

Drug-Drug Interaction Detection (DDI) Over the Social Media using Convolutional Neural Networks

Kelechi Iwuorie ¹ and Sabah Mohammed ²

¹Lakehead University

²Affiliation not available

October 30, 2023

Abstract

A project on DDI extraction from medical text using a Convolutional Neural Network Model.

Drug-Drug Interaction Detection (DDI) Over the Social Media using Convolutional Neural Networks

Kelechi Iwuorie
Department of Computer Science
Lakehead University
Thunder Bay, Canada
kiwuorie@lakeheadu.ca

Dr. Sabah Mohammed
COMP5800 Supervisor
Department of Computer Science
Lakehead University
Thunder Bay, Canada
mohammed@lakeheadu.ca

Abstract— Drug-drug interaction (DDI) detection is a challenging problem for drug manufacturers, drug regulatory authorities, and medical professionals alike. It is impossible to run trials or be aware of every single case involving an entire population. Research in the use of social media data is currently gaining attention, and with the application of machine learning techniques has been successfully applied in businesses. This paper presents a project extracting DDI from biomedical text using a Convolutional Neural Network (CNN) classifier. The classifier is trained on the SemEval 2013 DDIExtraction challenge dataset and aims to automatically learn the best feature representation on the input of the given task. Different models have been proposed, which make use of position embeddings in combination with word embeddings trained on the machine learning model to learn features. Word embeddings are necessary for providing dense vector representation of words that can be trained, but a large amount of data is required to train an effective vector representation of words. To compensate for the shortage of data, the CNN model is trained on a pre-trained PubMed word embedding, which provides a vector dimension of size 200 for the representation of each word. This project aims to provide a trained CNN model for which vector representation of words is provided by weights that have been trained for medical text classification purposes.

Keywords—*drug-drug interactions; convolutional neural networks*

I. INTRODUCTION

Increased awareness of adverse drug reactions has made this a priority for drug consumers and regulators all over the world. Adverse drug reactions are undesired reactions on the body caused by the intake of medication, and the process for monitoring drug use to detect these reactions is pharmacovigilance. Although there are measures and standards put in place by drug regulatory agencies to guarantee the safety of a drug before it is released for public use, the clinical trials

covered by these regulations are still limited. Drug-drug interaction (DDI) is a phenomenon that occurs when the activity of a drug is influenced by another drug, which happens when two or more drugs are taken simultaneously. It is an essential area in healthcare that emphasizes patient safety, as unexpected side effects may lead to deaths or an increase in healthcare costs. Clinical trials do not account for a lot of variables likely to occur from drug use, and the trials are usually conducted on a small group of people, which cannot account for all gender, age, and race demographic of the population. There is also the case of rare adverse reactions, which may take longer to manifest and can only appear through continued drug use [5]. The limitations of pre-approval clinical trials make it impossible to thoroughly assess the consequences of the use of a drug before it is released [6]. This makes surveillance of drugs after their release into the market by regulatory agencies and drug manufacturers a priority. Although reporting systems are put in place to help with this process, these reporting systems are still underutilized by clinicians and patients. It is also found that the majority of adverse drug reactions are unreported in developing countries; hence there is a need for more automated processes for tracking, identifying, and reporting adverse drug reactions [7]. There are different methods used for finding DDI, primarily through clinical trials and pharmacology experiments. These methods are, however, limited to the amount of information that can be obtained as they do not cover every possible criterion for DDI detection. Therefore, extracting information from both structured and unstructured sources of data can significantly improve the identification and extraction of relevant DDI information. The use of online drug interactions databases created to help detect DDI makes this

possible. As the size and use of healthcare social media grow, alternative sources of data for health surveillance create opportunities for new or undiscovered DDIs to be found. However, the rapid growth of data makes it impossible for professionals to keep up, which drives the need for techniques that reduce the time spent by professionals in reviewing the literature.

II. RELATED RESEARCH

A good amount of effort has been put into research on DDI detection in text, with different studies making use of different data sources for DDI signal detection. In order to conduct an experiment on identifying DDI in user comments from healthcare social media platforms, [1] proposed a novel idea that employed traditional text mining techniques for identifying DDI in text. This study speculated the assumption that if two drugs and an ADR (adverse drug reaction) are mentioned multiple times in a thread of comments, an association can be inferred between the drugs and ADR. Using the DrugBank as the standard on which performance was evaluated, they experiment for three criteria to detect DDI by mining the association between the two drug names and the ADR within multiple user comment threads from MedHelp. Therefore, a count of the number of threads containing the ADR (R), the drugs (D1 and D2), and the drugs and ADR (D1, D2, and R) are calculated to find the co-occurrence of DDI in the thread. [9] demonstrated the use of fundamental knowledge in the form of relationships to discover potential DDI in biomedical literature, in order to show that potential DDI occurring in patient clinical data can be detected. This involved the construction of DDI schemas based on relationships extracted from the SemMed database. The idea was to identify theoretical DDIs that may not have been found through clinical trials by associating two separate assertions which are combined through gene and biological functions. Although it is shown to have good performance in DDI signal detection, the performance of these methods is dependent on text mentions of the drugs and ADR. However, most data sources suffer from the limitations of a low reporting ratio. Machine learning and Natural Language Processing (NLP) research have provided new insight towards achieving better performance in DDI signal detection. These methods, however, rely heavily on comprehensively annotated corpora for the training of efficient models. [8] created a manually annotated corpus from a combination of biomedical texts compiled from both Medline and DrugBank databases. The corpus was found to contain a total of 18,502 single mentions of pharmacological substances

and 5028 DDIs, which included both pharmacokinetic and pharmacodynamic DDIs. The DDIs are divided into four types, *mechanism*, *effect*, *advice*, *int*. The dataset, after its use in the SemEval 2013 DDIExtraction challenge, has become the benchmark standard for training DDI signal detection models. With the creation of the annotated DDI corpus, more research has been conducted on the benefit of machine learning for DDI signal detection. [11] proposed a technique that made use of Support Vector Machines and linear kernel to train a classifier on the DDI corpus. This study demonstrated the usefulness of supervised machine learning models for detecting DDI signals in biomedical literature given a rich set of lexical and syntactic features. The system incorporated features such as a dependency graph, parse tree and noun phrase-constrained coordination features. Although these SVM systems have remarkably good performance, they require extensive feature engineering, which tends to be complex and may not be applicable for large scale datasets. The systems may also suffer from the fact that most of its features are generated with NLP toolkits, which are imperfect and cause errors that propagate in the system. The resurgence of deep neural networks has renewed interest in the application of such methods for DDI detection. [10] trained a Recurrent Neural Network (RNN) model on the DDI corpus, the study evaluated the use of a Word-Level RNN architecture where the input was a tokenized n-gram of words in a sentence. Words are represented by their low-dimension vector representation as a means to learn their positions relative to the drug mention in the sentence. The study also evaluates the use of Character-Level RNN, which made use of randomly initialized character embeddings to form word vectors. An advantage of deep neural network models over traditional machine learning models is that they do not need manually defined features to have good classification performance. [2] demonstrates this in their application of a Convolutional Neural Network (CNN) to train a DDI detection classifier on the DDI corpus. This method makes use of word embeddings and position embeddings to capture semantic information of words and relative distances between words in the sentence and the drug pair. [4] trained a CNN classifier which only used word-level embeddings for its input feature. [3] research the use of Long-Short Term Memory (LSTM) based neural networks as opposed to the use of CNN. The study proposed three models, which are; B-LSTM, AB-LSTM, and joint AB-LSTM. A Bi-LSTM is used to encode word and position features, while the B-LSTM and AB-LSTM use max-pooling and attentive pooling

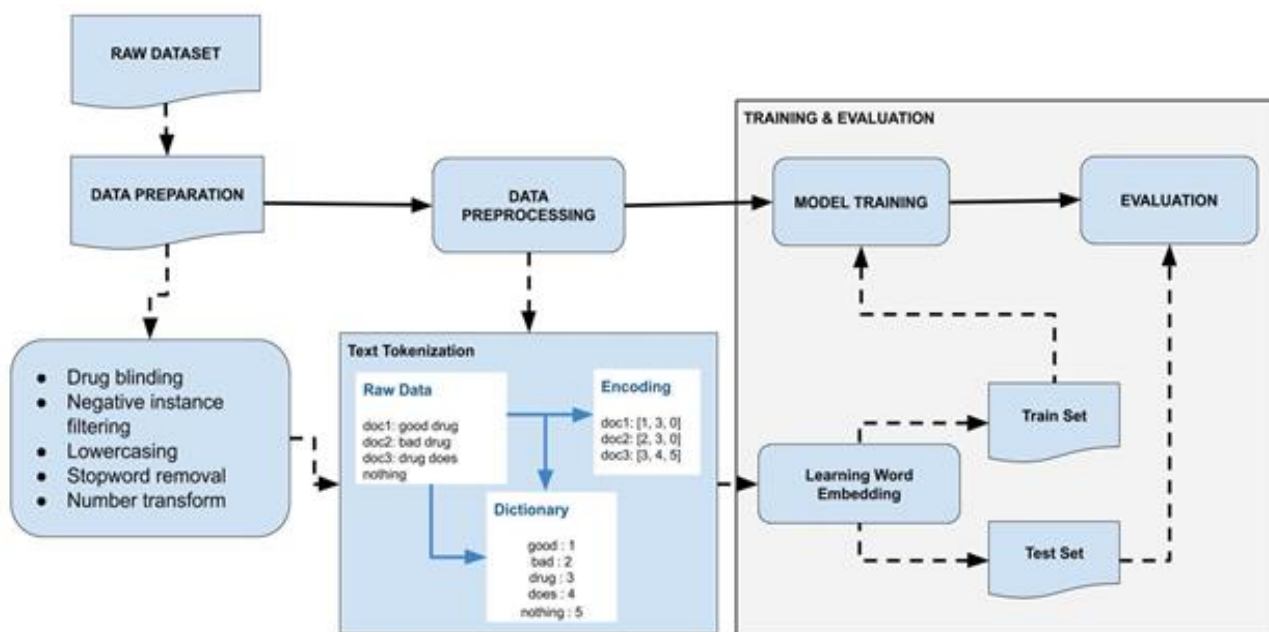


Figure 1. Workflow of DDI classification method.

to extract fixed-length features. Studies involving the use of deep neural network models to train DDI detection classifiers have demonstrated that these networks outperform traditional machine learning and rule-based DDI extraction models. Although these models have high performance, CNN models have given a better performance than other candidates making it a good choice for DDI extraction. While the LSTM models are considered the highest performing models after CNN, and can potentially surpass CNN models depending on the feature engineering, these models require high computation cost compared to their CNN counterparts. Despite the potential for LSTMs to surpass CNNs, the gap between both models is considerably small, making the computation speed and cost-effectiveness a more viable trade-off than the increased performance of LSTM. When compared with RNNs, CNN is found to be more suitable for spatial data representations and more powerful than RNN. Despite RNN being more ideal for speech and text analysis, CNNs have performed better for text classification tasks. A downside of CNN models is that the performance of the model depends on the amount of data it is trained on [12].

In this project, we train a CNN classifier to detect DDI on the SemEval 2013 DDIE extraction Challenge dataset. The model makes use of pre-trained PubMed word embeddings. Several preprocessing techniques and methods applied by previous studies on state-of-the-art DDI extraction systems have been incorporated in the project. The use of a pre-trained word embedding drives the expectation that the model can produce a good

performance for classifying DDIs given that the trainable weights used for vector representation of words are trained for biomedical text classification purposes.

III. METHODOLOGY

The DDI corpus is an annotated corpus which has been provided as the benchmark standard for training and evaluating DDI extraction classifiers. The corpus combines biomedical texts from documents describing DDIs in DrugBank database and MedLine abstracts, which is distributed in XML documents so as to unify the multiple formats of the documents. The corpus is found to contain 792 texts about DDI from the DrugBank database and 233 texts from MedLine abstracts and is manually annotated with 18 502 pharmacological substances and 5028 DDIs. Each interacting pair of substances in the corpus is annotated with one of the following types, mechanism, effect, int [8].

- Mechanism: This DDI type describes the pharmacokinetic relationship between the substance pair mentioned in the text.
- Effect: This type is used to describe the annotation of substances where an effect can be inferred when both substances interact.
- Advice: This type is used to annotate DDIs, where a recommendation or advice is given.

- Int: This type represents texts where DDI is established between two substances without providing any additional information.

The distribution of DDI types in the corpus is shown in table 1.

Table 1: DDI types in DDI corpus [4]

DDI Types	DrugBank	MedLine	Total
Advice	1035	15	1050
Effect	1855	214	2069
Int	272	12	284
Mechanism	1539	86	1625
Total	4701	327	5028

Extracting DDI from the DDI corpus is a multiclass classification problem for all possible drug pairs in a sentence, as each pair can be classified into one of the predefined DDI type or as a non-interacting pair. Figure 1 describes the workflow of the approach taken to build and train a CNN classifier for extracting DDI from the corpus. The preparation and preprocessing module first performs drug blinding, number filtering, and duplicate instance filtering on the text. Tokenization is performed, and the tokenized text is normalized, after normalization, the text is trained on a CNN model and all DDI instances are classified as a DDI type or non-interacting pair.

A. Preprocessing

1) Negative Instance Filtering

To ensure the use of standard methods which provide good results, preprocessing steps that have been applied by state-of-the-art CNN-based studies [2, 4] was applied to blind drugs in a sentence. The drug blinding process replaces the two interacting entity pairs in the sentence with “drug1” and “drug2” according to their order of occurrence. All other drug mentions in the sentence are replaced by “drug0”, therefore given an example; “The effects of *adenosine* are antagonized by *methylxanthines* such as *caffeine* and *theophylline*.”, where the drug entities are highlighted in italics, table 2 shows how DDI instances are generated to blind the drug entities for all drug pairs in the sentence and the snippet in figure 2 shows the function applied to achieve this process.

After blinding drugs in the sentence, negative instances are removed for non-interacting DDI pairs. The processing of negative instance filtering is performed to provide a more balanced distribution between DDI and non-DDI instances in the dataset. This is due to the fact that non-DDI instances that highlight non-interacting drug pairs make up over 70% of the dataset, and some of the instances have been found to be duplicates of

```
def blind_interacting_pair(data):
    new_set = []
    for text, val, e1, e2 in data:
        if e1 != e2:
            text = text.replace(e1, 'drug1')
            text = text.replace(e2, 'drug2')
            text = blind_non_interacting_pair(text)
            new_set.append((text, val))
    return new_set
```

Figure 2. Snippet for drug blinding in text.

themselves or follow a set of special rules provided by [2]. Taking this into account, a sentence containing the signals “drug1” and “drug2” is identified as a negative instance if;

- drug1 and drug2 are the same.
- drug1 is an abbreviation or acronym of drug2.
- drug2 is a type of drug1 (subset).
- a sentence containing drug1 and drug2 is an exact duplicate of another.

The first criterion is determined by lowercasing and matching the strings for drug1 and drug2. As seen in table 2, the fourth and fifth instances can be identified as negative instances as they fall under the third criterion. Rules are defined using a regular expression to convert these instances to exact duplicates of each other. After conversion, these duplicates, along with other instances that are identified by the fourth criterion, are filtered out by simple rules for removing duplicate instances.

2) Tokenization

Correct preprocessing techniques help with improving the performance of classification models; therefore, some of the standard text cleaning steps are conducted, i.e., lowercasing and punctuation removal. Lowercasing and symbol removal is the most common preprocessing techniques applied in NLP. Words are converted to lowercase to prevent distinctions being made between similar words. Punctuations, alphanumeric characters, and symbols are removed from the document as these particular words and symbols do not add to the semantic meaning of the text. This is performed by replacing these special characters with white spaces. Stop-words are words that do not provide any semantic or syntactic meaning to the text. These words are removed so that the model maintains focus on words that are essential for the text classification problem. For the purpose of this project, the focus is maintained on the removal of medical stop-words. Also, following the practices of [12], numbers are transformed into a consistent form. This is due to the fact that there is a frequent occurrence

Table 2: Candidates in sentence after drug blinding [2]

Drug Pair	DDI sentence after drug blinding
(adenosine, methylxanthines)	The effects of <i>drug1</i> are antagonized by <i>drug2</i> (e.g. caffeine and theophylline).
(adenosine, caffeine)	The effects of <i>drug1</i> are antagonized by methylxanthines (e.g. <i>drug2</i> and theophylline).
(adenosine, theophylline)	The effects of <i>drug1</i> are antagonized by methylxanthines (e.g. caffeine and <i>drug2</i>).
(methylxanthines, caffeine)	The effects of adenosine are antagonized by <i>drug1</i> (e.g. <i>drug2</i> and theophylline).
(caffeine, theophylline)	The effects of adenosine are antagonized by methylxanthines (e.g. <i>drug1</i> and <i>drug2</i>).

```
'These pharmacokinetic effects seen during drug1 coadministration can result in increased clinical effects of both drug0 and drug2.',
'Therefore, co-administration of drug1 with other drug0 that are extensively metabolized by this isozyme and which have a narrow therapeutic
index, however co administration of drug1 with either drug2 or drug0 led to increased plasma concentrations of drug0.',
'Drugs that reportedly may increase oral drug1 response, ie, increased prothrombin response, in man include:alcohol*;allopurinol;aminosalicylic
acid;patients who begin taking drug1 or who increase their drug1 dose or any other drug0 while taking drug0, drug2, or drug0 may develop toxicity;
Caution should be observed when drug1 is coadministered with other drug0, drug0, drug2, or drug0, as these agents may increase respiratory and
cardiovascular depression.',
'drug0 - Taking drug1 and a drug2 together may increase the risk of convulsions (seizures).',
'Consider additive drug0 effects and confusional states to emerge, if drug1 is given with drug2 or drug0.',

'these pharmacokinetic effects seen during drug1 coadministration can result in increased clinical effects of both drug0 and drug2',
'therefore co administration of drug1 with other drug0 that are extensively metabolized by this isozyme and which have narrow therapeutic
index, however co administration of drug1 with either drug2 or drug0 led to increased plasma concentrations of drug0',
'drugs that reportedly may increase oral drug1 response ie increased prothrombin response in man include alcohol allopurinol aminosalicylic
acid;patients who begin taking drug1 or who increase their drug1 dose or any other drug0 while taking drug0 drug2 or drug0 may develop toxicity;
caution should be observed when drug1 is coadministered with other drug0 drug0 drug2 or drug0 as these agents may increase respiratory and
cardiovascular depression.',
'drug0 taking drug1 and drug2 together may increase the risk of convulsions seizures',
'consider additive drug0 effects and confusional states to emerge if drug1 is given with drug2 or drug0',
```

Figure 4. Result before and after performing text preprocessing on dataset.

of numbers in the DDI corpus, therefore, transforming numbers to a consistent expression prevents a change in the semantic expression of a sentence in the corpus. For examples, given the sentence; “The serum concentration of phenytoin increased dramatically from 16.6 to 49.1 microg/ml”, the values are transformed to the form ‘number’ and the result returns; “The serum concentration of phenytoin increased dramatically from number to number microg/ml”. These preprocessing steps are performed using Keras NLP preprocessing toolkit. Figure 3 shows the snippet after for cleaning text and figure 4 shows the result before and after cleaning.

```
cleaned_doc = list()
for sentence in doc:
    token = text_to_word_sequence(sentence)
    table = str.maketrans('', '', string.punctuation)
    token = [word.translate(table) for word in token]
    token = number_blinding(token)
    token = [word for word in token if len(word) > 1]
    token = ' '.join(token)
    cleaned_doc.append(token)
return cleaned_doc
```

Figure 3. Snippet to text preprocessing

After the transformations are performed, the sentences are tokenized, also using Keras preprocessing toolkit by transforming words in a sentence into a real number

representation. The process takes a text and returns a tokenized sequence of values, and it is performed by generating a dictionary of words in the document and a unique index for each word. Words are separated by means of an n-gram model for which the dictionary can contain individual words, the unigram representation of word is used. Tokenization creates a matrix representation of instances that is suitable for a CNN architecture, as all instances should have the same length. Therefore zero-padding is applied to shorter sequences for even distribution [4]. The length is given by the size of the longest sentence in the corpus, and shorter sentences are padded with zero up to the length to make the dimensions.

B. Building the Model

The CNN model applied for this DDI extraction task is a simple 1-layer model, a variant of models commonly used for prototyping. However, unlike standard practices, a pre-trained word-embedding has been applied in the embedding layer of the network architecture.

1) Embedding Layer

Although words in the matrix are represented by a real value vector that identifies their location or indexing in the word vocabulary, the provided matrix does not provide any syntactic or semantic meaning for the words in the corpus. Also, CNN models are found to have the best performance when working with dense

vectors; therefore, a pre-trained word embedding is used to substitute each word by its word embedding vector. This produces an embedding matrix with dimensions: $N \times M$, where N is the length of vocabulary and M , is the dense vector representation of each word in the vocabulary. Words not found in the pre-trained word embedding are initialized with zero values for their vector representation. The pre-trained word embedding is the BioWordVec, a set of word-vector embedding provided by [13]. The embedding is trained on 27,599,238 articles extracted from PubMed database, creating dense vector representations for 3,658,450,658 tokens. The embeddings are extracted via the Gensim toolkit and used in the CNN embedding layer as the lookup table for training the DDI sentences on the model. The lookup table takes instances of the DDI corpus as input and generates their vector representation in the embedding layer. The embedding layer is initialized with the vocabulary size (N) as its input dimension and the lookup table dimension (M) as its output dimension.

2) Convolutional Layer

After obtaining the input matrix, this layer applies a filter matrix f with a sliding window of size w to extract higher-level features. These features are represented as score sequences for the sentence. It is essential to take into consideration the size of the filter, as this can directly impact the performance of the model [4]. For this project, the convolutional layer adopts 128 filters with a window size of 3 as the is a more commonly adopted parameter.

3) Pooling Layer

The goal of the pooling layer is the extraction of the most relevant features of each filter in the network. This is done by the use of an aggregation function commonly known as max-pooling, pooling layers perform dimensionality reduction on the output matrix of the convolution layer by subsampling and combining neighboring elements. Therefore, a 1-dimensional max-pooling layer with a pool size of 5 and a stride of 1 is used to perform dimensionality reduction and extract the essential features in the network.

4) Softmax Layer

This layer is necessary to prevent neural networks from overfitting, therefore like [2], a softmax layer is applied to drop units from the network during training randomly. The fully connected softmax layer makes use of randomly initialized weights to compute the output prediction values for the classification of the DDI sentences as one of the given five classes (mechanism, advice, effect, int, non-ddi).

IV. PROTOTYPING AND DISCUSSION

1) Prototype Settings

As stated in Section III, the instances of the DDI corpus are classified into five classes (the four different types of DDI and one non-DDI category). After reshaping the corpus, a total of 30902 instances are extracted for the interacting pairs in the corpus. Instance classified with a DDI type amount to 3920 while the non-DDI classes provide 23672 instances. After drug blinding and negative instance filtering is applied to the dataset, it is normalized to 3245 for instances classified with a DDI type and 17048 for instances with non-DDI class. After preprocessing and tokenization are performed, the dataset is split by a ratio of 80:20, 80% is used for training the model, and 20% is used as the test set for evaluating the model. In order to investigate the effects of different factors on the model, 200 instances are selected from the dataset (100 DDI instances and 100 non-DDI instances). These instances are removed from the dataset before any preprocessing or tokenization is performed and, therefore, contain words that may not be in the vocabulary.

The CNN model makes use of one kind of filter, which is set to 5 for convolution, and 128 filters are used with rectified linear units as the activation function for the convolutional layer. The softmax function is used as the activation function for the output layer with an output size of 5 (for the number of classes or categories in the dataset). The preprocessing methods and CNN model are coded using Natural Language Processing libraries built with Python. The model is trained using Google Colab, an open-source Python compiler that enables compilation and execution of Python code on cloud provided GPUs and CPUs. When executed on a GPU, the model trains for a few minutes while a longer amount of execution time is consumed for CPUs.

2) Discussion

The performance of the model is compared on different versions of the test data. The first version compares the model against the filtered and preprocessed dataset. The result of the model is provided in figure 3. The result gives the model an average performance accuracy of 81.69%, with 3316 test samples correctly classified out of a combined total of 4059 samples (advice: 117, effect: 295, int: 30, mechanism: 207, non_ddi: 3410).

	precision	recall	f1-score	support
advice	0.40	0.37	0.38	117
effect	0.47	0.37	0.41	295
int	0.50	0.10	0.17	30
mechanism	0.39	0.27	0.32	207
non_ddi	0.88	0.92	0.90	3410

Figure 5. Evaluation result on test samples

From figure 5, the model achieves the highest performance when detecting samples for the non-DDI category, with a precision score of 0.88 and an F1 score of 0.90. The increase in performance in this category is attributed to the distribution of the dataset the model is trained on. The number of instances in the training set for the five DDI types (advice, effect, int, mechanism, and non-ddi) are 826, 1687, 188, 1319, and 23 772, respectively, with non-DDI constituting over 60% of the entire dataset. The model, however, achieves a lower performance when classifying instances for other DDI classes. Having an imbalanced data provides a major impact on the performance of the model.

In order to compare the performance of the model against a different version of the test data, performance is evaluated on a set of samples that were not included during preprocessing and may contain words that do not belong to the vocabulary. The first experiment evaluates the model on a version of these samples that have not been blinded or filtered, with only the basic preprocessing performed on the text (the result is shown in figure 6).

	precision	recall	f1-score	support
advice	1.00	0.28	0.43	18
effect	0.67	0.34	0.45	53
int	0.00	0.00	0.00	0
mechanism	0.87	0.48	0.62	27
non_ddi	0.62	0.92	0.74	99

Figure 6. Evaluation result on unfiltered probing samples

In order to ensure credibility of the result, all duplicate instances are filtered out, leaving only the unique samples for a total of 197 samples (advice: 18, effect: 53, int: 0, mechanism: 27, non_ddi: 99). The model achieved an accuracy of 64.47%, correctly finding the DDI category of 127 out of 197 samples. It was found to have a good performance for all categories except the “int” which did not have any sample as this category has the smallest distribution among all the DDI instances. Negative instance filtering and drug blinding is applied to the samples and tested on the model, this version of the test does produce any major difference in the result (figure 7) with the counterpart (figure 6). However, there is 0.63 percent drop in performance accuracy as

	precision	recall	f1-score	support
advice	1.00	0.28	0.43	18
effect	0.90	0.21	0.34	43
int	0.00	0.00	0.00	0
mechanism	0.82	0.38	0.51	24
non_ddi	0.60	0.98	0.74	92

Figure 7. Evaluation result on filtered probing samples

this model detects DDI correctly for 113 out of 177 samples.

The reduction in sample size happens as a result of the filter rules applied, which also identifies more duplicates in “effect” and “non_ddi” instances than other instances. Negative instance filtering is beneficial to this model as it is able to identify negative instances and duplicates and removes a large number of imbalanced samples in the dataset [2, 4]. The use of pre-trained word embeddings is another beneficial aspect of this model, and as when originally trained without the pre-trained PubMed embeddings on the same parameters, the model performed considerably poorly as opposed to its current performance. Given the amount of data, it is easily understandable that the system performs better for the “non-ddi” than other categories, especially the “int” category. This development can be attributed to the sample distribution among these categories, as CNN models are found to have the best performance given sufficient data. A possible improvement is the provision of more annotated data to balance out the distribution between DDI samples and non-DDI samples. Although state-of-the-art models are found to use trained word embeddings and position embeddings to improve performance, this system made use of only pre-trained word embeddings and achieved considerably good performance. It is speculated that the inclusion of position embeddings of words can improve the DDI detection of the model [4], especially for DDI instances.

V. CONCLUSION

In this project, previous practices are adopted to build a CNN-based classifier for extracting DDI from text. Pretrained word embeddings which capture the semantic meaning of words are used to represent the embedding model. The model is trained and evaluated on the 2013 DDIExtraction challenge corpus and demonstrates that given sufficient data, the use of task-specific pre-trained word embeddings can provide a good model for identifying DDI in text.

REFERENCES

- [1] H. Yang and C. Yang, "Harnessing Social Media for Drug-Drug Interactions Detection", 2013 IEEE International Conference on Healthcare Informatics, Philadelphia, PA, 2013, pp. 22-29.
- [2] S. Liu, B. Tang, Q. Chen and X. Wang, "Drug-Drug Interaction Extraction via Convolutional Neural Networks", Computational and Mathematical Methods in Medicine, vol. 2016, pp. 1-8, 2016.
- [3] S. Sahu and A. Anand, "Drug-drug interaction extraction from biomedical texts using long short-term memory network", Journal of Biomedical Informatics, vol. 86, pp. 15-24, 2018.
- [4] V. Suárez-Paniagua, I. Segura-Bedmar and P. Martínez, "Exploring convolutional neural networks for drug-drug interaction extraction", Database, vol. 2017, 2017.

- [5] S. Sunil, S. Palaian, B. Shestha, S. KC and S. Khanal, "The Potential Role of Social Media in Pharmacovigilance in Nepal: Glimpse from a Resource-limited Setting", *Journal of Clinical and Diagnostic Research*, vol. 13, no. 3, pp. 4 - 7, 2019.
- [6] A. Sarker et al., "Utilizing social media data for pharmacovigilance: A review", *Journal of Biomedical Informatics*, vol. 54, pp. 202-212, 2015.
- [7] D. Pappa and L. Stergioulas, "Harnessing social media data for pharmacovigilance: a review of current state of the art, challenges and future directions", *International Journal of Data Science and Analytics*, vol. 8, no. 2, pp. 113-135, 2019.
- [8] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez and T. Declerck, "The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions", *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 914-920, 2013.
- [9] R. Zhang et al., "Using semantic predications to uncover drug–drug interactions in clinical data", *Journal of Biomedical Informatics*, vol. 49, pp. 134-147, 2014.
- [10] R. Kavuluru, A. Rios and T. Tran, "Extracting Drug-Drug Interactions with Word and Character-Level Recurrent Neural Networks", 2017 IEEE International Conference on Healthcare Informatics (ICHI), 2017.
- [11] S. Kim, H. Liu, L. Yeganova and W. Wilbur, "Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach", *Journal of Biomedical Informatics*, vol. 55, pp. 23-30, 2015.
- [12] Z. Zhao, Z. Yang, L. Luo, H. Lin and J. Wang, "Drug drug interaction extraction from biomedical literature using syntax convolutional neural network", *Bioinformatics*, p. btw486, 2016.
- [13] Y. Zhang, Q. Chen, Z. Yang, H. Lin and Z. Lu, "BioWordVec, improving biomedical word embeddings with subword information and MeSH", *Scientific Data*, vol. 6, no. 1, 2019.