater reductions in BMI percentiles than BMC alone. [24456698]	Conclusion	Method

Table 8: Examples of prediction errors of our HSLN-RNN model trained on the PubMed 20k dataset. Each sentence is followed by the PMID of the abstract that this sentence belongs to, which is enclosed in middle brackets. The "Predicted" column indicates the label predicted by our model for a given sentence. The "Gold" column indicates the gold label of the sentence.

are using for evaluation is also from PubMed abstracts, using only the PubMed abstracts together with MIMIC notes without the Wikipedia corpus does not guarantee better result (see the "FastText-P.M.+MIMIC" embeddings in Table 9), which may be because the corpus size of PubMed abstracts plus MIMIC notes (about 12.8 million abstracts and 1 million notes) is not large enough for good embedding training compared with the corpus consisting of at least billion tokens such as the Wikipedia.

Embedding	Dimension	P.M. 20k
Glove-wiki	200	92.0
FastText-wiki	300	92.2
FastText-P.M.+MIMIC	300	92.0
Word2vec-News	300	92.2
Word2vec-wiki	200	92.1
Word2vec-wiki+P.M.	200	**92.6**

Table 9: Comparison of performance with different choices of word embeddings for our HSLN-RNN model trained on the PubMed 20k dataset (reported on F1-scores on the test set). "P.M." means PubMed.

6 Conclusion

In this work, we have presented an ANN based hierarchical sequential labeling network to classify sentences that appear sequentially in text. We demonstrate that incorporating the contextual information from surrounding sentences to help classify the current one by using an LSTM layer to sequentially process the encoded sentence representations can improve the overall quality of predictions. Our model outperforms the state-of-the-art results by 2%-3% on two datasets for sequential sentence classification in medical abstracts. We expect that our proposed model can be generalized to any problem that is related to sequential sentence classification, such as the paragraph-level sequential sentence categorization in full-text articles for better text mining and document retrieval (Westergaard et al., 2018).

7 Future Work

Although the whole PubMed database contains over 2 million abstracts with part of them accompanied by full-text articles, only a small fraction of them are structured and contain the label information utilized in this work. We plan to make use of the rest unannotated abstracts or full texts to pre-train our model and then fine tune it to the target annotated datasets inspired by the work from (Howard and Ruder, 2018) so that the performance can be further boosted.

Acknowledgments

This work was supported by funding grant U54-HG007963 from National Human Genome Research Institute (NHGRI).

References

American National Standards Institute. 1979. American national standard for writing abstracts (ansi z39. 14-1979).

Iman Amini, David Martinez, Diego Molla, et al. 2012. Overview of the alta 2012 shared task.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Grace Y Chung. 2009. Sentence retrieval for abstracts of randomized controlled trials. *BMC medical informatics and decision making*, 9(1):10.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

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

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2016. Neural networks for joint sentence classification in medical paper abstracts. *arXiv preprint arXiv:1612.05251*.

G David Forney. 1973. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2014. Identifying scientific artefacts in biomedical literature: The evidence based medicine use case. *Journal of biomedical informatics*, 49:159–170.

Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, and Mitsuru Ishizuka. 2008. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Arif E Jinha. 2010. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing*, 23(3):258–263.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC bioinformatics*, volume 12, page S5. BioMed Central.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *HLT-NAACL*.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Jimmy Lin, Damianos Karakos, Dina Demner-Fushman, and Sanjeev Khudanpur. 2006. Generative content models for structural analysis of medical abstracts. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pages 65–72. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Yuanchao Liu, Feng Wu, Ming Liu, and Bingquan Liu. 2013. Abstract sentence classification for scientific papers based on transductive svm. *Computer and Information Science*, 6(4):125.

Marco Lui. 2012. Feature stacking for sentence classification in evidence-based medicine. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 134–138.

Mingbo Ma, Liang Huang, Bing Xiang, and Bowen Zhou. 2015. Dependency-based convolutional neural networks for sentence embedding. *arXiv preprint arXiv:1507.01839*.

Xuezhe Ma, Yingkai Gao, Zhiting Hu, Yaoliang Yu, Yuntian Deng, and Eduard Hovy. 2016. Dropout with expectation-linear regularization. *arXiv preprint arXiv:1609.08017*.

Larry McKnight and Padmini Srinivasan. 2003. Categorization of sentence types in medical abstracts. In *AMIA Annual Symposium Proceedings*, volume 2003, page 440. American Medical Informatics Association.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43.

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

Patrick Ruch, Celia Boyer, Christine Chichester, Imad Tbahriti, Antoine Geissbühler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, et al. 2007. Using argumentation to extract key sentences from biomedical abstracts. *International journal of medical informatics*, 76(2-3):195–200.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Mark Ware and Michael Mabe. 2015. The STM report: An overview of scientific and scholarly journal publishing.

David Westergaard, Hans-Henrik Stærfeldt, Christian Tønsberg, Lars Juhl Jensen, and Søren Brunak. 2018. A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts. *PLoS computational biology*, 14(2):e1005962.

Yasunori Yamamoto and Toshihisa Takagi. 2005. A sentence classification system for multi biomedical literature summarization. In *Data Engineering Workshops, 2005. 21st International Conference on*, pages 1163–1163. IEEE.

Rui Zhang, Honglak Lee, and Dragomir Radev. 2016. Dependency sensitive convolutional neural networks for modeling sentences and documents. *arXiv preprint arXiv:1611.02361*.