Detecting Drug-Drug Interactions in Biomedical Text

Jason Liu

April 21, 2018

Abstract

Drug-drug interactions are a major source of medication related adverse events. While most are identified in the drug development process, there are some that go undetected. There is a wealth of drug-drug interaction information buried in various medical texts such as case reports, animal studies, clinical trials, and pharmacologic reports and an automated method to mine these texts for drug-drug interaction information is of great interest from a healthcare and patient safety perspective. In this paper, we use an ensemble based convolutional neural network (CNN) model, to classify drug-drug interactions between drug pairs in text. Applied to the 2013 DDICorpus, the single CNN model and the ensemble CNN model achieve similar performance with an F1-score of 0.82 for the single model and 0.81 for the ensemble model.

Project code is available at https://github.com/jliu531/ddi-prediction

1. Introduction

The World Health Organization (WHO) defines pharmacovigilance as the science and activities relating to the detection, assessment, understanding, and prevention of adverse effects or any other drug-related problem.[1] Pharmacovigilance and post-marketing monitoring of drugs for any new adverse events or drug-drug interactions (DDIs) after approval is an important part of the drug development process to ensure the safety of the patient population. Identification of DDIs is an important component of pharmacovigilance as doctors and pharmacists can use this information to make better informed decisions regarding patient care. To facilitate this, there are many DDI databases; however, these are often maintained manually by an expert committee and require tedious upkeep as new biomedical literature is published.

The publication rate of biomedical literature significantly outpaces the rate at which experts can review the literature for drug-drug interactions. In addition, sifting through the literature to assess for DDIs is cumbersome and requires a large amount of man-hours. A search for drug interactions in the PubMed database reveals over 9,000 manuscripts published on the subject in the past year. Developing a model that can automatically evaluate biomedical literature and identify potential DDIs not only saves time in the review process but also gets crucial information to doctors and clinicians faster.

2. Background

To stimulate research in developing models for DDI extraction from text, Isabel Segura-Bedmar and her team at Universidad Carlos III de Madrid hosted the DDI Extraction Challenge (2011 and 2013) where teams around the world were encouraged to develop and submit their own models.[2,3] The 2013 DDI Extraction challenge consisted of two tasks – the first was to recognize and classify drug names and the second was to classify drug-drug interactions between given the sentence text and pairs of drugs. We are most interested in the second task.[3] To facilitate the challenge, Segura-Bedmar and her team created the

DDICorpus, an annotated corpus containing 18,502 pharmacologic substances and 5,028 DDI pairs derived from 792 texts selected form the DrugBank database and 233 Medline abstracts.[4] This dataset is discussed in more detail in Section 3.1.

The highest ranking model in the 2013 DDI Extraction Challenge was developed by Chowdhury and colleagues.[5] Their model broke down the task of DDI detection and classification into two steps. The first step involved classification of DDIs without regard to the specific type of DDI. To do this, a binary Support Vector Machines (SVM) classifier was trained on various linguistic and contextual features, such as negation cues (no, n't, not), whether the same entity was mentioned twice in a sentence, and presence of anti-positive governors. Negative instances of DDIs predicted by this classifier were then discarded from the test set, a hybrid kernel combining a shallow linguistic kernel and Path-enclosed Tree kernel was used to train a classifier to predict the type of drug interaction described. The top run for this model achieved an F1-score of 0.80 for DDI detection and 0.65 for DDI detection and classification.

After the conclusion of the 2013 DDI Extraction challenge, models continued to be developed that improved upon Chowdhury and colleagues' work. Neural network based models in particular showed the most promise. Convolutional neural network (CNN) models were explored by Suárez-Paniaqua et al[6] and Liu et al[7] who achieved F1-scores of 0.6198 and 0.6975 for DDI detection and classification. Lim et al developed a tree-LSTM based model which achieved an F1-score of 0.838 for detection and 0.735 for classification.[8] To our knowledge this model is the current best performing model.

Lim briefly explored neural-network based ensemble methods which improved performance compared to their single models by about one percentage point.[8] Previous research has explored various single CNN models but none have explored the performance of CNN ensembles.[6,7] The objective of this paper is to explore the performance of a CNN ensemble compared to a CNN at classifying drug-drug interactions.

3. Methods

3.1. Data

The DDICorpus is an annotated corpus containing 18,502 pharmacologic substances and 5,028 DDI pairs (not including 'none' class) derived from 792 texts selected form the DrugBank database and 233 Medline abstracts.[4] Each sentence is an annotated XML entry containing the sentence text, drug pair, position of each drug in the pair as it appears in text, and type of DDI described. Each statement can contain multiple drug substances and therefore have multiple combinations drug-drug pairs as interaction candidates. There are about 27,000 pairs in the training set and 5,800 pairs in the test set. The task is to classify a given sentence and drug pair into one of five categories described below.

DDIs are classified as the following:

- Advice – The statement is a recommendation or advice given against the use of two drugs concomitantly.
- Effect – The statement describes what happens pharmacologically, clinically, or therapeutically when two drugs are administered together
- Mechanism – The statement describes a pharmacokinetic interaction
- Int – The statement establishes that an interaction occurs without providing further information.
- None – The statement does not describe a drug-drug interaction

3.2. Baseline Model

We train a Naïve Bayes baseline model with sentence text and drug pair features. Drugs in the test set not seen in the training set are encoded as '<unknown>'.

3.3. Preprocessing

To better generalize methods, drug names in the sentences are blinded using a similar approach from previous studies where the two drugs of interest are replaced with 'drug1' and 'drug2' and all other drugs are replaced by 'drug0'.[7,8] The large number of non-interacting drug pairs can negatively impact performance of models trained on the set. Therefore, these instances are filtered out of the dataset using similar conditions described in previous expriments.[5,7,8] Briefly, the conditions include drugs that have the same name or if they appear in the same coordinate structure. Should these appear in the test set, they are automatically classified as negative. Therefore, if the ground truth for a negative instance is in fact a drug-drug interaction, it will be counted as a false negative for scoring purposes. As shown in Table 1, applying these preprocessing steps to the data provides some alleviation to the class imbalance by removing over 9,000 instances of 'none' but even after this step, the imbalance remains fairly significant.

Table 1: Data removed due to negative-instance filtering

Class	Before Filtering	After Filtering	Number Removed
Advise	826	781	45
Effect	1690	1653	37
Int	188	185	3
Mechanism	1325	1276	49
None	23846	14155	9691

3.4. Convolutional Neural Network Model

For the CNN, we use a similar model to that proposed by Suárez-Paniagua et al.[6] Briefly, we randomly initialized an embedding layer to represent each word in the sentence as a vector. This is followed by a convolutional layer with a widow of size 3. Next, the outputs of this layer are passed to a max-pooling layer followed by a fully connected softmax layer. To prevent overfitting, we perform dropout (rate = 0.5) prior to the softmax layer and add L2-regularization ($\lambda = 3$) to the softmax layer weights.

3.5. CNN Ensemble

The CNN Ensemble uses the model described in Section 3.4 as component models. We use a bagging approach where we train 10 component CNNs on a random subset of 85% of the training data with replacement. We then sum the outputs of those CNNs and choose the class corresponding to the greatest value.

4. Results and Discussion

4.1. Learning Curve

The learning curve for our component CNN is shown in Figure 1. The curve shows the best validation score occurring at around 20 epochs which we used to train our component CNNs for the ensemble

model. The gap between the training curve and validation curve seems fairly small and does not show any evidence of overfitting.

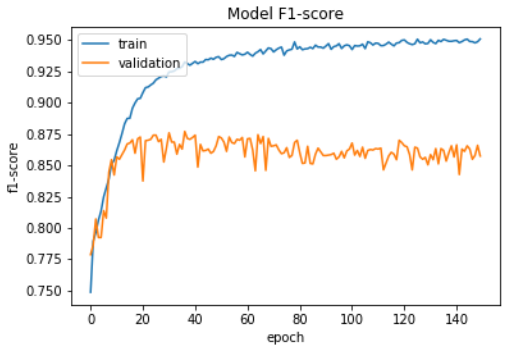

Figure 1: Component CNN Learning Curve

4.2. Model Performance and Analysis

Performance of the models for each DDI classification are shown in Table 2. Both the single CNN and ensemble CNN models performed similarly achieving F1-scores of 0.82 and 0.81, respectively. As expected, the models have the worst performance predicting 'int' classes. This is most likely because 'int' is the least represented class in the dataset with only 185 examples out of 18,050 after negative instance filtering. After the 'none' class, the 'advise' class is the best performing class. One possible reason for this is that sentences in this class use very similar words and phrases such as 'recommended' (ex. Careful monitoring of prothrombin time in patients receiving drug1 and drug2 is **recommended**) and 'should be' (ex. Patients who take both drug1 and drug2 **should be** carefully monitored). These findings are similar to those who have explored identical CNN models.[6,7]

One interesting observation is that there appears to be a trade-off between precision and recall for the single CNN and ensemble models. The single CNN appears to perform better with respect for recall while the ensemble performs better with respect to precision.

Table 2: Results of Naïve Bayes, Single CNN, and Ensemble CNN models

	Naïve Bayes			**Single CNN**			**Ensemble**		
Class	P	R	F1	P	R	F1	P	R	F1
Advise	0.10	0.46	0.17	0.75	**0.58**	**0.65**	**0.80**	0.47	0.59
Effect	0.12	0.36	0.18	0.56	**0.54**	**0.55**	**0.63**	0.46	0.53
Int	0.05	**0.54**	0.08	**0.96**	0.28	**0.44**	**0.96**	0.28	**0.44**
Mechansim	0.12	0.37	0.18	0.70	**0.45**	**0.55**	**0.73**	0.40	0.52
None	0.87	0.28	0.42	**0.87**	0.93	**0.90**	0.85	**0.96**	**0.90**
Overall	0.74	0.30	0.38	**0.82**	**0.83**	**0.82**	**0.82**	**0.83**	0.81

To examine model performance further, we look to the confusion matrices shown in Figure 2.

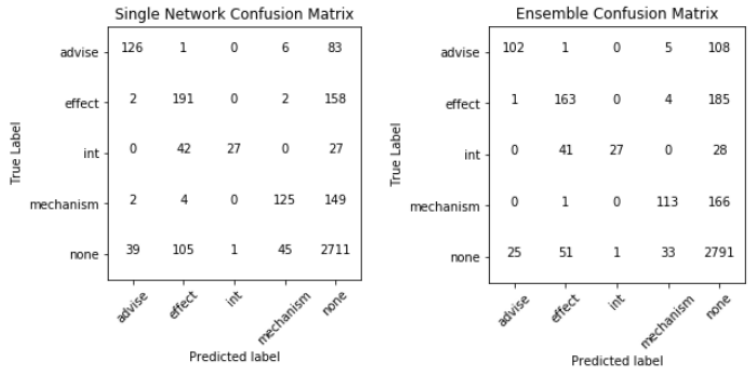

Figure 2: Confusion matrices for single and ensemble CNNs

Examining the confusion matrices, it appears that both models make similar classification errors for confusions between non 'none' pairs. Both models often misclassify 'int' labels as 'effect' labels with similar frequencies and upon further examination, both models misclassify similar sets of sentences. This may be due to two factors: the relative dominance of 'effect' class representation in the dataset over 'int' class (1653 examples vs 185 examples) and the similarity of sentences in the 'effect' and 'int' class. Sentences in the 'effect' class are essentially extensions of the 'int' class in that changing or adding only a few words in an 'int' class sentence can essentially turn it into an 'effect' class sentence. For example, one 'int' sentence the models misclassified an 'effect' was 'concomitant use of drug1 with drug2 may result in an adverse drug interaction'; however, if this sentence were instead 'concomitant use of drug1 with drug2 may result in hypoglycemia', the models correctly classify this sentence as an 'effect'.

From Table 2, we observed that the single CNN appears to perform better with respect for recall while the ensemble performs better with respect to precision. Examining the confusion matrices, give some additional information to why this might occur. Comparing the rightmost columns of the two confusion matrices, the ensemble model appears to be making more incorrect 'none' predictions compared to the single CNN model. One hypothesis for this occurrence is the class imbalance with 'none' examples. Since each component model of the ensemble is trained on data with a relatively greater amount of 'none' examples compared to other examples, combining the models together may exacerbate this overrepresentation causing more incorrect 'none' predictions.

5. Conclusion and Future Directions

In this paper, we demonstrated a CNN and CNN ensemble model to predict drug-drug interactions given a piece of text. Performance of the single CNN and CNN ensemble model were similar with trade-offs being made between precision and recall between the two types of models.

There are many avenues to explore for improving these models. Use of a pre-trained embedding layer rather than a randomly initialized one has led to increased performance for many neural-network models on this dataset.[6–8] In addition, many models also include position embeddings that encode the distance between the two interacting drugs.[6,7] Incorporating these features into future single CNN and ensemble CNN models may be able to widen the performance gap between the single and ensemble model.

Class imbalance in the DDI Corpus was a major obstacle in terms of model performance. Techniques to address this such as undersampling the training data to train future models may improve performance. A

recent study by Collell et al. described a threshold moving technique to address class imbalance issues in ensemble models.[9] Future work exploring the use of this technique to improve CNN ensemble performance may yield interesting results.

6. References

1. WHO | Pharmacovigilance. WHO. http://www.who.int/medicines/areas/quality_safety/safety_efficacy/pharmvigi/en/. Accessed February 10, 2018.
2. Segura Bedmar I, Martinez P, Sánchez Cisneros D. The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. 2011.
3. Segura-Bedmar I, Martínez P, Zazo MH. 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)*. Vol 2. ; 2013:341–350.
4. Herrero-Zazo M, Segura-Bedmar I, Martínez P, Declerck T. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *J Biomed Inform*. 2013;46(5):914-920. doi:10.1016/j.jbi.2013.07.011
5. Chowdhury MFM, Lavelli A. FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Vol 2. ; 2013:351–355.
6. Suárez-Paniagua V, Segura-Bedmar I, Martínez P. Exploring convolutional neural networks for drug–drug interaction extraction. *Database J Biol Databases Curation*. 2017;2017. doi:10.1093/database/bax019
7. Liu S, Tang B, Chen Q, Wang X. Drug-Drug Interaction Extraction via Convolutional Neural Networks. *Comput Math Methods Med*. 2016;2016. doi:10.1155/2016/6918381
8. Lim S, Lee K, Kang J. Drug drug interaction extraction from the literature using a recursive neural network. *PLOS ONE*. 2018;13(1):e0190926. doi:10.1371/journal.pone.0190926
9. Collell G, Prelec D, Patil KR. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*. 2018;275:330-340. doi:10.1016/j.neucom.2017.08.035