Relation extraction from clinical texts using domain invariant convolutional neural network

Sunil Kumar Sahu$^{\Psi}$ [*], **Ashish Anand**$^{\Psi}$, **Krishnadev Oruganty**♣, **Mahanandeeshwar Gattu**♣
$^{\Psi}$Department of Computer Science and Engineering, IIT Guwahati, Assam, India
♣Excelra Knowledge Solutions Pvt Ltd, Hyderabad, Telangana, India
{sunil.sahu, anand.ashish}iitg.ernet.in
{krishnadev.oruganty, nandu.gattu}gvkbio.com

Abstract

In recent years extracting relevant information from biomedical and clinical texts such as research articles, discharge summaries, or electronic health records have been a subject of many research efforts and shared challenges. Relation extraction is the process of detecting and classifying the semantic relation among entities in a given piece of texts. Existing models for this task in biomedical domain use either manually engineered features or kernel methods to create feature vector. These features are then fed to classifier for the prediction of the correct class. It turns out that the results of these methods are highly dependent on quality of user designed features and also suffer from curse of dimensionality. In this work we focus on extracting relations from clinical discharge summaries. Our main objective is to exploit the power of convolution neural network (CNN) to learn features automatically and thus reduce the dependency on manual feature engineering. We evaluate performance of the proposed model on i2b2-2010 clinical relation extraction challenge dataset. Our results indicate that convolution neural network can be a good model for relation exaction in clinical text without being dependent on expert's knowledge on defining quality features.

1 Introduction

The increasing amount of biomedical and clinical texts such as research articles, clinical trials, discharge summaries, and other texts created by social network users, represents immeasurable source of information. Automatic extraction of relevant information from these resources can be useful for many applications such as drug repositioning, medical knowledge base creation etc. The performance of concept entity recognition systems for detecting mention of proteins, genes, drugs, diseases, tests and treatments has achieved sufficient level of accuracy, which gives us opportunity for using these data to do next level tasks of natural language processing (NLP). Relation extraction is the process of identifying how given entities are related in considered sentence or text. As given in the example sentence [S1] below, the entities *Lexix* and *congestive heart failure* are related by *treatment administered medical problem* relation. These relations are important for other upper level NLP tasks and also in biomedical and clinical research (Shang et al., 2011).

[S1]: *He was given* **Lexix** *to prevent him from* **congestive heart failure** .

Relation extraction task in unstructured text has been modeled in many different ways. *co-occurrence* based methods due to their simplicity and flexibility are most widely used methods in biomedical and clinical domain. In co-occurrence analysis it is assumed that if two entities are coming together in many sentences, their must be a relation between them (Bunescu et al., 2006; Song et al., 2011). Quite obviously this method can not differentiate types of relations and suffers from low precision and recall. To improve its results, different statistical measures such as point wise mutual information, chi-square or log-likelihood ratio has been used in this approach (Stapley and Benoit, 2000).

Rule based methods are another commonly adapted methods for relation extraction task (Thomas et al., 2000; Park et al., 2001; Leroy et al., 2003). Rules are created by carefully observing the syntactic and semantic patterns in rela-

[*] Part of this work was done while *Sunil Kumar Sahu* was doing internship at *Excelra Knowledge Solutions Pvt Ltd*, Hyderabad, Telangana, India.

Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pages 206–215,
Berlin, Germany, August 12, 2016. ©2016 Association for Computational Linguistics

tion instances. *Bootstrapping method* (Xu, 2008) is used to improve the performance of rule based methods. Bootstrapping uses small number of known relation pair of each relation type as a seed and use these seeds to search patterns in huge unannotated text (Xu, 2008) in iterative fashion. Bootstrapping method generates lots of irrelevant patterns too, which can be controlled by *distantly supervised* approach. Distantly supervised method uses large knowledge base such as UMLS or Freebase as an input and extract patterns from huge corpus for all pair of relations present in knowledge base (Mintz et al., 2009; Riedel et al., 2010; Roller and Stevenson, 2014). The advantage of bootstrapping and distantly supervised methods over supervised methods is that they do not require lots of manually labeled training data which is generally very hard to get.

Feature based methods use sentences with predefined entities to construct feature vector through feature extraction (Hong, 2005; Minard et al., 2011b; Rink et al., 2011). Feature extraction is mainly based on linguistic and domain knowledge. Extracted feature vectors are used to decide correct class of relation present between entities in the sentence through any classification techniques. *Kernel methods* are extension of feature based methods which utilize kernel functions to exploit rich syntactic information such as parse trees (Zelenko et al., 2003; Culotta and Sorensen, 2004; Qian and Zhou, 2012; Zeng et al., 2014). State of the art results have been obtained by these class of methods.

However, the performance of feature and kernel based methods are highly dependent on suitable feature set selection, which is not only tedious and time consuming task but also require domain knowledge and is dependent on other NLP systems. Often such dependencies make many existing work less reproducible simply because of absence of the full and finer details of feature extraction. Further often these methods lead to huge number of features and may get affected from curse of dimensionality issues (Bengio et al., 2003; Collobert et al., 2011). Another issue faced by these methods is feature extraction will have to be adjusted according to the data source. As discussed earlier we are having multiple but diverse information resources such as research articles, discharge summaries, clinical trials outcome etc. While in one hand multiple sources bring more information but the other hand it makes it challenging to extract meaningful information automatically simply because of diverse nature of the data source. For example, sentences in research articles are well formed and likely to use only well accepted technical terms. But sentences in clinical discharge summaries may not be well formed sentences instead it could be fragmented sentences with lots of acronyms or terms used only locally. Similarly social media articles may use slang or terms which are not technically used. This makes it difficult for above discussed methods.

Motivated by these issues, this work aims to exploit recent advances in machine learning and NLP domains to reduce such dependencies and utilize convolutional neural network to learn important features with minimal manual dependencies. Convolution neural network has shown to be a powerful model for image processing, computer vision (Krizhevsky et al., 2012; Karpathy and Fei-Fei, 2014) and subsequently in natural language processing it has given state of the art results in different tasks such as sentence classification (Kim, 2014; Kalchbrenner et al., 2014; Hu et al., 2014; Sharma et al., 2016), relation classification (Zeng et al., 2014; dos Santos et al., 2015) and semantic role labeling (Collobert et al., 2011).

In this paper we propose a new framework for extracting relations among *problem*, *treatment* and *test* in clinical discharge summaries. In particular we use data available under clinical relation extraction task organized by Informatics for Integrating Biology and the Bedside (i2b2) in 2010 as part of i2b2/VA challenge (Uzuner et al., 2011). Extracting relations in clinical texts is more challenging compared to research articles as it contains incomplete or fragmented sentences, and lots of acronyms. Current state of the art methods heavily depend on manual feature engineering and use hundreds of thousands of features (Minard et al., 2011b; Rink et al., 2011). Our result indicates the proposed model can outperform the current state of the art models by using only a small fraction of features. However the main observation is the features used in our model is easy to replicate and adapt as per the data source compared to the feature sets generally used in these tasks.

2 Related Research

i2b2 organized a shared task in 2010 (Uzuner et al., 2011). In this challenge discharge sum-

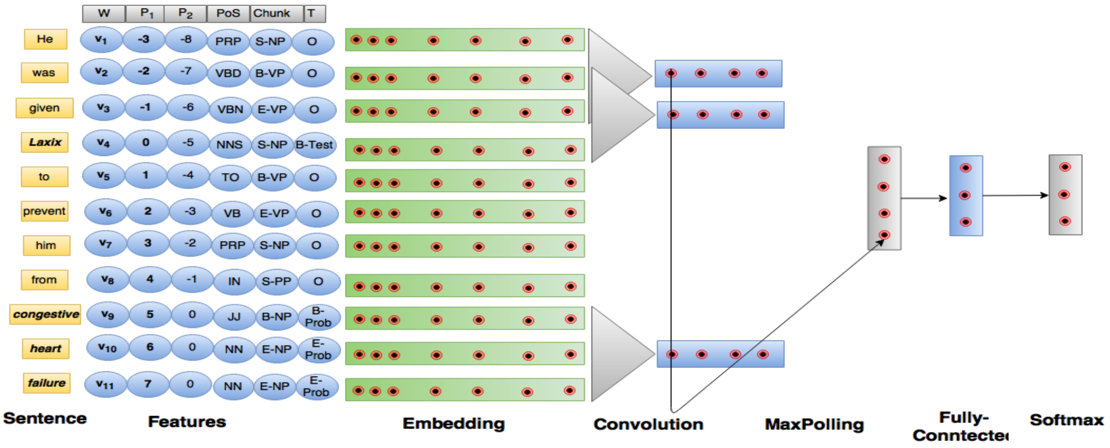

Figure 1: CNN model for relation extraction.

maries from three different sources were annotated for extracting relations among clinical entities such as *problem*, *treatment* and *test*. Most of the participants in this challenge used support vector machine (SVM) with manually designed features (Uzuner et al., 2010). Model proposed by Rink et al. (2011) had first place in this task, which used six classes of features namely, context features, similarity features, nested related relation features, Wikipedia features, single concept features and vicinity features. They formulated the relation extraction task as a multiclass classification problem and SVM with linear kernel were used for classification.

For extracting relation among disease and treatment, Rosario and Hearst (2004) used various graphical and neural network models. They used variety of lexical, semantic and syntactic features for classification and found that semantic features were contributing most among all. The dataset used in this study was relatively smaller and was prepared from biomedical research articles. Li et al. (2008) proposed kernel methods for relation extraction between entities in MEDLINE® articles. They modified the tree kernel function by incorporating trace kernel to capture richer contextual features for classifying the relation. Their results shows that tree kernel outperform other kernel methods such as word and sequence kernels for the considered task.

Conditional random field (CRF) has been used for relation extraction between disease treatment and gene by (Bundschus et al., 2008). In this experiment setting, they did not assume that entities were given, instead their model also predicted entities and its type. They developed two variants of CRF both modeled relation extraction task as sequence labeling task. Recently Bravo et al. (2015) proposed a system for identifying association between drug disease and target in EU-ADR dataset (van Mulligen et al., 2012) and named it BeeFree. BeeFree usese combination of shallow linguistic kernel and dependency kernel for identifying relations.

In contrast to above methods recently there are few work applying convolution neural network based models (Zeng et al., 2014; dos Santos et al., 2015) for *relation classification* in SemEval 2010 relation classification dataset (Hendrickx et al., 2009). Convolution neural network used in this models are using constant length filters, and word embedding and distance embedding as features. Our model leverage on the linguistic features also and we considered *relation extraction* task in clinical notes which is much more informal, rich with acronyms and number of samples for each relations are not stable (Uzuner et al., 2011).

3 CNN for Clinical Relation Extraction

The proposed model based on CNN is first summarized in the next section. Subsequent sections describe it in more detail.

3.1 Model Architecture

The proposed model architecture is shown in the figure 1, which takes a complete sentence with mentioned entities as an input and outputs a probability vector corresponding to all possible relation types. Each feature is having vector representation which is initialized randomly except word embedding feature. For word embedding, we used pre-trained word vector (TH et al., 2015) learned on Pubmed articles using word2vec tool (Mikolov et al., 2013b).

Embedding layer maps every feature value with its corresponding feature vectors and concatenate them. In order to get local features from each part of the sentence we have used multiple filters of different lengths (Kim, 2014) in all possible continuous n-gram of the sentence, where n is the length of filter (We have shown four filters with constant length three in the figure 1). We use max pooling over time to get global features through all filters. Here time indicates filter running over the length of the sentence. Pooled features are then fed to fully connected feed-forward neural network to make inference. In the output layer we use softmax classifier with number of outputs equal to number of possible relations between entities.

3.2 Feature Layer

We represent each word in the sentence with 6 discrete features namely word itself (W), distance from the first entity (P_1), distance from the second entity (P_2), parts of speech tag of the word (PoS), chunk tag of the word (Chunk) and entity type (T). Each feature is briefly described below:

1. W : Exact word appeared in the sentence.

2. P_1: Distance from the first entity in terms of number of words (Collobert and Weston, 2008). For instance in our earlier example [S1] *He* is at -3 distance and *prevent* is at $+2$ distance away from the first entity *Lexis*. This value would be zero for all words which is a part of the first entity.

3. P_2: Similar to P_1 but considers distance from the second entity.

4. PoS: Parts of speech tag of the considered word. We use genia tagger[1] to obtain pos tag of each word.

5. $Chunk$: Chunk tag of considered word. Again we use genia tagger to obtain chunk tag of each word.

6. T: Type of the considered word. For example, it would be entity type such as $B-Prob$, $I-Prob$ etc. for entity word and $Other$ for rest words following the *BIO* tagging convention.

This way a word $w \in D^1 \times D^2 \times .....D^6$, where D^i is the dictionary for i^{th} local features.

3.3 Embedding Layer

In lookup or embedding layer each feature value is mapped to its vector representation using feature embedding matrix. Lets say $M^i \in \mathbb{R}^{n \times N}$ is the feature embedding matrix for i^{th} local feature (here n represents dimension of feature embedding and N is number of possible values or size of the dictionary for i^{th} local feature). Each column of M^i is vector of corresponding value of i^{th} features. Mapping can be done by taking product of one hot vector of feature value with its embedding matrix (Collobert and Weston, 2008). Suppose $a_j^{(i)}$ is the one hot vector for j^{th} feature value of i^{th} feature then:

$$f_j^{(i)} = M^i \, a_j^{(i)} \tag{1}$$

$$x^i = f_1^{(i)} \oplus f_2^{(i)} .... \oplus f_6^{(i)} \tag{2}$$

Here $\oplus$ is concatenation operation so $x^i \in \mathbb{R}^{(n_1 + ...n_6)}$ is feature vector for i^{th} word in sentence and n_k is dimension of k^{th} feature. For word embedding we used pre-trained word vector obtained after running word2vec tool (Mikolov et al., 2013b; Mikolov et al., 2013a) on huge Pubmed open source articles (TH et al., 2015). Other feature matrix were initialized randomly at the beginning. Since number of elements in all feature dictionary except word dictionary (D^1) are not huge, we assume that while training these vectors will get sufficient updation.

3.4 Convolution Layer

We apply convolution on text to get local features from each part of the sentence (Collobert and Weston, 2008). Consider $x^1 x^2 .....x^m$ is the sequence of feature vectors of a sentence, where $x^i \in \mathbb{R}^d$ is a vector obtained by concatenating all feature vector of i^{th} word. Let $x^{i:i+j}$ represents concatenation of $x^i .....x^{i+j}$ feature vectors. Suppose there is a *filter* parameterized by weight vector $w \in \mathbb{R}^{cd}$ where c is the length of filter (in figure 1 filter length is three). Then output sequence of convolution layer would be

$$h^i = f(w \cdot x^{i:i+c-1} + b) \tag{3}$$

[1] http://www.nactem.ac.uk/GENIA/tagger/

Where $i = 1, 2, \ldots m - c + 1$, . is dot product, f is rectify linear unit (ReLu) function and $b \in \mathbb{R}$ is biased term. w and b are the learning parameters and will remain same for all $i = 1, 2, \ldots m-c+1$.

3.5 Max Pooling Layer

Output of convolution layer length $(m-c+1)$ will vary based on number of words m in the sentence. We applied max pooling (Collobert and Weston, 2008) over time to get fixed length global features for whole sentence. The intuition behind using max pooling is to consider only most useful feature from entire sentence.

$$z = \max_{1 \le i \le (m-c+1)} [h^i] \tag{4}$$

We have just explained the process of extracting one feature from a whole sentence using one filter. In figure 1 we extracted four features using four filters of the same length three. In our experiment we use multiple such filters of variable length (Kim, 2014; Yin and Schtze, 2015). The objective of using different length filter is to accommodate context in varying window size around words.

3.6 Fully Connected Layer

The output of max pooling layer is sequence z came with different filters. We call this global feature because it came by taking max over entire sentence. To make classifier over extracted global feature, we used fully connected feed forward layer. Suppose $z^i \in \mathbb{R}^l$ is output of max pooling layer for entire filters then output of fully-connected layer would be

$$o^{(i)} = W^o z^i + b^o \tag{5}$$

Here $W^o \in \mathbb{R}^{[r] \times l}$ and $b^o \in \mathbb{R}^{[r]}$ are parameters of neural network and $[r]$ denotes number of classes.

3.7 Softmax Layer

In output layer we used softmax classifier for which objective function would be minimization of

$$L_i = -\log\left(\frac{e^{o^{(i)}_{y_i}}}{\sum_{\forall j} e^{o^{(i)}_j}}\right) \tag{6}$$

for i^{th} sentence. Here y_i is correct class of relation for i^{th} instance.

3.8 Implementation

We experiment with filter lengths in two different experiment settings. In first, we use 100 different filters of a fixed length in the convolutional layer, while in another set of experiments we use varying length filters, but used 100 different filters for each varying length. So, in the first setting, we obtain 100 features after max pooling, while in the second, we obtain 100 times number of different length filter features. For regularization (Srivastava et al., 2014), we follow (Kim, 2014) and use *dropout* technique in output of max pooling layer. Dropout prevents co-adaptation of hidden units by randomly dropping few nodes. We set this value to 0.5 during training and 1 while testing. We use Adam technique (Kingma and Ba, 2014) to optimize our loss function. Entire neural network parameters and feature vectors are updated while training. We have implemented the proposed model in Python language using tensorflow package (Abadi et al., 2015) and will make it available on request. Results of each filter length were explained in results section. Dimension of word vector is set to 50 and rest all feature embedding size is kept to 5.

4 Dataset and Experimental Settings

In recent years several challenges have been organized to automatically extract information from clinical texts (Uzuner et al., 2007; Uzuner et al., 2008; Uzuner et al., 2011; Uzuner et al., 2010; Sun et al., 2013). i2b2 has released dataset for clinical concept extraction, assertion classification and relation extraction as a part of i2b2-2010 shared task challenge. This dataset was collected from three different hospitals and was manually annotated by medical practitioners for identifying problems, treatments and test entities, and eight relation types among them. These relations were: *treatment caused medical problems* (**TrCP**), *treatment administered medical problem* (**TrAP**), *treatment worsen medical problem* (**TrWP**), *treatment improve or cure medical problem* (**TrIP**), *treatment was not administered because of medical problem* (**TrNAP**), *test reveal medical problem* (**TeRP**), *Test conducted to investigate medical problem* (**TeCP**), *Medical problem indicates medical problems* (**PIP**). (Uzuner et al., 2011) has given the exact definition of each relation type.

While during the challenge original dataset had 394 documents for training and 477 documents for testing but when we downloaded this dataset from i2b2 website we got only 170 documents for training and 256 documents for testing. After preliminary experiment we found that we did not have

Name	Number instances
TeCP	503
TrCP	525
PIP	2202
TrAP	2616
TeRP	3052
No Relation	55600

Table 1: Relation types and number of instances of i2b2 dataset (partial)

enough training samples for all relation classes present in the dataset, therefore we decided to remove 3 relation classes along with their instances (*TrWP* (132 instances), *TrIP* (202 instances) and *TrNAP* (173 instances)). Statistics of the dataset is shown in the Table 1.

For extracting relations among entities we considered all sentences having more than one entities in each discharge summary to check whether any relation exists between them or not. In our experiment we assume that entities and their types are already known like other existing works (Rink et al., 2011; Minard et al., 2011a; Minard et al., 2011b). We created data sample for every pair of entities present in the sentence and labeled it with the existing relation type. For example in sentence [S2] (all continuous bold phrases are entities) entity pairs (*"her white count", "elevated"*) label would be *"TeRP"*, for entity pair (*"her g-csf", "elevated"*) label would be *"TrNAP"* and for (*"her white count", "her G-CSF"*) label would be "None".

[S2]: **Her white count** *remained* **elevated** *despite discontinuing* **her G-CSF** .

5 Results and Discussion

5.1 Influence of filter lengths

We combined the training and testing data and performed five-fold cross-validation on the available limited i2b2 dataset for all our evaluations. First we evaluate the influence of filter lengths. We experiment with selection of filter length using all features. Results as average of five-fold experiment are shown in the Table 2.

In case of single filter, the results indicate increasing the size of filter length generally tends to improve the performance. Using only single filter the best performance with F1 score as 70.43% was obtained by using filter length of 6. However further increasing the filter length did not improve the result. Intuitively it also seems that selection of either of too small or too large filter length may not be a good option. Filter length gives the window size to capture context features. One can expect that too small filter length (window size) may not capture enough good context feature and too big filter length may include noise or irrelevant contexts.

Filter length	Precision	Recall	F Score
[3]	74.54	64.29	68.44
[4]	74.90	65.50	69.19
[5]	76.17	64.68	69.61
[6]	76.05	66.56	70.43
[7]	76.76	64.49	69.23
[3,4]	74.96	64.65	68.91
[3,5]	74.66	66.81	70.10
[4,5]	74.90	68.20	70.91
[4,6]	**76.34**	**67.35**	**71.16**
[5,6]	76.08	65.31	69.77
[3,4,5]	75.83	65.10	69.30
[4,5,6]	76.12	65.68	70.15
[2,3,4,5]	74.99	65.19	69.34
[3,4,5,6]	75.88	65.98	70.13

Table 2: Comparative performance of the proposed model using filters of different lengths separately and together. Each of the models used all features (WV+P_1+P_2+PoS+Chunk+Type) and 100 filters for each filter length.

Further, we used multiple filters to see whether it improves the result. Results indicate that combination of small and mid-length filter size is perhaps the better choice. For example, combination of filter lengths 3 and 4 together did not improve the performance compared to the single filter length of 3 or 4. On the other hand combination of filter lengths 3 and 5, and 4 and 5 improved the performance compared to use of single filters of either length. It can be seen, the best result with F1 score as 71.16% is obtained by using filter lengths of 4 and 6 together. But adding more than two filters did not lead to performance improvement.

5.2 Classwise Performance

We took the best combination of filter lengths and looked at the classwise performance. Results are described in the Table 3.

We see from the results that as number of training examples (see Table 1) increases, performance of the model also improves. The relation class

Name	Precision	Recall	F Score
TeCP	63.48	43.67	50.56
TrCP	63.60	43.67	56.44
PIP	67.32	63.30	64.92
TrAP	73.49	65.83	69.23
TeRP	82.74	79.88	81.25

Table 3: Class wise performance with all features (filter size : [4,6] each with 100 filters)

TeRP has the maximum number of training examples and the model obtained quite a good F1 score. On the other hand, the model could not perhaps able to learn well for the relation classes *TeCP* and *TrCP* having relatively lesser number of training examples.

5.3 Contribution of Each Features

In order to investigate the contribution of each feature in final result we gradually include one feature in our model and compared the performance. Table 4 shows the obtained results. First we use only random vector (RV) representation along with entity types (T) (first row in the table) as a baseline for our comparison. Adding position features (2nd row) lead to approximately 15% increase in recall, 7% in precision and 11.7% in F1 score. However including PoS and Chunk features although improved recall and F1 score by 4.3% and 1.3% but precision was decreased by 3.6%. In the second set of experiments, we first use pre-trained word vectors along with entity types (4th row) and later repeated the similar experiments as previously. Here again, inclusion of position features improved the recall by more than 14% and F1 score by around 11%. This clearly indicates word position relative to the entities of interest plays important role in deciding their influence in the context. Further including PoS and Chunk features also led to performance improvement.

Name	P	R	F
RV + T	67.21	52.97	57.87
+*(P_1+P_2)*	71.86	60.69	64.66
+*(PoS+Chunk)*	69.25	63.34	65.52
WV + T	70.75	59.17	63.82
+*(P_1+P_2)*	75.54	67.69	70.97
+*(PoS+Chunk)*	**76.34**	**67.35**	**71.16**

Table 4: Contribution of each features (filter size : [4,6] each with 100 filters)

5.4 Comparison with Feature Based Method

We could not compare our results directly with the state of the art results obtained on the i2b2 dataset as we did not have the complete dataset. We build a linear SVM classifier using similar features as defined in earlier studies (Rink et al., 2011) as a baseline for comparison. The following features are used for each entity pair instance:

- Any word between relation arguments
- Any PoS tags between relation arguments. We used genia tagger for PoS
- Any bigram between relation arguments
- Word preceding first and second argument
- Any three words succeeding the first and second arguments
- Sequence of $chunk$ tags between relation arguments. We used genia tagger for $chunk$ tag
- String of words between relation arguments
- First and second argument type (problem, treatment and test)
- Order of argument type appeared in sentence
- Distance between two arguments in terms of number of words
- Presence of only punctuation sign between arguments.

This way we prepared attribute-value and numerical features for each instances. Table 5 shows the comparison of best results obtained by the proposed model and SVM based model. Linear SVM classifier with different cost parameter C was implemented using scikit learn (Pedregosa et al., 2011). Here again results shown are average over the 5-folds.

Name	P	R	F
CNN (FL=[4,6])	**76.34**	**67.35**	**71.16**
SVM (Linear, C=0.01)	72.23	57.75	58.96
SVM (Linear, C=0.1)	73.75	64.18	67.35
SVM (Linear, C=1)	73.17	64.18	67.32

Table 5: Comparative performance of SVM and CNN with filter length [4,6] each with 100 filters

Based on the results, We can make following observations:

- Instead of SVM, other classifier could have been also used. We decided to use SVM as SVM based model with similar features obtained the best performance in the 2010 challenge.

- In any case we still would have to define huge number of features and only few of them would have non-zero values in any given sample or instance.

- The proposed model with limited number of features (75 * number of words in the sentence; 5 dimensional vector for 5 features other than word embedding, which is 50 dimensional vector) still gave the better performance.

- Consistent with our observations in the section 5.1, too many features trying to capture more contexts adversely affect the performance of classifier. If we look at the features defined above it includes features which try to capture context of all possible window size between the mentioned entities.

6 Conclusion

In this work we present a new framework based on CNN for extracting relations among clinical entities in clinical texts. The proposed model has shown better performance by using only a small fraction of features compared to the SVM based baseline model. Our results indicate that CNN is able to learn global features which can capture contextual features quite well and thus helps in improving the performance.

Acknowledgments

We would like to thank i2b2 National Center for Biomedical Computing funded by U54LM008748, for providing the clinical records originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow.org*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. 2015. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC bioinformatics*, 16(1):1.

Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. 2008. Extraction of semantic biomedical relations from text using conditional random fields. *BMC bioinformatics*, 9(1):1.

Razvan Bunescu, Raymond Mooney, Arun Ramani, and Edward Marcotte. 2006. Integrating cooccurrence statistics with information extraction for robust retrieval of protein interactions from medline. In *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pages 49–56. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics.

Cıcero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 626–634.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Gumwon Hong. 2005. Relation extraction using support vector machine. In *Natural Language Processing–IJCNLP 2005*, pages 366–377. Springer.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*, pages 2042–2050.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Andrej Karpathy and Li Fei-Fei. 2014. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. pages 1097–1105.

Gondy Leroy, Hsinchun Chen, and Jesse D Martinez. 2003. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of biomedical Informatics*, 36(3):145–158.

Jiexun Li, Zhu Zhang, Xin Li, and Hsinchun Chen. 2008. Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5):756–769.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Anne-Lyse Minard, Anne-Laure Ligozat, Asma Ben Abacha, Delphine Bernhard, Bruno Cartoni, Louise Deléger, Brigitte Grau, Sophie Rosset, Pierre Zweigenbaum, and Cyril Grouin. 2011a. Hybrid methods for improving information access in clinical documents: concept, assertion, and relation identification. *Journal of the American Medical Informatics Association*, 18(5):588–593.

Anne-Lyse Minard, Anne-Laure Ligozat, and Brigitte Grau. 2011b. Multi-class svm for relation extraction from clinical reports. In *RANLP*, pages 604–609.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jong C Park, Hyun Sook Kim, and Jung-Jae Kim. 2001. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. In *Pacific Symposium on Biocomputing*, volume 6, pages 396–407.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Longhua Qian and Guodong Zhou. 2012. Tree kernel-based proteinprotein interaction extraction from biomedical literature. *Journal of Biomedical Informatics*, 45(3):535 – 543.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Bryan Rink, Sanda Harabagiu, and Kirk Roberts. 2011. Automatic extraction of relations between medical concepts in clinical texts. *Journal of the American Medical Informatics Association*, 18(5):594–600.

Roland Roller and Mark Stevenson. 2014. Applying umls for distantly supervised relation detection. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, pages 80–84, Gothenburg, Sweden, April. Association for Computational Linguistics.

Barbara Rosario and Marti A. Hearst. 2004. Classifying semantic relations in bioscience texts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yue Shang, Yanpeng Li, Hongfei Lin, and Zhihao Yang. 2011. Enhancing biomedical text summarization using semantic relation extraction. *PLoS ONE*, 6(8):1–10, 08.

Ranti D Sharma, Samarth Tripathi, Sunil K Sahu, Sudhanshu Mittal, and Ashish Anand. 2016. Predicting online doctor ratings from user reviews using convolutional neural networks. *International Journal of Machine Learning and Computing*, 6(2):149.

Qiang Song, Yousuke Watanabe, and Haruo Yokota. 2011. Relationship extraction methods based on co-occurrence in web pages and files. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, iiWAS '11, pages 82–89, New York, NY, USA. ACM.

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.

Benjamin J Stapley and Gerry Benoit. 2000. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in medline abstracts. In *Pac Symp Biocomput*, volume 5, pages 529–540.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5).

MUNEEB TH, Sunil Sahu, and Ashish Anand. 2015. Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of BioNLP 15*, pages 158–163, Beijing, China, July. Association for Computational Linguistics.

James Thomas, David Milward, Christos Ouzounis, Stephen Pulman, and Mark Carroll. 2000. Automatic extraction of protein interactions from scientific. In *Pacific symposium on biocomputing*, volume 5, pages 538–549.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. 2008. Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association*, 15(1):14–24.

Özlem Uzuner, Imre Solti, and Eithon Cadag. 2010. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17(5):514–518.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Erik M. van Mulligen, Annie Fourrier-Reglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifiro, Jan Kors, and Laura I Furlong. 2012. The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of biomedical informatics*, 45(5):879–884.

Fei-Yu Xu. 2008. *Bootstrapping Relation Extraction from Semantic Seeds*. Ph.D. thesis, Saarland University.

Wenpeng Yin and Hinrich Schtze. 2015. Multichannel variable-size convolution for sentence classification. In Afra Alishahi and Alessandro Moschitti, editors, *CoNLL*, pages 204–214. ACL.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2335–2344.