Improving the learning of chemical-protein interactions from literature using transfer learning and word embeddings.

P. Corbett, J. Boyle
Data Science Group, Technology Dept., The Royal Society of Chemistry

***Abstract*—In this paper we explore the application of artificial neural network ("deep learning") methods to the problem of detecting chemical-protein interactions in PubMed abstracts. We present here a system using multiple Long Short Term Memory layers to analyse candidate interactions, to determine whether there is a relation, and which type. A particular feature of our system is the use of unlabeled data, both to pre-train word embeddings, and also pre-train LSTM layers in the neural network. On the BioCreative VI CHEMPROT test corpus, our system achieves an F score of 61.51% (56.10% precision, 67.84% recall).**

Keywords—chemicals, proteins, text mining, relationship extraction, deep learning, LSTM.

I. INTRODUCTION

The BioCreative VI CHEMPROT task concerns the detection of mentions of interactions between chemical compounds/drugs and genes/proteins. Whereas there has been much work in the detection of chemical and gene/protein named entities, and in the recognition of some interactions involving these entities (1), there has been little work (2,3) so far on chemical-protein interactions.

The CHEMPROT corpus consists of PubMed abstracts manually annotated with chemical compound mentions, gene/protein mentions, and chemical compound-protein relations. Each relation annotation has one chemical compound mention, one gene/protein mention, and a relationship type. There are 22 relationship types, collected into ten groups, of which five groups are used in the CHEMPROT task (relations from the other five groups are discarded). The five relation groups are upregulator/activator (CPR:3), downregulator/inhibitor (CPR:4), agonist (CPR:5), antagonist (CPR:6), and substrate/product (CPR:9). The annotated abstracts are provided in three groups – 1020 training abstracts, 612 development abstracts and 3399 test abstracts.

We decided to examine the use of the neural network techniques known as "deep learning" (4) – in particular, using the recurrent neural network components known as Long Short Term Memory (LSTM) layers. One advantage of these deep learning methods is that they provide methods for exploiting unlabelled data. One method that we explored was the use of word embeddings – *n*-dimension vector representations of words – pretrained on large, relevant corpora. A second method was to exploit transfer learning – to build a neural network that can be trained on some task with just the use of an unlabelled corpus, train it, and then re-use some of the trained components from that network in the task of interest. One possible training task is to predict the next (or previous) word in a sentence or paragraph given all of the preceding (or following) words.

II. METHODS

A. RESOURCES

We used various external components in our system. The software components include the deep learning toolkit keras (5) - using tensorflow as the back end, python 3.6.1, and the tokeniser chemtok, as implemented in the chemical named entity recognition system ChemListem (6).

To prepare pre-trained word embeddings, we used the Stanford GloVe software (as checked out from version control July 3 2017). GloVe offers both some public pre-trained embeddings (based on Wikipedia and Common Crawl), and also the software to compile your own – in previous work (6) we had success with the public embeddings, but here, we compiled our own. To create these embeddings we prepared three corpora – the full texts of patents, consisting of patents with CPC codes A61K31 or A61P, from 2006 to November 2016, the full text of chemistry journal papers, consisting of papers published by the Royal Society of Chemistry from 2000 to end of 2016, and the titles and abstracts from PubMed records from 1809 to the end of 2015.

To make the initial embeddings, we extracted the text from the documents in the three corpora, tokenised it, outputting whitespace-separated tokens as one large text file. The contents of one document was separated from the next using lines consisting of "$GLOVEDUMMY " repeated 16 times. We used the GloVe software to extract 300-dimensional vectors from the file, using the window size = 15, xmax = 100.

We also prepared a file for transfer learning, taking the titles and abstracts from PubMed as mentioned above. The file consisted of on paragraph (usually a title or abstract) per line, in a random order. The file had approximately 24 million lines.

B. NEURAL NETWORK

The neural network system consisted of two neural networks – the "pretraining" network and the "recognition" network - with some components shared by both networks, and

other components being used by only one network or the other. The training procedure consisted of a series of epochs, the first five of which were divided into two phases – one (phase 1) to train the "pretraining" network, one (phase 2) to train the "recognition" network. All subsequent epochs after the fifth omitted phase 1 and ran phase 2 only. At the end of each epoch, the system was evaluated using the development abstracts, and an answer file was produced using the test abstracts. The epoch that gave the best F score in the evaluation phase – in the run submitted, the 33rd epoch – was selected, and the answer file from that was submitted for official evaluation.

Each run of phase 1 was divided into 25 sub epochs. In each sub-epoch, 12000 lines of the PubMed file were read in, tokenised, and sorted into batches of 32 lines each, grouping the smallest 32 lines (by number of tokens) into one batch, the next smallest 32 into another batch, etc. Within each batch, lines that are shorter (in terms of number of tokens) than the maximum length were padded with special padding tokens. The system was trained on the batches in a random order.

For each line, a token sequence was generated, consisting of an integer representing the index of each token in a token dictionary, with a special value for unknown tokens. From this a "substituted" sequence – where each token has a 0.5 chance of being replaced by a token randomly sampled from the lines read in that sub-epoch – was generated.

The inputs to the "pretraining" network consisted of the token sequence (input **i1**), the "substituted" sequence shifted one token to the right (input **i2**) (starting with padding), and the substituted sequence shifted one token to the left (ending with padding) (input **i3**). There were two outputs (**d2** and **d4**), one for each of the substituted shifted sequences, consisting of a sequence of numbers – 1 if the token in the substituted sequence is from the original sequence, 0 if it was randomly selected.

The network consisted of various layers, as shown in Table I. In all cases the number of output neurons is per token. The three embedding layers all shared the same embedding tensor. All LSTM layers were trained with a dropout and recurrent_dropout parameter of 0.5, and with return_sequences set to True.

TABLE I. LAYERS IN PRETRAINING NETWORK

Layer	Type	Input(s)	Number of output neurons	Notes
e1	Embedding	**i1**	300	
e2	Embedding	**i2**	300	
e3	Embedding	**i3**	300	
l1	LSTM	**e1**	300	
l2	LSTM	**e1**	300	Reversed
c1	concatenate	**l1**, **e2**	600	
c2	concatenate	**l2**, **e3**	600	
d1	TimeDistributed Dense	**c1**	300	activation is relu
d2	TimeDistributed Dense	**d1**	1	activation is sigmoid.
d3	TimeDistributed Dense	**c2**	300	activation is relu
d4	TimeDistributed Dense	**d3**	1	activation is sigmoid.

The "pretraining" network was trained using RMSProp optimizer, with the binary cross-entropy loss function.

In the second phase, chemical-protein relations were detected and classified. Each epoch consisted of a single pass through the training corpus to train the network, a single pass through the development corpus to evaluate the current state of the system, and a single pass through the test corpus to generate an answer file for submission.

In each pass, for each abstract, all possible chemical-protein pairs were found. Those pairs where the first token of the first entity was 60 or fewer tokens from the last token of the last entity were selected. A subsequence of tokens from the abstract was then taken, starting from 5 tokens before the first entity to 5 tokens after the last entity. The tokens for the chemical entity were replaced with "$CHEMICAL" and those for the protein entity were replaced with "$PROTEIN" – those appearing in both entities were replaces with "$BOTH". The token sequence was then converted to an integer sequence, in the same manner as the pretraining sequences were processed. Additional input sequences for each pair were also generated, consisting of an array of binary features for each token in the subsequence. One input sequence (input **i4**) consists of information about the entities in the abstract, regardless of whether they participated in the relation in question – these were features to say whether the token is in, at the start of, at the end of, overlapping the start of or overlapping the end of any chemical or protein entity. Another input sequence (input **i5**) consists of binary features to say whether the token is a part of the protein entity in question, and whether the token is a part of the chemical entity in question.

The output for the network (**d5**) was an array of 6 binary features, encoding whether and which relation exists between the two entities.

The network consisted of various layers, as shown in Table II. The number of output layers is per token, except for layers **p1** and **d5**, where it is the total number overall. The layers **e1**, **l1** and **l2** are shared with the pretraining network. Again, all LSTM layers were trained with a dropout and recurrent_dropout parameter of 0.5, and with return_sequences set to True.

TABLE II. LAYERS IN RECOGNITION NETWORK

Layer	Type	Input(s)	Number of output neurons	Notes
e1	Embedding	**i1**	300	
l1	LSTM	**e1**	300	
l2	LSTM	**e1**	300	Reversed

v1	Conv1D	**i3**	48	width=3, activation is relu
v2	Conv1D	**i4**	6	width=3, activation is relu
c3	concatenate	**l1**, **l2**, **v1**, **v2**	652	
l3	Bidirectional LSTM	**c3**	128 per direction, total 256	
p1	GlobalMaxPooling1D	**l3**	256	
d5	Dense	**p1**	6	activation is softmax

The network was trained using RMSProp, with the mean squared error loss function. During training, the candidate relationships were grouped into batches by length, if necessary padding the sequences to make the length of all the sequences in a batch uniform. The batches were then used for training in a random order.

C. ADDITIONAL EXPERIMENTS

To assess the importance of the techniques that used the unlabelled data, two additional runs were performed. One run ("No Phase 1") omitted the phase 1 from training. The second ("Random") also omitted phase 1, and also used randomly-initialised embeddings rather than the GloVe-trained ones. On the No Phase 1 run, the best epoch was the 15th epoch, and on the Random run, the best epoch was the 17th epoch.

III. RESULTS AND DISCUSSION

Table III shows the results from the task:

TABLE III. RESULTS

Corpus	*Precision*	*Recall*	*F*
Development	56.52%	70.42%	62.71%
Test	56.10%	67.84%	61.41%

The F of less than 63% indicates that there is considerable room for improvement on this task. This is the first time that BioCreative has tackled a chemical-protein interaction task – however, in the past it has considered chemical-disease relations (getting a maximum F score of 57.03%) and protein-protein interactions (getting a maximum F of 55%). These relationship-mining tasks appear to be harder than named entity extraction tasks, where F scores in excess of 80% are routine and F scores above 90% are not unknown (10). There appears to have been a slight loss of performance between the development and test – it is possible that this is because the gains from selecting the best epoch did not generalize well.

Table IV shows a confusion matrix for the development data.

TABLE IV. CONFUSION MATRIX FOR DEVELOPMENT DATA

Actual	Predicted					
	NONE	***CPR:3***	***CPR:4***	***CPR:5***	***CPR:6***	***CPR:9***
NONE	26196	214	413	75	76	351
CPR:3	163	287	82	4	4	9
CPR:4	159	24	896	0	5	6
CPR:5	23	0	0	89	4	0
CPR:6	28	1	7	3	160	0
CPR:9	166	4	16	0	2	258

The major source of error seems to be non-relations being mistaken for relations and vice versa. There is something of a problem with upregulation (CPR:4) being mistaken for downregulation (CPR:3) but this is not the biggest problem for either relationship class.

TABLE V. DEVELOPMENT DATA RESULTS BY RELATIONSHIP CLASS

Class	Precision	Recall	F
CPR:3	54.16%	52.28%	53.20%
CPR:4	63.37%	82.20%	71.57%
CPR:5	52.04%	76.72%	62.02%
CPR:6	63.75%	80.40%	71.11%
CPR:9	41.35%	57.85%	48.22%

There is considerable variation in how well these entities are recognised – CPR:4 (downregulator/inhibitor) and CPR:6 (antagonist) are well-recognised, CPR:3 (upregulator) and CPR:9 (substrate/product) are poorly recognised. The F scores do not appear to be correlated with the number of mentions in the corpus.

TABLE VI. RESULTS ON DEVELOPMENT

Run	*Precision*	*Recall*	*F*
Full	56.52%	70.42%	62.71%
No Phase 1	62.97%	57.25%	59.97%
Random	45.05%	50.66%	47.70%

Table VI shows the results of re-running the system, progressively disabling parts of the system that make use of unlabelled data. The Phase 1 training of the lower LSTMs is shown to improve performance by 2.7 percentage points; the GloVe-trained embeddings are worth 12.2 percentage points.

IV. CONCLUSION

Methods based on "deep learning" recurrent neural networks can be used to detect relationships between chemicals and protein, with results comparable with those observed in other biomedical relationship extraction tasks. The deep learning structure allows the use of large amounts of unlabelled

text to boost performance, especially via the use of pre-trained word embeddings.

The source code for our system is available on-line, as a part of the distribution for ChemListem, at https://bitbucket.org/rscapplications/chemlistem.

ACKNOWLEDGMENT

We would like to thank Colin Batchelor, Aileen Day, Nicholas Bailey and Jeff White for valuable discussions.

REFERENCES

1. Krallinger,M., Rabal,O., Lourenço,A., Oyarzabal,J. and Valencia,A. (2017) Information Retrieval and Text Mining Technologies for Chemistry. *Chem. Rev.*, **117**, 7673-7761.

2. Craven,M. and Kumlien,J.. (1999) Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, **1999**, 77-86.

3. Rindflesch, T. C., Tanabe, L., Weinstein, J. N., and Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing,* 517.

4. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep Learning. *Nature* **521** 436-444.

5. Chollet, F. (2015) "Keras" https://github.com/fchollet/keras.

6. Corbett, P, Boyle, J. Chemlistem - chemical named entity recognition using recurrent neural networks. In *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 61-68.

7. Pennington,J., Socher,R. and Manning,C. (2014) GloVe: Global Vectors for Word Representation.

8. Wei,C.-H., Peng,Y., Leaman,R., Davis,A.P., Mattingly,C.J., Li,J., Wiegers, T.C. and Lu,Z. (2015) Overview of the BioCreative V Chemical Disease Relation (CDR) Task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, 154-166.

9. Kralliner, M., Vazquez, M., Leitner, F. And Valencia, A. (2011) Results of the BioCreative III Interaction Method Task. In *Proceedings of BioCreative III Workshop*, 5-12.

10. Pérez Pérez,M., Rabal,O., Pérez Rodríguez,G., Vazquez,M., Fdez-Riverola,F., Oyarzabal,J., Valencia,A., Lourenco,A. and Krallinger,M. (2017) Evaluation of chemical and gene/protein entity recognition systems at BioCreative V.5: the CEMP and GPRO patents tracks., in *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 3-11.