Drug and Chemical Compound Named Entity Recognition using Convolutional Networks

Mark J. Berger
Department of Computer Science
Stanford University
Stanford, CA 94305
mjberger@stanford.edu

1 Introduction

As biomedical literature continues to grow at an explosive rate, researchers are unable to process the vast amounts of information generated by one another. In order to account for this, text mining and information extraction systems have been developed in order to help researchers find information that is relevant to their respective research. However, text mining systems have also been developed to infer new knowledge, and further biological understanding. Examples include inferring protein-protein interactions, and gene-disease interactions, from publicly available literature in order to further the understanding of systems biology, and ultimately, to better combat disease (Krallinger et al. 2010).

In order for all of these systems to produce strong results, each system must be able to accurately determine the name of a gene, protein, or any other item of interest. Therefore, strong named entity recognition (NER) systems must be developed in order to prevent errors from propagating through the system, and negatively affecting performance. In order to strengthen the ability for text mining systems to infer novel biological knowledge, we focus on the task of improving named entity recognition for drugs and chemical compounds.

Specifically, in this work, we explore NER systems which do not incorporate linguistic or domain-dependent information, instead opting to explore how text representation affects performance. For all of the models discussed in this paper, the input to our algorithm is the abstract of a biomedical paper. We then use a convolutional neural network to predict whether a given token is part of a drug or chemical compound. Given this restriction, we show that a convolutional neural network outperforms a strong baseline. However, future work is necessary in order to rival the performance of existing systems.

2 Related Work

With the induction of the chemical mention recognition task to BioCreative IV (Krallinger et al. 2015), multiple systems have addressed this task (Leaman, Lu, & Munkhdalai). Despite a variety of systems, the top performing systems all use Conditional Random Fields, which is a form of undirected graphical model. The model derives its power from being able to account for the probability of neighboring samples, therefore making them a natural fit for named entity recognition tasks (McCallum et al.).

For input and processing, the models differ, but all of them follow a general trend. Lu et al. focus on representing the documents in high dimensional spaces by generating vector representations with Brown clustering (Brown et al.) and the Skip-gram model described in word2vec (Mikolov et al.). They supply both of these representations to their CRF model to determine the entities. In contrast, Leaman et al. opt to use an n-gram representation with linguistically informed features, such as lemmatization and part-of-speech tags. They also incorporate a variety of domain-specific knowl-

edge, such as amino acids and common chemical elements. Finally, Munkhdalai et al. combine both approaches by using Brown clustering and word2vec to generate high dimensional representations, while also supplying domain specific features to the CRF model. Despite their differences, all of these models produce comparable results when measuring performance with micro F1.

However, when we examine these CRF model closely, we notice an underlying limitation: CRFs are linear models, and distributed representations have relationships which cannot be linearly modeled in a low dimensional space (Wang & Manning). Therefore, we explore whether a deep architectures, which address these two limitations, can produce superior performance.

3 Data Set and Preprocessing

3.1 Data Set

For our data set, we use the CHEMDNER corpus of chemicals and drugs from the BioCreative IV chemical mention recognition task (Krallinger et al. 2015). Overall, the data set contains a collection of 10,000 randomly selected PubMed abstracts, which contain a total of 84,355 chemical entity mentions. The annotations were hand labeled by experts in the field, and therefore are considered to be reliable. While the data set has two sub-tasks, we specifically focus on the chemical document indexing (CDI) task, which requires the system to return all unique chemical entities within a given abstract.

> Polycyclic aromatic hydrocarbons (PCAHs), particularly those with a high molecular weight, have been classified as probable carcinogens to humans... Different samples will be collected and analyzed for five PCAHs including pyrene, benzo(a)anthracene, benzo(e)pyrene, benzoflouroanthene, and benzo(a)pyrene.

Figure 1: An excerpt of a biomedical abstract from the CHEMDNER data set, where the first instance of each chemical compound is highlighted.

Officially, the data set has the following split: 3500 articles for training, 3500 articles for development evaluation, and 3000 articles for the held out test set. Since neural network models are known to require a lot of training data, we rearranged the data set by randomly selecting 1000 articles from the development set to become our new development set. We then used the remaining 2500 articles as training data. The test set was not modified. Since the number of named entities is rather small compared to the overall number of tokens in the set, we also duplicate all named entity data tokens in order to combat class imbalance.

3.2 Preprocessing

Before training any models, we tokenize the data by splitting on the symbols in figure 2 using the following regular expression: ([SYMBOLS]). This symbol table was previously used by Lu et al. on the same data set. While this approach is considered to be rather trivial compared to using a statistical parser, for chemical compounds, this offers superior performance. Since statistical parsers are trained on more general text, they will often split domain specific words, such as N-(4-hydroxy-3-mercaptonaphthalen-1-yl)amides, in odd or unpredictable ways. This can cause the named entities to be combined with other words, which ruins our ability to accurately return the entities within the text. Therefore, using a naive regular expression is more suitable for the task at hand.

~ • ! @ # $ % ^ & * - = _ + ¯ ( ) [ ] { } ; ' : " , . / < > × > < ≤ ≥ ↑ ↓ ← →
• ′ ° ~ ≈ ? Δ ÷ ≠ | ‘ ’ “ ” §£ € \ 0 1 2 3 4 5 6 7 8 9

Figure 2: Symbols used to split the text (Lu et al. 2015).

For input to our neural network models, we opted to train word vectors for each token using the popular word2vec system with the default parameters and a vector length of 200 (Mikolov et al.). As training data, we used the 2014 version of the BioASQ data set, which contains roughly 12.5 million biomedical abstracts from the PubMed database (Tsatsaronis et al.). We removed any abstracts from the BioASQ data set which were also in our development and test sets.

4 Models

4.1 Baseline

As a baseline, we use logistic regression with L2 regularization and a bag-of-words character-based ngram representation. Each token in the abstract is classified, and consecutive positively classified tokens are joined to form an entire entity. These final entities are then evaluated to determine an F1 score.

4.2 Convolutional Neural Network

For our experiments we explore the use of a convolutional neural network (CNN) model (Krizhevsky et al., LeCun et al.), which has been used for NER in the past (Collobert 2008, 2011). We use a slight variation of the CNN architecture presented by Kim, and we briefly describe it here. In order to represent a token, we construct an n by k matrix, where n is the number of tokens and k is the dimension of the word vectors. For each row in the matrix, row i contains the word vector for the ith word in the document. Tokens which occur at the beginning and end of the document are zero padded to have a length of $1 + 2 \cdot w$ where w is the window size.

After constructing our matrix, we conduct a convolutional operation over each document's respective matrix. Specifically, we apply a weight matrix (known as a filter) $W \in \mathbb{R}^{hk}$ to a window of size h by k, where h is the length of an n-gram, to produce a scalar c_i. Specifically, this is calculated by:

$$c_i = f(W x_{i:i+h-1} + b^{(c)})$$

where b is the bias term and f is a chosen non-linearity. During convolution, we apply this filter over all possible $N - h + 1$ windows to produce the set of features $\{c_1, c_2, ..., c_{N-h+1}\}$. We then apply a max-over-time pooling operation (Collobert et al. 2011), to the set of $N - h + 1$ features to produce the feature $\hat{c}$.

$$\hat{c} = \max(c_1, c_2, ..., c_{N-h+1})$$

Therefore, we extract a single feature from the set of features produced by the filter. We repeat this same process for multiple feature maps of size h by k, as well as for filters with different sizes of h. We then concatenate all of these features into a single vector, creating the vector c', and supply c' to a set of fully connected layers. In order to prevent feature adaption, we apply the popular dropout method with a probability of p to the penultimate layer.

For the final layer, we add a fully connected output layer with a sigmoid activation function, which produces a scalar. We treat this scalar as the probability that the center token is contained within a named entity.

$$\hat{y} = \sigma(U c' + b^{(o)})$$

Intuitively, we believe this model has the ability to offer superior performance to CRFs because of its filter-based approach. The character composition of chemical compounds tend to differ from other English words, and therefore, we believe that the CNN model will learn filters which are activated by their unusual token compositions.

As in the baseline, each token in the abstract is classified, and consecutive positively classified tokens are joined to form an entire entity. We use a threshold .5 to determine whether a token is positively classified.

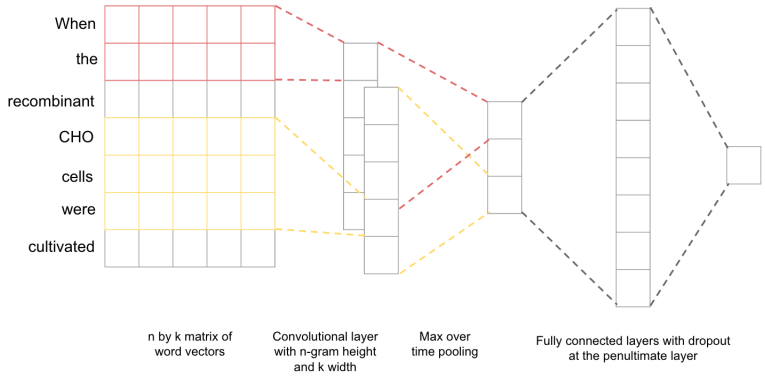

Figure 3: Our model inspired by Kim (2014).

5 Experiments & Results

5.1 Baseline

For our logistic regression model, we performed a grid search over the window sizes 0 through 5, and the ngram sizes 1 through 7. Our model performed best on the development set with a window size of 2 and an ngram size of 5 and was implemented using the popular Scikit-Learn library (Pedregosa et al.).

5.2 Convolutional Neural Network

For our CNN model, we evaluated window sizes of 1 and 2 with the following combinations of filters: 100 unigram, 100 bigram, 100 trigram filters, and 300 trigram filters. Our model performed best on 300 trigram filters. For all experiments, we fixed the fully connected layers to a single layer with a size of 600, a Leaky ReLU non-linearity with alpha of 0.1 (Mass et al.), a dropout probability of 0.5 (Hinton et al.), and constrained the column norm to 1.618. The convolutional weight matrices are initialized via drawing from a Gaussian distribution with mean 0 and standard deviation of .01. The fully connected layers were initialized using He initialization (He et al.), and the model was trained with stochastic gradient descent with momentum (Nesterov) and a batch size of 128. We performed two experiments, one with fixed word vectors and one with those that were updated. Our implementation was built using theano (Bergstra et al.)

6 Results and Analysis

In order to evaluate our models, and compare them to existing models, we use the popular micro precision, micro recall, and micro F1 score metrics. Overall, our baseline model is able to achieve an F1 score of .5395 on the test set. Our two CNN models performed nearly identically producing a F1 score of .6655 and .6660, respectively. However, these results do not rival the best performing systems in this task.

After analysing our results, we believe our model has a variety of problems. As seen in figure 4, the precision and recall of our models on the token level is significantly better than the entity level, with a F1 score of .8137. However, once these tokens are joined to form entities, the F1 significantly decreases to .6660. By evaluating these results and examining the system's output by hand, we noticed that our model will often drop a middle token in an entity, which will cause it to form two entities. In a similar vein, additional, incorrect, tokens will be placed at the beginning or end of a correct entity. We believe this is caused by the short nature of the tokens due to our aggressive tokenization technique, which in turn creates similar context windows for tokens which do, and do not, belong in an entity. Additionally, unlike the conditional random field models used in other systems, the CNN cannot condition on the likelihood of the surrounding tokens also being positive

Model	Precision	Recall	F1
Our Baseline	.6130	.4817	.5395
Our Model - static	.6174	.7218	.6655
Our Model - non-static	.6094	.7343	.6660
Lu et al.	.87018	.89408	.88197
tmChem (Leaman et al.)	.8781	.8724	.8752
BANNER-CHEMDNER (Munkhdalai et al.)	.8890	.8268	.8568

Table 1: Micro F1 results on the CHEMDNER test set. In the case where other papers contained multiple models, we chose the model with the highest F1 score.

instances. While it should be able to learn a similar output from the surrounding window, it is not able to do joint inference like in the CRF models.

Finally, we believe our model is suffering from high bias after evaluating the error produced on the training and evaluation data sets. Throughout the course of the entire experiment, the error on the training and evaluation sets stay roughly the same, and the model never begins to overfit the test set. Therefore, the model may not be expressive enough to capture the underlying function we are attempting to model.

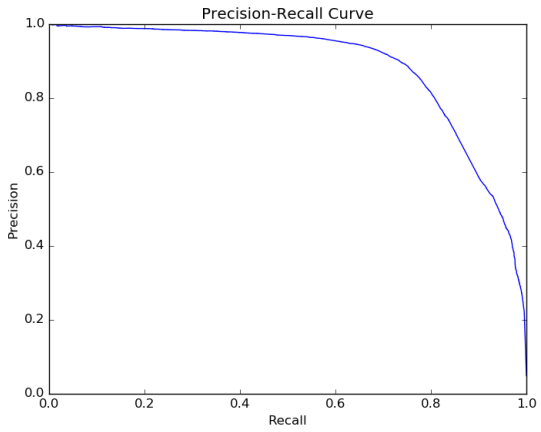

Figure 4: The Precision-Recall curve for our best model at the token level.

7 Future Work

In order to address these concerns, we wish to explore a variety of options. First, we wish to significantly increase the number of convolutional filters, and the size of the fully connected layers, in order to make our model more expressive. This will hopefully combat the high bias problem we are experiencing. Similarly, we can also add multiple convolutional layers, instead of using the aggressive pooling scheme in Kim. Additionally, we would like to experiment with using one-hot character representations, and using convolutional layers to extract features from that sparse representation (Santos & Zadrozny). Finally, we would like to experiment with other models which combine aspects of conditional random fields and convolutional networks, such as neural conditional random fields (Do & Artieres).

Acknowledgements

The author would like to acknowledge Mitch Sanford for his help in conceiving the project and parsing the CHEMDNER data set. The author would also like to acknowledge Professor Gill Bejerano for access to his lab's computing resources.

References

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., ... & Bengio, Y. (2010, June). Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for scientific computing conference (SciPy) (Vol. 4, p. 3).

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.

Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167). ACM. Chicago

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K.,& Kuksa, P. (2011). Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12, 2493-2537.

Do, T., & Arti, T. (2010). Neural conditional random fields. In International Conference on Artificial Intelligence and Statistics (pp. 177-184).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. arXiv preprint arXiv:1502.01852.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

Krallinger, M., Leitner, F., & Valencia, A. (2010). Analysis of biological processes and diseases using text mining approaches. In Bioinformatics Methods in Clinical Research (pp. 341-382). Humana Press.

Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., ... & Segura-Bedmar, I. (2015). The CHEMDNER corpus of chemicals and drugs and its annotation principles. Journal of cheminformatics, 7(Suppl 1), S2.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Leaman, R., Wei, C. H., & Lu, Z. (2015). tmChem: a high performance approach for chemical named entity recognition and normalization. Journal of cheminformatics, 7(supplement 1).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

Lu, Y., Ji, D., Yao, X., Wei, X., & Liang, X. (2015). CHEMDNER system with mixed conditional random fields and multi-scale word clustering. Journal of cheminformatics, 7(Suppl 1), S4.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML (Vol. 30).

McCallum, A., & Li, W. (2003, May). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (pp. 188-191). Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

Munkhdalai, T., Li, M., Batsuren, K., Park, H. A., Choi, N. H., & Ryu, K. H. (2015). Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. Journal of cheminformatics, 7(Suppl 1), S9.

Nesterov, Y. (1983, February). A method of solving a convex programming problem with convergence rate O (1/k2). In Soviet Mathematics Doklady (Vol. 27, No. 2, pp. 372-376).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 12, 2825-2830.

Santos, C. D., & Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st International Conference on Machine Learning (ICML-14) (pp. 1818-1826).

Tang, B., Feng, Y., Wang, X., Wu, Y., Zhang, Y., Jiang, M., ... & Xu, H. (2015). A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature. Journal of cheminformatics, 7(supplement 1).

Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Androutsopoulos, I., Gaussier, E., ... & Ngomo, A. C. N. (2012, October). BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering. In AAAI Fall Symposium: Information Retrieval and Knowledge Discovery in Biomedical Text.

Wang, M., & Manning, C. D. (2013). Effect of non-linear deep architecture in sequence labeling. In Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP).