Attention-based Neural Networks for Chemical Protein Relation Extraction

Sijia Liu¹², Feichen Shen¹, Yanshan Wang¹, Majid Rastegar-Mojarad¹³, Ravikumar Komandur Elayavilli¹, Vipin Chaundary², Hongfang Liu¹

¹Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA
²Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA
³University of Wisconsin-Milwaukee, Milwaukee, WI, USA

Abstract— Relation extraction is an important task in the field of natural language processing and text mining. In this paper, we described our participation in Biocreative VI Task 5: Text mining chemical-protein interactions (CHEMPROT). We used deep neural networks, including convolutional neural networks and attention based recurrent neural networks, to extract chemical protein relationships. We submitted four runs as final submission, with the best performing micro average F1 score of 0.506 on the development set and 0.494 on the test set. Our experimental results indicate that attention based RNN models outperform the CNN models. The replacement of raw texts into semantic types (chemical, gene) also improved the performance.

Keywords—relation extraction, natural language processing, recurrent neural networks, attention mechanism

I. INTRODUCTION

Efficient and accurate access to the information from scientific literature is among the most widely studied topics in text mining. To facilitate and promote information extraction systems on PubMed abstracts, Biocreative VI CHEMPROT task aims to automatically detect the relations among chemical compounds, genes and proteins. There have been several chemical- or drug-related relation extraction share tasks organized in recent years. Two DDI-Extraction share tasks were organized to extract the drug-disease interaction information from knowledge base (DrugBank) and scientific articles (Medline) in 2011 and 2013, respectively [1]. In 2015, Biocreative V Track 3 targeted the extraction of relationships in Chemical-Induced Diseases (CID) [2]. Chemical Protein interaction (Chempot) is an interesting topic for biomedical relation extraction [3].

Deep Neural Network (DNN) models have been used for relation extraction tasks recently, including Convolutional Neural Networks (CNN) [4] and Recurrent Neural Networks (RNN) [5]. Moreover, attention mechanism on top of deep neural networks has shown promising results in various Natural Language Processing (NLP) tasks, such as machine translation [6], question answering [7] and document classification [8]. Attention based model has been used in multiple relation extraction tasks [9][10]. In this paper, we described our participation in Biocreative VI Task 5: Text mining chemical-protein interactions (CHEMPROT). We used deep neural networks, including CNN and attention based RNN to extract chemical protein relationships. We achieved a micro average F1 score of 0.506 on the development set and 0.494 on the test set as our best result. The experimental results indicated that attention based Gated Recurrent Units (GRU) outperforms other DNN models.

II. MATERIALS

We used the ChemProt corpus provided by task organizers. The corpus consisted of 4966 PubMed abstracts with 126,457 annotated chemical and protein entities. The relations were annotated with 10 chemical-protein relations. According to the share task description, only 5 out of 10 semantic relation types would be evaluated. Therefore, we focused only on the relation groups included in the official evaluation (CPR 3, 4, 5, 6 and 9).

Table I shows the corpus statistics of the training, development and testing datasets, including the number of documents in the dataset, the average number of entities per document (abstract) and the average number of positive relations per document. The test set gold standard relation annotation was not available by the time we prepared the manuscript. More details of the datasets can be found in the CHEMPROT task description paper.

III. METHOD

We used DNN for the task of relation extraction on the annotated named entity mentions. We model the relation extraction problem as a relation classification problem among all the chemical–protein relation pairs within one sentence. The rationale of limiting the relations into within-sentence relations is that we only found three and six cross-sentence relations from training and development sets, respectively. Therefore, on
the CHEMPROT corpus, eliminating all cross-sentence relations will not impact the performance significantly.

A. Relation Instance Generation

We extract the sentences from sentence detectors. For each potential chemical protein pair in the sentence, we assign a relation label “NA” for the pair without annotated gold standard annotation provided by the challenge organizers. Here we consider the relations other than the five evaluated types as negative relations (CPR 1, 2, 7, 8 and 10).

B. Input Representation

In our proposed system, the input to the neural network models is expected to be low dimension semantic token-level vectors. We use word embeddings and position embeddings as the input to the neural network models.

We use two different methods to represent the pre-annotated entities. In the first method, the tokens are directly sent to the word embedding model to retrieve the word embeddings, regardless if it is an annotated token as part of an entity. The limitation for this method is that a lot of chemical, gene and protein entities are not found from the pretrained model. We then apply a second representation. We replace all the entity tokens by the entity types. Specifically, for chemical entity mentions, all the tokens of entities are replaced by “chemical”. Similarly, all the gene and protein entity mentions, all the tokens of entities are replaced by “gene”, regardless of whether it can be normalized or associated to a biological database identifier (i.e. “GENE-Y” or “GENE-N”).

Word embeddings. We use 300-dimension Glove pre-trained model. Our preliminary experiments show that the Glove-6B outperforms the word embedding models trained by CBOW from PubMed. If a word cannot be found from the word embedding model, the embedding will be generated randomly and appended into the model.

Position embeddings. We follow the method by Zeng et al. [4] to generate the position embedding of the entities in each narrative sentence. The position embedding is generated based on the relative distances of tokens to the entities. An example of relative distance is shown in Figure 2.

For each relation instance, there will be two position embeddings for each token from two entities (one chemical and one gene/protein). The two position embeddings are concatenated to the word embeddings of the token as the input to the neural network models.

C. Convolutional neural network

Our CNN models for relation extraction are similar to [4]. The model architecture is shown in Figure 1 (a). The convolutional layer can capture contextual information of filters of a pre-defined filter length. The convolutional filters are expected to generate high-level local features from the input vector representations. The output of the convolutional layer is then forwarded to the Global Max-pooling layer, where the maximum values of each filter outputs are pooled and concatenated. Finally, a non-linear function can be used on the max-pooling vector to predict the probability-like values of each relation label. We then select the label with the highest value from the non-linear layer as the relation label.

D. Attention based RNN

Attention mechanism is proposed to emphasize the contribution of the informative neural units in the model. Instead of directly receiving the signals from consecutive RNN units, the additional attention layer overlooks all the RNN units of the sequence and assigns different attention weights to each unit according to their importance. The intuition for applying attention-based model is to have higher weights for tokens that are indicators of the semantic relations. We use the equations from Luong, et. al. [6] to calculate attention weights for each token in a sentence. The output of the RNN units is denoted as \( h = [h_1, h_2, ..., h_T] \), where \( T \) is the sentence length. Given token representation as \( w \), we define the hidden representation of \( h_t \) as \( u_t \) and word importance vector \( u_w \). The relation representation vector \( s \) is the weighted sum of RNN outputs \( h \) and the attention weights \( \alpha \).

\[
\begin{align*}
    u_t &= \tanh(W_w h_t + b_w) \\
    \alpha_t &= \frac{\exp(u_t^T u_w)}{\sum_t \exp(u_t^T u_w)}
\end{align*}
\]
\[ s = \sum \alpha_t h_t \]

where \( W_w \) and \( b_w \) are the weight matrix and bias of the attention layer, respectively. We then apply the non-linear layer and relation prediction method as described in the CNN model (Section III.C) to predict relation labels.

### E. Evaluation

The model performance is evaluated via standard micro-average F1-score. True Positive (TP) denotes the number of correctly detected positive relation instances. False Positive (FP) denotes the number of relations “NA” in the gold standard but are predicted as one of the positive relations by the DNN model. False Negative (FN) denotes the number of positive instances that are not detected by the model. The micro-average F1-score is defined as:

\[
\text{Precision} = \frac{TP}{TP + FP} \\
\text{Recall} = \frac{TP}{TP + FN} \\
F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

The system performance is evaluated by the official evaluation package provided by the task organizers\(^1\).

### IV. RESULT

#### A. Experimental settings

We used Punkt sentence detector in NLTK [11], [12] to detect sentence boundaries. The models are implemented using Keras 2.0.5\(^2\) with Tensorflow\(^3\) backend. The implementation of attention layer is inspired by [13]. We applied the hyperparameters shown in Table II for the CNN and ATTention-based GRU model (ATT-GRU). The models are trained using Adam optimizer on the loss function of sparse categorical cross entropy. Dropout was applied to the non-linear layers to prevent overfitting [14]. The dropout rate was set to 0.5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>64</td>
</tr>
<tr>
<td>Number of filters</td>
<td>100</td>
</tr>
<tr>
<td>Filter length</td>
<td>3</td>
</tr>
<tr>
<td>Hidden dimensions</td>
<td>200</td>
</tr>
<tr>
<td>RNN units</td>
<td>128</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table II: Parameter settings for CNN and ATT-GRU models

#### B. Submitted Runs

We submitted four runs to the challenge evaluation. The details of each run is shown as follows:

- **Run 1**: CNN with raw tokens as input, without replacing annotated entity tokens.
- **Run 2**: ATT-GRU raw tokens as input, without replacing annotated entity tokens.
- **Run 3**: CNN with raw tokens replaced as labels.
- **Run 4**: ATT-GRU with raw tokens replaced by entity labels.

We also implemented conventional RNN, GRU, LSTM, and attention based LSTM as preliminary experiments. However, the performance was not as good as the performance of ATT GRU models. Due to the limitation of the number of submission runs, we did not include them into the official submissions.

Table IV shows the system performance of each submitted run in the development dataset. Based on our experimental results, the attention based GRU on entity labels outperforms the rest of runs with a F1 score of 0.506. Attention based GRU systems outperform CNN models. Replacing raw entity texts by the entity labels enhances the models slightly for both CNN and ATT-GRU. It is mainly due to the fact that most of the chemical, gene and protein tokens cannot be found in the pre-trained word embedding models. Thus, replacing them into meaningful entity labels (“chemical”, “gene”) can help find reasonable word embeddings instead of random assigned ones. The models generally do not suffer much from overfitting. All the submission runs observed F1 score decreases of less than 0.01.

The performance breakdown of our best run (ATT-GRU) on development set is shown in Table IV. The classification report is done via scikit-learn\(^4\). CPR:4 has the highest performance among all the classes, and has the largest proportion in all relations as well. CPR:3 is one of the hardest relation type to classify, not only for the ATT-GRU model but also for our other models.

#### Table III: Performance Breakdown of ATT-GRU on the Development Set

<table>
<thead>
<tr>
<th>Label</th>
<th>Support</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPR:3</td>
<td>498</td>
<td>0.473</td>
<td>0.388</td>
<td>0.426</td>
</tr>
<tr>
<td>CPR:4</td>
<td>990</td>
<td>0.569</td>
<td>0.663</td>
<td>0.613</td>
</tr>
<tr>
<td>CPR:5</td>
<td>112</td>
<td>0.357</td>
<td>0.634</td>
<td>0.457</td>
</tr>
<tr>
<td>CPR:6</td>
<td>184</td>
<td>0.505</td>
<td>0.609</td>
<td>0.552</td>
</tr>
<tr>
<td>CPR:9</td>
<td>407</td>
<td>0.468</td>
<td>0.442</td>
<td>0.455</td>
</tr>
<tr>
<td>Total *</td>
<td>2191</td>
<td>0.512</td>
<td>0.553</td>
<td>0.528</td>
</tr>
</tbody>
</table>

* The “Total” line is calculated from weighted F1-score by support counts, which is different from the micro-F1 score of the official evaluation.

#### V. Conclusion and Future Work

In this paper, we described our participation of Task 5: Text mining chemical-protein interactions (CHEMPROT). We developed deep neural network models including CNN and attention-based RNN using the embedding of either raw entity mention tokens or annotated entity type labels as input. The model using annotated entity type labels and attention-based GRU model achieves the best performance on both the

\(^1\) http://www.biocreative.org/media/store/files/2017/evaluation-kit.zip
\(^2\) https://keras.io/
\(^3\) https://github.com/tensorflow/tensorflow
\(^4\) http://scikit-learn.org/stable
development and the test set, with F1-scores of 0.506 and 0.494, respectively.

There are some directions to investigate to improve this work. We would like to see if an external knowledge base can be used to improve our machine learning based system, which is dependent on the provided corpus. Another interesting problem is how to utilize the token-level weights from the attention activations. The weights can be further applied for tasks such as pattern mining and key word extraction. The current system is proposed for within-sentence relations. The model could be further extended into abstract level by introducing hierarchical neural network models [8].

ACKNOWLEDGMENT

The authors would like to thank the support of the U.S. National Institutes of Health grant RO1 LM011829 and an NSF IPA grant. Opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health and National Science Foundation.

REFERENCES


