A Deep Learning Architecture for Protein-Protein Interaction Article Identification

Shweta*, Asif Ekbal†, Sriparna Saha‡ and Pushpak Bhattacharyya§
Department of Computer Science and Engineering, IIT Patna
India
Email: *shweta.pcs14@iitp.ac.in, †asif@iitp.ac.in, ‡sriparna@iitp.ac.in, §pb@iitp.ac.in

***Abstract*—In recent past there has been phenomenal growth in biomedical literature and health care records. Robust text mining techniques are essential in order to properly organise the documents as well as to extract relevant information. Traditional techniques for document classification focus on machine learning algorithms where learning of classifier is decided on the basis of labelled data and the features that are prominent. In this paper we focus on developing an automated technique for classifying biomedical articles containing protein-protein interaction related information against the others. Our proposed approach is based on deep neural network framework. We investigate the role of convolution neural network (CNN) and propose two model variants. We evaluate the proposed approach on the benchmark datasets of BioCreative-II Interaction Article Subtask (IAS) data sets. Effectiveness of our proposed model is evident with the significant performance gains, 2.8 points in terms of F-measure and 5 points in terms of accuracy over the traditional models.**

***Index Terms*—Protein Protein Interaction (PPI), Word Embedding, Convolutional Neural Network (CNN), Interaction Article Subtask (IAS).**

I. Introduction

Rapid growth of digital prints in recent past had made it difficult even for the scientists to assimilate the relevant publications of their needs. Organizing the documents as per the needs is, therefore, has drawn the attention of researchers at large. The problem is related to text classification (also called document categorization), an important task in biomedical document processing [3]. Robust text mining techniques need to be investigated in order to properly organise the documents as well as to extract the relevant information. Physical protein-protein interaction (PPI) is an extensively popular research area because of its crucial role in controlling important functionality like cell division and its implication in various human diseases such as cancer. In the past 20 years, biomedical literature has grown tremendously. PubMed is considered to be one of the most widely used databases consisting of more than 25 million citations for biomedical literature from MEDLINE, life science journals, and online books. It was observed that, in recent past size of MEDLINE has increased by 4.2% and each year 3.1% of increments are being seen in the new entries. Searching within these literature and limitations of keyword based search techniques to rank relevant articles has increased huge interest in investigating more efficient search technique. Generally, biologist who identify protein interaction information look at whole text articles. This shows that only the titles are not sufficient enough to curate full-text articles. This demands an efficient system for curating the article by inspecting whole article rather than just looking only at the title. This can not be achieved manually due to the complexity involved in the process. Therefore, an automated text mining system is required to be put in place in order to reduce the efforts.

In this paper, we propose a technique based on deep learning architecture for solving the problem of document classification. Our study focuses on deep convolutional neural networks (CNN) [16]. Deep learning techniques have shown great promise in the domains of computer vision [7]. Researchers in text mining and natural language processing (NLP) have recently started developing models based on different models of deep learning architecture. Effective representation of documents could lead to performance improvement in text classification problems. Word representation also known as the word embedding is able to capture the word semantic and syntactic information [17]. The model of word embedding can address the shortcoming of traditional bag-of-word (BoW) representation of documents with the assumption that similar words do appear in similar context.

Convolution neural networks (CNN) is a biologically inspired feed forward neural network [16] whose convolutional layers alternate with sub-sampling layers, similar to the case of the mammalian visual cortex. CNN layers are developed by performing convolution operation which is later preceded by the pooling operation. With the wide-spread interest in deep learning techniques, CNN is widely used for solving various NLP problems, including semantic parsing[20], sentence modeling[9] and some other basic NLP tasks [4].

We develop a system for PPI article identification in line with the framework introduced in BioCreAtIvE-II [1] Interaction Article Subtask (IAS) challenge. The goal of the IAS was to identify interaction annotation relevant articles based on PubMed titles and abstracts. The aim was to discriminate between curatable and non-curatable articles. The documents are derived from MEDLINE, and consists of 5,495 training and 667 test documents. A document is classified as curatable if it contains protein interaction information, otherwise it is considered as non-curatable. At first we develop a baseline model which is based on supervised machine learning algorithm, namely Support Vector Machine (SVM) [8]. The classifier was trained with a set of features extracted from the training documents. We build our second baseline model by training

SVM with word embedding feature [17], which is capable of containing latent syntactic and semantic information of a word. Finally, we use CNN having single layer of convolution with word embedding as an input.

II. Related Works

In the existing literature, there are several methods that have been used to characterize interaction information, mainly focusing on gene and protein interaction from the biomedical literature. Most of the techniques explored prior to BioCreative-II Challenge were focused on co-occurrence of protein names associated with interaction words [5]. Most of the participants of BioCreative-II Challenge rely on traditional bag-of-words (BoW) approach for representing the documents [12]. The best team reported in [14] have used simple BoW features with SVM as the underlying classifier. A machine learning technique where BoWs were extended to bag-of-nlp was introduced by Grover et al.[5]. Apart from considering just words, they used a different variety of NLP based features. Along with BoW, domain dependent features such as interaction trigger keywords, protein named entities (NEs) were used in [15] to improve the performance. Feature selection based on Chi-square within linear SVM was used by Cohen et al.[3] to understand the sensitive features for document classification. In recent times, neural network based architecture[13] has been applied for text classification. The work reported in [13] used a recurrent structure to capture contextual information on four different datasets and claimed a significant performance improvement over the existing techniques. Nevertheless, to the best of our knowledge the importance of using CNN to detect local and higher order features to biomedical document classification has not been explored so far.

III. Overview of CNN Architecture

In general, convolutional neural network (CNN) for classification consists of the following layers [4].

- Encoding the words into real-valued vector by word embedding.
- Convolutional layer is used to identify the n-grams.
- Pooling layer is used to identify the most relevant feature set.
- A softmax layer is required to perform classification.

Figure-1 gives an overview of our proposed model for document classification. Let us assume that we have an abstract A containing n words.

$$A = \{w_1, w_2, w_3.....w_n\}$$

In the following subsection we describe each step of CNN.

A. Word Embeddings

The input to the convolution network is a fixed length word vector, each bit of which is represented by a numerical value. Each word $w_i \in A$ is encoded by a real value v_i. In order to fix the length, padding is performed whenever abstract is having less than n words. The k-dimensional word vector corresponding to the i^{th} word in the abstract is v_i in R^k. We generate a word vector matrix $W_{n\times k}$ for each abstract A. An abstract having size n can be represented as follows using concatenation operator $\oplus$:

$$v_{1:n} = v_1 \oplus v_2 \oplus ..... \oplus v_n \tag{1}$$

We use the publicly available implementation *word2vec* [17] tool for extracting word embedding features.

B. Convolution

Vector representation of a word is followed by convolution operation, where *filter* $\mathbf{w}$ is applied to window of h words to obtain new features. For example, a feature f_i is generated from the window of word vector $v_{i:i+h-1}$

$$f_i = g(\mathbf{w} \cdot v_{i:i+h-1} + b) \tag{2}$$

where b is called the bias term and g is a non linear function. In our experiments we have used the rectified linear unit as a non linear function. We obtain a feature map f by applying given filter $\mathbf{w}$ to every possible window of word in the abstract.

$$\begin{aligned} f = [&g(\mathbf{w} \cdot v_{1:1+h-1} + b), g(\mathbf{w} \cdot v_{2:2+h-1} + b), g(\mathbf{w} \cdot v_{3:3+h-1} \\ &+ b), ..., g(\mathbf{w} \cdot v_{n-h+1:n} + b)] \\ = &[f_1, f_2, f_3.....f_{n-h+1}] \end{aligned} \tag{3}$$

This procedure can be performed by using several filters with distinct window sizes in order to increment the coverage of n-gram model.

C. Pooling

The incorporation of pooling layer in CNN architecture is to further abstract the features produced from the convolution layer. There are several ways for performing pooling operation, such as taking average, mean, maximum or a learned linear combination of the neurons. Most widely accepted pooling function is max-pooling [4] which is used as the most suitable one in recognizing the relevant features. We have applied max-pooling operation over the feature map and set the maximum value as a feature for this particular filter.

$$\hat{d} = max(f_1, f_2, f_3.....f_{n-h+1}) \tag{4}$$

With the incorporation of pooling operation, the size of representation is reduced by almost half which, in turn, improves the computational cost and helps in filtering out unwanted word compositions. We generate different configurations (or, features) by applying multiple filters with varying window sizes. These features help in forming up the penultimate layers which are provided as input to the fully connected softmax layer. The output of the softmax layer is a probability distribution of our two classes, *curatable* and *non-curatable*.

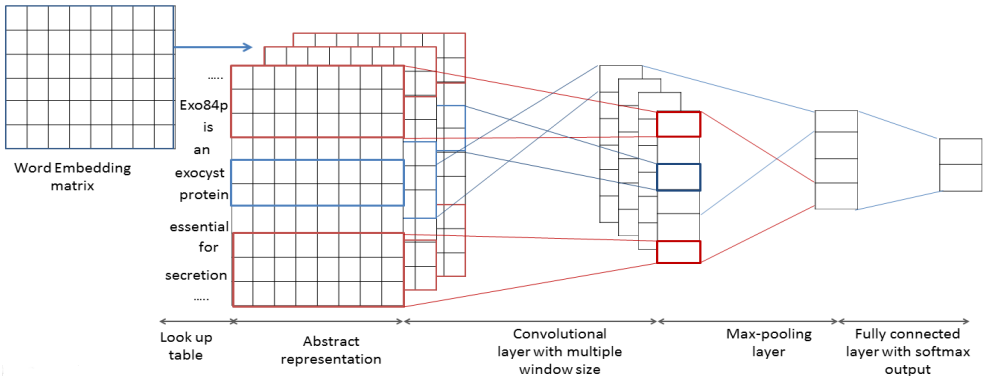

Fig. 1: Proposed system architecture

D. Regularization and Classification

At the end we form a single feature vector z by concatenating all the feature vectors obtained from every filter.

$$z = [\hat{d}_1, \hat{d}_2, \hat{d}_3, ..., \hat{d}_m] \tag{5}$$

where $\hat{d}_j$ is a feature value obtained using j-th filter and m is total number of filters used by the model. Dropouts are performed on the penultimate layer with a constraint on l_2-norms of weight vectors [7]. We generate new feature vector z^d by randomly setting proportion ϕ [1] of original feature vector as zero. After executing dropout z^d is fed instead of z into fully connected layer of the network. For output unit y in forward propagation:

$$y = C \cdot z^d + a \tag{6}$$

where C is model parameter and a is bias term.
At test time unseen articles get features using the learned weight vectors that are not dropped out. For this we have also scaled the learned weight vectors C by $\hat{C}$=ϕ.C as Kim et al.[10].

IV. Features for PPI Article Identification

In order to build a strong baseline against our proposed model, we design a good set of handcrafted features for training of SVM. We pre-process each article to extract *article* and *title* from the XML document, perform stemming of words in abstract using Porter stemmer algorithm [18], and remove the stop words.

1) Single word feature: This feature is used to cope up with the high dimensionality of simple BoW features. Probability of each word w is computed from the training data to take into account its appearance in the positive abstract (i.e. PPI rich articles) denoted as $p_+(w)$, and in the negative abstract (i.e. non-PPI related articles) denoted as $p_-(w)$. They are computed as the ratio of the number of abstracts where w appears to the total number of positive/negative abstracts. Filtering is performed to remove the words having length less than or equal to 2. All the words are ranked on the basis of score

$$S(w) = |p_+(w) - p_-(w)| \tag{7}$$

The words with high scores denote strong association with positive or negative class. The BoW is generated by considering top scorer 800 words.
2) Word pair feature: This is an extension to the above feature set in which we design a feature which consists of word pairs generated from 800 words. This leads to 640,000 total word pairs. We generate the word bigram that occurs in the window of 10 words. In order to filter the word pair we follow the same strategy as we follow to generate the single word feature set:

$$S(w_i, w_j) = |p_+(w_i, w_j) - p_-(w_i, w_j)| \tag{8}$$

where $p_+(w_i, w_j)$ and $p_-(w_i, w_j)$ are the probabilities of such word pairs appearing in the positive abstract and negative abstract set, respectively. We choose 1,500 word pairs based on their ranks (depending upon the scores). We generate a bag of the selected word pairs and define a binary-valued feature, setting 1 for the presence of word pair and 0 otherwise.
3) Protein count: indent For extracting the unique protein mentions from the abstracts, we use ABNER [19], an existing biomedical NER system. Based on the NE information, a feature is defined that counts the number of unique protein mentions per abstract. Higher the protein count, higher is the probability of abstract occurring in the curatable document.
4) Title-proteins: We define a feature that checks whether the protein appears in the title of the document or not. A value of 1 is set if there is a protein term in the title, otherwise 0. Title of any article may contain some evidences if it is relevant to PPI.
5) Protein relation: This feature picks up the most relevant words that occur between/across two protein terms. Through ABNER, we are able to extract the protein terms. We extract words occurring more than 5 times from training set. A feature is then defined that checks

[1] we set proportion ϕ of the element of original feature vector zero by performing Bernoulli Distribution

the number of protein relations present in an abstract.
(i) *PROTIEN*.*interact.*with.**PROTIEN*
(ii) *PROTIEN*.*associate.*with.**PROTIEN*
(iii) *PROTIEN*.*in.*complex.*with.**PROTIEN*

6) Negation feature: This feature is highly efficient in identifying the abstracts which do not contain any protein relevant information. This feature is based on the count of the following types of phrases in abstract: "*not required*", "*not significant*", "*not affect*" etc. These words are drawn from the single word feature set preceded by the negation words like "not". We define this feature that takes the value equal to the number of occurrences of these phrases in the abstract.
7) Number of trigger words: Trigger words are those instances which occur very frequently with the protein names. We extract the most frequently occurring 30 words that appear very frequently with the protein names. We define a feature which is set equal to the number of trigger words present in the abstract. More the trigger words present in the abstract higher is the chance of its protein relevance.
8) Dependency feature: indent This feature plays a very important role in identifying the protein pair relations from the abstract. We use the Standford dependency parser[2] to generate the dependency relations between any two proteins. As such we get the relations between the protein word and words of the abstract, say *relation*(*PROTIENWORD*,*WORD*) also termed as tuples extracted from the training set. We extract a total of $9,842$ tuples having frequencies > 2 and generate a binary-valued feature vector of length equal to the number of tuples. At the time of testing we assign the value 1 for all those relation tuples which occur in the abstract else 0.

V. Datasets and Experiments

We use the datasets and evaluation scheme as made available in the BioCreative-II shared task on Interaction Article Subtask. The datasets comprise of PubMed titles and abstracts. Training dataset consists of two collections, namely positive (relevant articles) and negative (non-relevant articles) collections. There are 3,536 and 1,959 PubMed titles and abstracts for positive and negative collection, respectively. The training dataset is, thus, highly imbalanced. The test dataset, on the other hand, consists of 677 total articles with 338 positive and 339 negative articles. The system is evaluated in terms of recall, precision, F-measure and accuracy.

A. Experimental setup

We conduct three kinds of experiments as mentioned below:
First set: Here, we define a strong baseline to compare with our proposed model.
Second set: In this set of experiments we conduct several experiments (through cross-validation) to find out optimal parameter settings of CNN by several variants of parameters.
Third set: This set of experiments aims at checking model's effectiveness by comparing with the existing systems.

The word vector was trained using skip-gram architecture [17]. Using freely available *word2vec*[3] tool we generate 200-dimensional word embeddings using the recent Wikipedia[4], PubMed[5] and PMC Open Access[6] biomedical literature.

We develop the following baselines for comparison with our proposed system.

- **Baseline-1**: This baseline is constructed by training SVM with a set of features as described in Section IV. We use SVMlight [8] implementation with linear kernel.
- **Baseline-2**: This baseline is constructed by training SVM with word embedding features. We represent a document by averaging vectors of all the distinct words a document contains.

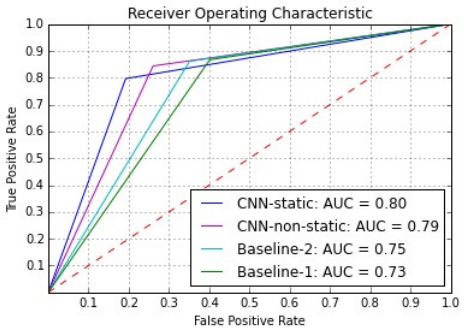

Fig. 2: ROC curve comparison between true positive rate & false positive rate over the baselines and proposed models

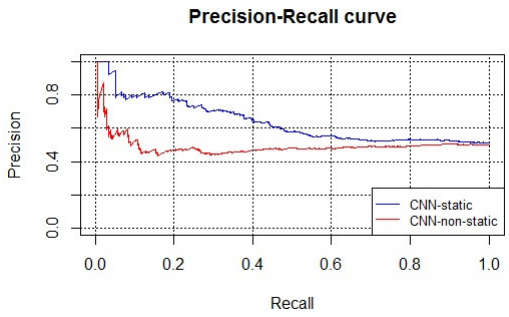

Fig. 3: Precision-Recall curve of CNN-static (IAS) & CNN-non-static (IAS)

B. Parameter tuning

We use the default parameters of SVMlight for both the SVM based baselines. We keep the word embedding size as 200 throughout the experiments. *Rectified linear unit* as non-linear function[7] was used in the experiment. We set the other parameters as follows: Feature map size= 100, dropout rate, ϕ=

[2]http://nlp.stanford.edu/software/stanford-dependencies.shtml

[3]https://code.google.com/p/word2vec/
[4]http://en.wikipedia.org/wiki/Main˙Page
[5]http://www.ncbi.nlm.nih.gov/pubmed
[6]http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist
[7]*tanh* was also used, but the results were not satisfactory

0.5, mini batch[8] size=50, l_2 regularization constraint of 3. All these parameters are tuned by using development data which is the 15% training data with uniform positive or negative instances. The maximum length of abstract that we use is having 200 words. All the experimental results are produced with 30 epochs. Training is done through stochastic gradient descent over shuffled mini-batches with the Adadelta update rule [21]. We develop two variants of our proposed CNN model: CNN-static(IAS) and CNN-nonstatic(IAS). In CNN-static(IAS), all words of abstract are kept static throughout the experiments and other hyperparameters are learnt while training. In CNN-nonstatic(IAS) model, word vectors apart from the hyperparameters are also fine-tuned.

C. Results and Discussions

We report the results of our proposed model in Table-I. We use the set of handcrafted features as well as its different combinations trained using SVM. We found that with the set of whole features learnt using SVM, the system performs quite well. This is considered as our first baseline. In our second baseline, we are able to improve the accuracy by 2.22% over the first baseline. Several experiments with the different window size settings were done to show the effectiveness of incorporating the CNN model. Results show that our CNN-static (IAS) model performs the best compared to all the other models. We also carry out experiments to understand the effects of model performance with varying window sizes. We vary window sizes as $h \in \{3,4,5,6\}$ in both the model variants. Detailed results are shown in Table-II. Results show that combination of different window sizes performs superior over the single window size. For both the model variants we obtain the best result (c.f. line no. 6 of Table-II) using the window size combination $\{3,4,5\}$. In order to understand the model behaviour we show the precision-recall curve in Fig-3. It shows that CNN-static (IAS) is more favorable to precision. We also evaluate our model using receiver operating characteristic (ROC) [6]. We show ROC curve in Fig-2 for the baselines and two variants of CNN. This also shows that AUC value is higher in CNN-static (IAS) compared to the other models.

D. Comparative Analysis

We compare our proposed technique with the existing systems reported in BioCreAtIvE II IAS task [1], and also with the systems reported thereafter. Comparisons show that our proposed model achieves state-of-the-art performance with a very less complex model. The best system [11] reported in BioCreative shared task is based on SVM and also made use of shallow parsing features. They reported the F-measure value of 77.95%, which is almost 2.25% lower compared to ours. The system reported in [5] made use of Conditional Random Field and Maximum Entropy Model, which were trained with BoW features along with some other features like chunk, phrase, protein information etc. They reported to have obtained the F-measure of 77.73%. A SVM based system proposed by William et al.[15] achieved the F-measure of 80.25% with majority voting of different run using BoW feature, protein named entity and some trigger word based features. One of the drawbacks of all these existing systems is that they are based on supervised classification algorithms, which need hand-crafted features in order to obtain reasonable accuracies. Our proposed CNN based models, in contrast, do not require any hand-crafted features, but still can achieve state-of-the-art performance. With two models, CNN-static(IAS) and CNN-non-static(IAS), we obtain the accuracies of 80.20% and 78.88%, respectively.

[8]http://deeplearning4j.org/troubleshootingneuralnets

E. Error-Analysis

Analysis of the outputs of both the model variants yield the following facts:

- Model wrongly predicts curatable document (i.e. PPI relevant) to non-curatable (i.e. non-PPI relevant) because of:
  - Presence of conflicting n-grams such as *lower binding*, *preventing aggregation*, *partially regulated* etc. Sometimes, the words like *lower*, *preventing* etc. appear in the vicinity of strong PPI bearing words like *inhibit*, *regulated*, *interaction* etc., and this actually suppresses the action of strong words resulting in the classification to non-curatable.
  - Implicit mentions of the PPI information. It is observed that some of the abstracts contain very less or no interaction bearing words. However, the main documents are actually relevant to curatable. These kinds of miss-classification occur as we are only allowed to use the abstracts of the document/literature.
  - Informative trigger words such as *self-oligomerization* never appears in training document leading to mis-classification.
- Some non-curatable documents are also miss-classified as curatable because of:
  - There are interaction bearing words that could also appear in some other contexts. For e.g. in *GSK-3 inhibitors suppressed Sema4D-induced growth* , the word *inhibitors* does not appear here in the context of PPI. However, this a very strong evidence bearing word for PPI. As such the system is unable to properly identify the context leading to classification in curatable document.
  - Some documents are suspected to be PPI relevant because of the appearance of some of the triggers like *protein*. These kinds of words peak the probability of classifying the document in curatable category, however it belongs to the non-curatable set.

VI. Conclusions

In this paper we have presented a framework for protein-protein interaction article identification based on deep learning. At first we develop SVM based models using the handcrafted features, and word embedding features obtained from a large

TABLE I: Performance comparison of our proposed approach with baselines and other existing approaches. **Best System BC II:** denotes the best system submitted in BioCreative-II in terms of accuracy

Sr. No.	System	Approach	Precision	Recall	F-measure	Accuracy
1	Baseline-1	Single Word Feature (SW) + SVM	64.90	83.14	72.89	69.13
		Word Pair Feature (WP) + SVM	64.43	82.54	72.36	68.54
		SW + WP + SVM	65.84	86.09	74.61	70.75
		SW + WP + Dependency + SVM	74.60	73.10	72.53	72.98
		All Feature + SVM	75.10	73.33	72.80	73.26
2	Baseline-2	Word Vector averaging + SVM	76.70	75.50	75.20	75.48
3	CNN-static(IAS)	Convolution Neural Network(static)	**80.20**	80.19	80.19	**80.20**
4	CNN-non-static(IAS)	Convoluation Neural Network (non-Static)	79.48	79.16	79.11	78.88
5	Best System BC II[11]	Shallow Parsing + PoS + SVM	75.07	81.09	77.95	77.10
6	Lan et al. [15]	Majority voting + SVM	71.81	90.93	80.25	77.40
7	A.Cohen et al.[3]	Bag of words + SVM	68.64	86.40	76.51	–
8	William et al.[2]	Semantic Feature + SVM + Naive Bayes	67.70	85.10	72.20	66.80
9	Grover et al.[5]	bag of nlp + SVM	69.94	87.47	77.73	74.93

TABLE II: Performance of model variants with different window sizes; **P**: precision, **R**: recall, **F**: F-measure, and **Acc**: Accuracy

Sr No.	Window Size (h)	CNN-static(IAS)				CNN-non-static(IAS)			
		P	R	F	Acc	P	R	F	Acc
1	3	76.73	76.66	76.70	76.66	76.73	76.66	76.70	76.66
2	4	76.93	76.81	76.87	76.81	76.70	76.37	76.53	76.37
3	5	76.23	76.22	76.22	76.23	77.13	77.10	77.12	77.10
4	3, 4	78.49	76.68	77.57	76.66	77.76	77.40	77.58	77.40
5	4, 5	77.75	77.69	77.72	77.70	78.14	77.70	77.92	77.70
6	3, 4, 5	80.20	80.19	**80.19**	80.20	79.48	79.16	**79.11**	78.88
7	4, 5, 6	76.64	76.22	76.43	76.22	76.47	75.93	76.20	75.72
8	3, 4, 5, 6	77.50	77.40	77.45	77.40	77.17	76.96	77.07	76.96

corpus. We have developed two variants of the CNN model. Experiments were carried out on the both variants using several filter with single and multiple window-size. Experiments on a benchmark setup shows the efficacy of the proposed approach with considerable performance increments over the baselines and the existing systems. The advantage of the proposed model is that they do not make use of any handcrafted features for classification, but still achieve the best performance. The network itself learns the relevant set of features from the given documents. The proposed architecture is very generic in nature, and we would like to extend it for the other related tasks.

REFERENCES

[1] R. Ando. Proceedings of the second biocreative challenge evaluation workshop. 2007.

[2] W. A. Baumgartner Jr, Z. Lu, H. L. Johnson, J. G. Caporaso, J. Paquette, A. Lindemann, E. K. White, O. Medvedeva, K. B. Cohen, and L. Hunter. An integrated approach to concept recognition in biomedical text. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 257–71. Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain, 2007.

[3] A. Cohen. Automatically expanded dictionaries with exclusion rules and support vector machine text classifiers: approaches to the biocreative 2 gn and ppi-ias tasks. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, pages 169–174, 2007.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[5] C. Grover, B. Haddow, E. Klein, M. Matthews, L. A. Nielsen, R. Tobin, and X. Wang. Adapting a relation extraction pipeline for the biocreative ii task. In *Proceedings of the BioCreAtIvE II Workshop*, volume 2, 2007.

[6] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[8] T. Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[10] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[11] M. Krallinger, F. Leitner, C. Rodriguez-Penagos, A. Valencia, et al. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome biology*, 9(Suppl 2):S4, 2008.

[12] M. Krallinger and A. Valencia. Evaluating the detection and ranking of protein interaction relevant articles: the biocreative challenge interaction article sub-task (ias). In *Proceedings of the Second Biocreative Challenge Evaluation Workshop*, 2007.

[13] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273, 2015.

[14] M. Lan, C. L. Tan, and J. Su. A term investigation and majority voting for protein interaction article sub-task 1 (ias). In *Proceedings of the BioCreative II workshop; Madrid, Spain*, pages 183–185, 2007.

[15] M. Lan, C. L. Tan, and J. Su. Feature generation and representations for protein–protein interaction classification. *Journal of biomedical informatics*, 42(5):866–872, 2009.

[16] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[18] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[19] B. Settles. Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[20] W.-t. Yih, K. N. Toutanova, C. A. Meek, and J. C. Platt. Learning discriminative projections for text similarity measures, June 14 2011. US Patent App. 13/160,485.

[21] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.