Combining Support Vector Machines and LSTM Networks for Chemical-Protein Relation Extraction

Farrokh Mehryary[1,2], Jari Björne[1,3], Tapio Salakoski[1,3], Filip Ginter[1]
1. Department of Future Technologies, University of Turku, Turku, Finland;
2. University of Turku Graduate School (UTUGS), University of Turku, Turku, Finland;
3. Turku Centre for Computer Science (TUCS), Turku, Finland

***Abstract*—We present the results of our participation in the BioCreative VI: Text mining chemical-protein interactions (CHEMPROT) track. The goal of this task is to promote the development and evaluation of systems capable of extracting relations between chemical compounds/drug and genes/proteins from biomedical literature. We participate with two systems: (1) an SVM system which relies on a rich set of features extracted from the parse graph and (2) an ensemble of neural networks that utilize LSTM networks and generate features along the shortest path of dependencies. We also combine the predictions from the two systems with the goal of increasing performance. On the development set, our system combination approach outperforms the two individual systems, achieving an F-score of 61.09 (according to the official evaluation metric). On the test set, our SVM system achieves the highest result for our submissions with an F-score of 60.99.**

Keywords — SVM; deep learning; ensemble learning; long short-term memory networks; LSTM; biomedical relation extraction;

I. Introduction

BioCreative VI Task 5 focuses on detection of statements of relations between chemical compounds/drugs and genes/proteins. The CHEMPROT corpus which provides such annotations is used as the training and test data in this task. The aim of the task is to promote the development of systems for extracting such relations for use in precision medicine, drug discovery and basic biomedical research[1].

BioCreative VI Task 5 follows the well-established approach of pairwise relation extraction in the field of biomedical text mining. Protein-protein interactions (PPI) were one of the extraction targets in the BioCreative II and BioCreative III challenges (1,2). The two Drug-Drug Interaction (DDI) shared tasks focused on the detection of adverse interactions between pairs of drugs (3,4). Considerable performance gains achieved using deep learning have recently been reported on the DDI Extraction 2013 corpus (5).

We approach the BioCreative VI Task 5 as a classification task where we classify each valid pair of entities as one of the annotated relation types or as a negative. We apply and compare two systems in this task, one based on artificial neural networks (ANN) and one on support vector machines (SVM). After optimizing these systems separately, we experiment with system combination, achieving increased performance on the development set. On the test set we note a considerable drop in the ANN performance, which requires further investigation.

[1] http://www.biocreative.org/tasks/biocreative-vi/track-5/

II. Data

The CHEMPROT corpus is a pairwise relation dataset. All entities are given as known data for the participants, thus the task is to predict the relations for valid pairs of these entities. The relations are directed, always connecting a GENE type entity (gene or protein) to a CHEMICAL type entity. A large set of distinct types are used for annotating the relations, but these types are combined into 10 groups which are used as the actual classes for this task. Further, only five of these classes are taken into account in the task evaluation.

III. Methods

We develop two different systems capable of extracting relations between CHEMICAL and GENE entities. Our first system relies on a rich set of features and a linear Support Vector Machine (SVM) classifier. Features for this system are generated from the shortest dependency path connecting the two candidate entities in the sentence syntax dependency graph, from the linear order of tokens, a sentence bag of words and all dependency paths within 1–3 dependencies from the two entities. Our second system requires less feature engineering and is a deep learning-based system, utilizing an ensemble of three-channel long short-term memory networks. Features for this system are generated based on three information channels: words, part-of-speech (POS) tags and dependency type and word-adjacency edges, along the shortest path connecting the two entities. Finally, we combine predictions of the two systems to boost the F-score, using a simple algorithm that is optimized on the official development set. In this section we discuss the details of each approach.

A. Preprocessing

We use the TEES system (6) to run a preprocessing pipeline of tokenization, part-of-speech tagging, and parsing. We convert the CHEMPROT corpus into the Interaction XML

format allowing it to be parsed by the TEES preprocessing system. We test different parses generated using the TEES preprocessor wrappers for the BLLIP, Stanford converter and SyntaxNet parser software (7-9). The default parsing pipeline in our experiments consists of BLLIP constituency parsing with the biomedical domain model of McClosky (10), followed by conversion to dependencies using the Stanford conversion tool (8). We test different variants of the Stanford Dependencies (SD) representation, with the *CCprocessed* variant being the default unless otherwise stated.

Since our systems work mainly based on the shortest dependency path connecting two candidate entities in a single sentence, we exclude any possible cross-sentence candidate pairs from the data. The training data incorporates ten different types of relations, 5 of them being evaluated in the task. We also define and add a ''negative'' type for the cases where no relation exists between the two candidate entities. Hence, we formulate this relation extraction task as an 11-class classification problem.

B. SVM-based system

The SVM-based system used in this work is the Turku Event Extraction System (6). The system is applied as-is, with no task-specific modifications. The TEES system uses the SVM^{multiclass} software as the multiclass classifier implementation (11).

The TEES system relies on a rich feature representation. While most features are generated from the shortest path of dependencies, dependency chains outside this shortest path, bags of words and the linear order of tokens are also used in generating features, in an attempt to capture more of the sentence context outside the direct relation between the two entities of interest.

We test several different parses and ways of predicting the CHEMPROT corpus with TEES, but find that none of these improve performance over the default approach. In total we compare three ways of representing the corpus, 12 parses and the use of the DrugBank dataset (12) as additional features. For the three CHEMPROT corpus representations the TEES system is trained with either the default of all 10 classes, with the four non-evaluated classes merged into a single class or with the non-evaluated classes entirely removed. For parses, we try the BLLIP parser, with or without the McClosky biomodel and with all five types of Stanford conversion, as well as the SyntaxNet parser with or without its Universal Dependencies model.

C. Deep learning-based system

Our deep learning-based system (ANN for short) requires less feature engineering than the SVM-based system and generates the features along the shortest path that connects the two candidate entities in the syntactic parse graph. The shortest dependency path is known to contain most of the relevant words for expressing the relation between the two entities while excluding less relevant and uninformative words (13), hence many successful systems have been built around utilizing it (6,13-17).

The architecture of our deep learning-based system is centered around utilizing an ensemble of artificial neural networks, all having identical structure, but trained with different initial random weights. This is done to stabilize the variance in the measured performance, caused by the random initialization of the network weights.

Each neural network in the ensemble utilizes three separate long short-term memory networks (chain of LSTM units): for representing the sequence of words, the sequence of POS tags and the sequence of dependency types (DT, i.e., edges in the parse-graph) along the shortest path. We always traverse the path from the CHEMICAL entity to the GENE entity when generating features along the shortest path, regardless of the order of the entity mentions in the sentence. We notice this approach results in significantly better generalization for unseen data. Besides the existing dependency type edges in the parse graph, we add an artificial edge between any two adjacent words of the sentence (word-adjacency edges). As discussed by Quirk et. al (18), this approach mitigates the parsing errors and increases accuracy and robustness when the system is confronted with linguistic variation. We give the weight one to dependency type edges and the weight five to word-adjacency edges when searching for the shortest path in the graph.

The sequences of words/POS tags/dependency types are first mapped into sequences of their corresponding vector representations, i.e. embeddings, by three separate embedding lookup layers and then used as input for the LSTMs. For words, we use pre-trained word-embeddings provided by Pyysalo et al. (19), which have been trained on the texts of all PubMed titles and abstracts and PubMed Central Open Access (PMC OA) full text articles using the word2vec method (20). During the training of our system, word embeddings are fine-tuned while randomly initialized POS and dependency type embeddings are learnt from scratch. The outputs of the last LSTM unit of each of the three chains are concatenated and the resulting vector is fed to a fully connected hidden layer. The hidden layer finally connects to the decision layer, having an output dimensionality corresponding to the number of labels in the data set (plus one for the ''negative'' label) with softmax activation.

The network is trained on the official training data using the Nadam optimization algorithm. Applying a dropout (21) with the rate of 0.2 on the output of the first dense layer is the only explicit regularization method used. The training is stopped once the performance on the development set is no longer improving, measured using the official evaluation metric. Table I shows the comprehensive list of the hyperparameters used.

TABLE I. HYPERPARAMETERS OF THE NETWORKS

Hyperparameters	Values	
	Optimal value	*Tested values*
Word-adjacency edge weight	5	[3,4,5,6]
Word embedding dimensionality	200	pre-trained
POS embedding dimensionality	25	[25,50,75,100]
DT embedding dimensionality	25	[25,50,75,100]
Word LSTM, output dimensionality	300	[100,200,300,400]
POS tags LSTM, output dimensionality	200	[100,200,300,400]
DT LSTM, output dimensionality	200	[100,200,300,400]
Hidden layer, output dimensionality	200	[100,200,300,400, 500,600,700,800]
Activation functions	tanh	[tanh, sigmoid]
Dropout rate	0.2	[0 0.2 0.3 0.4 0.5]

[a] The optimal and tested values for hyperparameters

To deal with the variance in the performance, we train an ensemble of 4 neural networks, all identical apart from the initial (random) weights. After training, each network predicts a set of confidences for each (development/test set) example. The final prediction for an example is generated by summing the confidences of all networks and choosing the label with the highest overall confidence.

D. System Combination

Our SVM and deep learning-based systems are trained with different sets of features. This is a potential case for investigating whether combining predictions of the two systems could help in achieving better performance for this task.

The system combination is implemented by merging the relation predictions from the two systems as either a union (OR) or an intersection (AND), and resolving overlapping predictions with conflicting types by using the classifier confidence scores. Since all entities are known data in this task, the predictions from the two systems can be aligned using pairs of gold standard entities.

If only one system predicts a relation for a given pair of entities, it is either included in (OR) or discarded from (AND) the combination. If both systems predict a relation, the relation with the higher confidence score is included in the combination. Both the SVM and ANN systems produce confidence scores in their own ranges. These ranges are normalized into the 0–1 interval for both systems, after which the normalized scores are compared. We experiment with combining all predictions, only positive predictions or only predictions for the evaluated classes and find that combining only positive predictions results in the best performance.

IV. RESULTS AND DISCUSSION

We conduct all of our experiments on the official development set using the official evaluation script provided by the organizers. Even though the data are annotated having ten different types of relations, the task only focuses on five of them by defining the official performance metric as the micro-averaged F-score of the five target classes. This is most likely due to the fact that there are much less training examples available in the data for the ignored classes. Table II shows the performance comparison of our different systems, evaluated on the development data.

TABLE II. PERFORMANCE OF THE SYSTEMS ON THE DEVELOPMENT SET

Evaluation on development set	Performance metrics		
	Precision	*Recall*	*F-Score*
SVM	64.55	54.72	59.23
ANN	61.90	55.01	58.25
SVM+ANN (OR, positive classes)	**58.45**	**63.99**	**61.09**
SVM+ANN (AND, positive classes)	75.42	48.14	58.77
SVM+ANN (OR, all classes)	65.82	55.55	60.25
SVM+ANN (AND, all classes)	65.82	55.55	60.25
SVM+ANN (OR, eval classes)	56.47	65.07	60.46
SVM+ANN (AND, eval classes)	79.28	45.78	58.04

As Table II shows, both the SVM and deep learning-based (ANN) systems have very similar performance on the task, with the SVM having an F-score 1pp above the ANN. This might be due to the fact that the ANN solely relies on the words and edges seen on the shortest path and we suspect that in many cases, the *trigger word* (i.e., a token or sequence of tokens which expresses the actual relation between the two candidate entities) might be absent from this path. Consequently, the ANN might not get the chance to see this information, whereas the SVM system generates features based on all tokens and dependencies near the two entities, as well as those on the shortest path connecting them. The best SVM performance is achieved with the TEES default settings, without using the DrugBank features, using the BLLIP+biomodel+CCProcessed parsing approach and including all ten CHEMPROT relation types in the training data.

For both systems, recall is considerably lower than precision (for instance, recall is 10pp below precision for the SVM). Using the OR operation in system combination considerably improves the recall (~9pp) while causing a comparatively lower drop in precision, leading to an approximately 1–1.5pp increase in the resulting F-score. We observe that discarding negative predictions and building the combination from all 10 positive classes results in the highest performance on the development set.

For predicting the test set, we combine the training and development data when training the SVM system. This is a quite common approach when using classifiers such as SVMs. However, training the neural networks on the combined data for the *optimal* number of epochs (found during the optimization) might lead to under/over-fitting, because more/less training epochs might be needed. Finding the optimal number of epochs for training the network on the combined data is challenging. In this task, participating teams were allowed to submit up to 5 different test set predictions. Hence, we submitted two sets of ANN predictions: (1) predictions of the ensemble of networks that are trained for 3 epochs (the optimal number found in optimization), (2) predictions of the ensemble when the networks are trained for 4 epochs. We also combined these two sets of predictions with the SVM system predictions (using our system combination approach), resulting in a total of five sets of test set predictions. Table III shows the official results for our submissions on the test set, as calculated by the task organizers.

TABLE III. PERFORMANCE OF THE SYSTEMS ON THE TEST SET

Evaluation on test set	**Performance metrics**		
	Precision	***Recall***	***F-Score***
SVM	66.08	56.62	60.99
ANN (3 epochs)	63.73	44.62	52.49
ANN (4 epochs)	63.37	43.87	51.85
SVM+ANN (3-epochs)	61.05	60.06	60.55
SVM+ANN (4-epochs)	60.88	59.89	60.38

As Table III shows, compared to the development set results, our SVM system has approximately the same level of performance on the test set, achieving an F-score of 60.99, with a similar imbalance between precision (66.88) and recall (56.62). However, for the ANN submissions we notice a significant drop in recall (~11pp) with a small increase in precision (~1pp), leading to an F-score of 52.49 (in the case the networks are trained for 3 epochs) or 51.85 (when the networks are trained for 4 epochs), which is about 6pp below the F-score seen on the development set. As a direct result, none of the two system combination approaches have been able to produce a result better than the SVM system alone.

Since at the time of manuscript submission test set labels have not yet been published, we cannot perform a comprehensive analysis on the results of the ANN system. One potential explanation could be related to (over) training the networks with the combined data, which together with the small mini-batch size may have led to some overfitting. Particularly, we notice a considerable drop in the training-loss between the second and third epochs, which is approximately double the difference of the loss between the third and fourth epochs. This might indicate that overfitting would have occurred when training the networks for the third epoch.

Since the organizers are going to publish the test set labels, as our first additional experiment we would like to investigate what would be the performance of the ANN system on the test set using the version of the networks trained solely on the training set and optimized on the development set.

V. CONCLUSIONS AND FUTURE WORK

We participated in the CHEMPROT track of the BioCreative VI shared task with two different systems. Our SVM system relies on a rich set of features, extracted from the sentence parse graph, whereas our deep learning-based system requires less feature engineering and is an ensemble of three-channel LSTM networks. Features for this system are generated based on the shortest path which connects the two candidate entities.

Experiments on the development set show that combining the predictions of the two systems can lead to overall performance higher than that of either of the two systems alone. While the SVM system performs equally well on the development and test sets, the ANN system performance is considerably lower on the test set. We aim to investigate the possible reasons behind this result as soon as the test set labels are published.

The SVM system uses many features in addition to those extracted from the shortest path of dependencies. As future work, we would like to study the effect of each of these feature types on the performance for this task, as well as investigate efficient ways to incorporate and utilize such features with our deep learning -based approach.

ACKNOWLEDGMENTS

This work was supported by an ATT Tieto käyttöön grant. Computational resources were provided by CSC - IT Center for Science Ltd., Espoo, Finland.

REFERENCES

1. Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A., et al. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology, 9(Suppl 2):S4.

2. Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., et al. (2011). The Protein-Protein Interaction tasks of BioCreative III: classi cation/ranking of articles and linking bio-ontology concepts to full text. BMC Bioinformatics, 12(Suppl 8):S3.

3. Segura-Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the

Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

4. Segura-Bedmar, I., Martínez, P., and Sánchez-Cisneros, D. (2011). The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. In Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011: 7 Sep 2011; Huelva, Spain, pages 1–9.

5. Liu, S., Chen, K., Chen,Q., and Tang, B. (2016) Dependency-based convolutional neural network for drug-drug interaction extraction. In Proceedings of IEEE-BIBM, pp.1074-1080.

6. Björne, J. (2014) Biomedical Event Extraction with Machine Learning. Ph.D. thesis, University of Turku.

7. Charniak, E., and Johnson, M. (2005) Coarse-to-fine N-best parsing and maxent discriminative reranking. In proceedings of ACL, pp.173:180.

8. de Marneffe, M.-C., MacCartney, B. and Manning, C. D. (2006) Generating Typed Dependency Parses from Phrase Structure Parses. In proceedings of the LREC-2006, pp.449:454.

9. Andor D., Alberti C., Weiss D., Severyn A., Presta A., Ganchev K., Petrov S., and Collins M. (2016) Globally Normalized Transition-Based Neural Networks. CoRR abs/1603.06042. http://arxiv.org/abs/1603.06042.

10. McClosky, D. (2010) Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Ph.D. thesis.

11. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. (2005) Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research (JMLR) 6(Sep), pp.1453–1484.

12. Knox C., Law V., Jewison T., et al. (2011) Drugbank 3.0: a comprehensive resource for omics research on drugs. Nucleic Acids Research, 39(Database-Issue):1035–1041.

13. Mehryary, F., Björne, J., Pyysalo, S., Salakoski, T. and Ginter, F. (2016) Deep Learning with Minimal Training Data: TurkuNLP Entry in the BioNLP Shared Task 2016. In proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, pp.71-81.

14. Mehryary, F., Hakala, K., Kaewphan, S., Björne, J., Salakoski, T. and Ginter, F. (2017) End-to-End System for Bacteria Habitat Extraction. In proceedings of BioNLP 2017, pp.80-90.

15. Cai, R., Wang, H., and Zhang, X. (2016) Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In proceedings of ACL, pp.756-765.

16. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H. & Jin, Z. (2015) Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In Proceedings of EMNLP, pp.1785-1794.

17. Bunescu, R. C.and and Mooney, R. J. (2005) A shortest path dependency kernel for relation extraction. In proceedings of HLT-EMNLP, pp.724–731.

18. Quirk, C., and Poon, H. (2017) Distant Supervision for Relation Extraction beyond the Sentence Boundary. In Proceedings of EACL, pp.1171-1182.

19. Pyysalo, S., Ginter, F., Moen, H., Salakoski, T. and Ananiadou, S. (2013) Distributional Semantics Resources for Biomedical Text Processing. In proceedings of LBM 2013. pp. 39-44.

20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality. In proceedings of NIPS, pp.3111-3119.

21. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *Machine Learning Research.*, **15**, 1929-1958.