International Journal of Multidisciplinary Innovation and Research Methodology (IJMIRM)
ISSN: 2960-2068, Volume 3, Issue 3, July-September, 2024, Available online at: https://ijmirm.com

Application of Biomedical Informatics Methods to Find Drug-Drug Interactions

Saeed Mohammed Aladadi[1], Mohanad Ahmad Alghamdi[2], Muath Rebdi Alrebdi[3]

[1]Senior Specialist Biomedical, Sciences Alnakheel Medical Complex, Riyadh KSA
[2]Pharmacist, Alnakheel Medical Complex, Riyadh KSA
[3]Health Informatics, Alnakheel Medical Complex, Riyadh KSA

ABSTRACT

In order to improve patient wellbeing and advance clinical navigation, biomedical informatics technologies play a key role in identifying drug interactions (DDIs). Drug interaction (DDI) detection has emerged as a critical component of overall health security. Thus, applying text mining techniques to distinguish DDIs from biological writing has garnered significant attention. Nevertheless, this investigation is just getting started, and there is plenty of room to advance its presentation. We introduce a DDI extraction method based on syntactic convolutional neural network (SCNN). This approach suggests a new kind of word embedding termed syntax word embedding that takes advantage of the syntactic information present in a sentence. Knowing where each word is and how to highlight its grammatical forms allows us to expand its embedding. Because of this, the auto-encoder eventually figures out that the thick real worth vector is the standard word pack highlight (the inadequate 0-1 vector). The last step in removing DDIs from medical records is training the softmax classifier with a mix of conventional and embedding-based convolutional highlights. When compared to other state-of-the-art methods, SCNN performs better in terms of presentation, according to experimental results on the DDI Extraction 2015 corpus (F-score = 0.688).

Keywords: Biomedical Informatics Methods, Drug-Drug Interactions, Detecting Drug-Drug Interaction, Convolutional Neural Network

INTRODUCTION

In today's healthcare, biomedical informatics has become an essential tool for identifying, analyzing, and forecasting drug interactions (DDIs). DDIs occur when two or more medications are taken concurrently and alter each other's properties within the body, potentially causing negative reactions or disappointing recovery. With polypharmacy becoming more complex in quiet consideration, particularly when managing chronic and multiple disorders, it is now essential to anticipate and mitigate harmful interactions. Biomedical informatics approaches, which include computational methods, clinical informatics, and bioinformatics, play a major role in supporting this test.

Using large scale data sets and information mining techniques is one of the primary uses of biomedical informatics in DDI detection. Clinical preliminary data, pharmacovigilance frameworks, and electronic health records (EHRs) generate vast amounts of verified data on medication side effects, patient outcomes, and unpleasant events. Informatics devices can sift through these datasets using sophisticated calculations and AI techniques to identify previously unknown correlations or validate existing ones. Additionally, to further improve the scope of DDI detection, natural language processing (NLP) is used to extract data from unstructured clinical notes and exploration writing.

Apart from information mining, network-based techniques in framework science have emerged as effective ways to identify DDIs. These approaches represent pharmaceuticals, their goals, and natural processes as complex networks, with medications or proteins as hubs and their interactions as edges. Scientists can predict possible connections in light of similar targets, pathways, or metabolic catalysts by examining the topography of these networks. Additionally, framework science provides more precise intercessions and facilitates a deeper comprehension of the basic systems underlying drug interactions.

Additionally, computational pharmacology enhances DDI expectation by the application of chemoinformatics, pharmacokinetic models, and atomic docking. These techniques replicate the subatomic interactions between different medications and molecules, transporters, or receptors while accounting for predictions regarding the potential for one drug to enhance or suppress the effects of another. In particular, pharmacokinetic models are able to simulate the processes of drug absorption, distribution, metabolism, and excretion (ADME), providing insights into the manner in which different medications may seek out comparable metabolic pathways, leading to interactions. Pharmacogenomics is another essential component of biomedical informatics in DDI research. Certain patients may be more susceptible to drug-drug interactions (DDIs) because to genetic variations in drug-using molecules, carriers, or drug targets that affect how individuals respond to particular medications. Customized risk assessments can be created by incorporating genetic data into drug interaction models, taking into account safer and more potent medications overall.

By combining large datasets, sophisticated computational models, and research using organic frameworks, biomedical informatics techniques are revolutionizing the identification and prediction of drug interactions. These techniques enable more precise identification of potentially harmful interactions, reduce the risk of hostile events, and contribute to safer, more individualized medicine. Biomedical informatics will continue to be an essential tool for managing the complexity of polypharmacy and achieving tolerable outcomes as the field develops.

REVIEW OF LITERATURE

Hashimoto, K. et al. (2014)provide a novel approach to word representation and generation skills using predicate-contention structures, advancing the study of natural language processing (NLP). The paper emphasizes how important predicate-contention structures are for working on semantic comprehension since they capture the relationships between action words and their contentions. In order to mutually develop better depictions, the suggested model combines both word portrayals and compositional skills. The ability to recognize syntactic and semantic relationships in complicated phrases is the focus of this collaborative learning system. The study evaluates the model on various phonetic tasks, such as semantic job marking and rework localization, and demonstrates that the joint learning system outperforms previous models in terms of further developed execution. The creative aspect of this study is how it integrates a deeper comprehension of sentence structure, so making a significant contribution to the development of increasingly sophisticated NLP frameworks.

Ibrahim, H. et al. (2021)provide a thorough overview of informatics-based techniques for pharmacovigilance drug interaction (DDI) signal detection. In order to work on the recognizable proof of expected DDIs, the survey focuses on how computerized reasoning (man-made intelligence) and AI calculations are applied to various information sources, such as electronic health records (EHRs), online entertainment, and unconstrained announcing frameworks. The study discusses the challenges that traditional pharmacovigilance approaches face, such as time delays and underreporting, and how artificial intelligence-based frameworks can overcome these limitations by providing real-time analysis of massive datasets. The authors discuss a variety of models that have been used to dissect medication interactions, such as information mining techniques and profound learning models. They assume that although methods based on artificial intelligence provide encouraging results in terms of improving signal recognition, more improvements are needed in areas such as computation simplicity, approval, and information sharing.

Kim, S. et al. (2015)examine the use of a straight part strategy based on rich elements to extract drug interactions (DDIs) from biomedical text. Because clinical writing is a huge and evolving metric, the review emphasizes the importance of mechanizing the extraction of DDIs. To construct a direct part-based AI model, the authors provide a method that makes use of a variety of components, including as lexical, syntactic, and semantic data. They evaluate their methods using a benchmark dataset and get significant results when compared to other frameworks. The work advances the field of biomedical informatics by presenting a robust, adaptable approach to writing-based DDI extraction. The significance of this technique lies in its ability to enhance the programmed recognition of possible drug interactions. This can aid in promoting tolerable welfare and facilitating healthcare navigation.

Lai et al. (2015)examine ways to generate high-quality word embeddings for tasks related to natural language processing. The main focus of the work is on the processes that affect the feasibility of word embeddings, such as the choice of preparation computations, the size of the preparation corpus, and the type of word representations. The authors consider popular models such as Continuous Bag of Words (CBOW) and Skip-Gram and explore the role of setting in creating semantically rich embeddings. The paper provides tidbits of information on how to enhance the syntactic and semantic properties of embeddings by dissecting different aspects of word embedding age. Word depictions have a fundamental role in the development of tasks like word closeness, relationship, and report order, which are enhanced by the review.

Mikolov, T. et al. (2013)examine the linguistic characteristics of word representations in continuous space and show how various linguistic regularities are captured by these embeddings. In word embeddings, the concept of vector arithmetic is presented in the paper. This allows one to deduce word relationships, such as analogies (e.g., "king" - "man" + "woman" = "queen"), using elementary mathematical operations.

Word embeddings have been developed thanks in large part to this study, which has numerous applications in natural language interpretation. The study assesses word embeddings through tasks such as syntactic and semantic regularity detection and word similitude, demonstrating the ability of continuous word space models to represent intricate language patterns. The review has become a mainstay of contemporary NLP and has a profound impact on the creation of vector-based language models.

MATERIALS AND METHODS

This section first presents a one-stage method (SCNN1) based on support vector networks (SCNs) that classifies each candidate instance into one of the five DDI kinds (ADVICE, EFFECT, INT, MECHANISM, and NEGATIVE). Since two-stage approaches typically perform better than one-stage methods, a two-stage method based on SVM is then introduced, called CNN2. As shown in Figure 1, our one-stage technique SCNN1 consists of six processing steps:

1. A negative instance filtering process that eliminates potential negative occurrences in order to rebalance the class distribution of the dataset.
2. A pre-processing stage that creates a corpus that is simple to comprehend for classifiers.
3. The learning word embedding step uses word2vec and the Enju parser to construct the syntax word embedding.
4. In the fourth stage of feature extraction, both conventional and convolutional features are extracted.
5. Educating the classifier5 steps that use the collected characteristics to train a five-class classifier.
6. At the DDI detection and classification stage, a convolutional neural network (CNN) model is used to classify each test set instance into one of the five DDI classes.

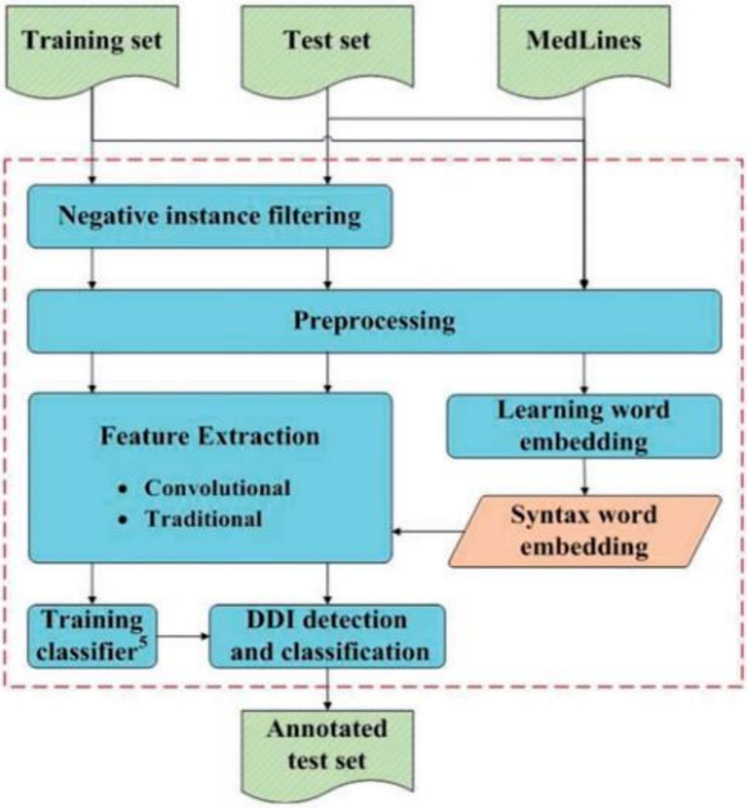


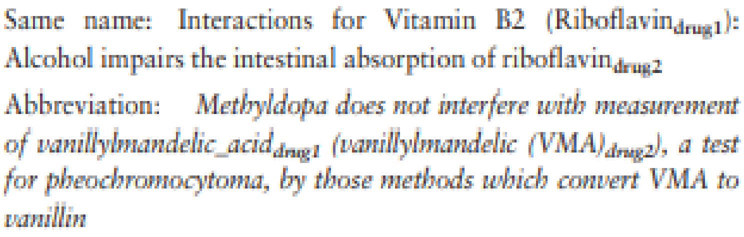
Fig. 1. Our one-stage method's processing flow SCN1.

The following sections provide descriptions of the specifics.

Negative instance filtering

Material characterization with an uneven distribution of classes has encountered a significant obstacle due to the fact that the majority of classifier learning algorithms presume a balanced distribution of classes and equal costs for misclassification. Also, there's a problem with the distribution of classes in the DDI Extraction 2015 challenge; for instance, there are 1.59 positive instances for every 5.91 negative ones in the training set. In order to mitigate this issue, we create a corpus that is less unbalanced by eliminating potential negative cases using the subsequent two rules:

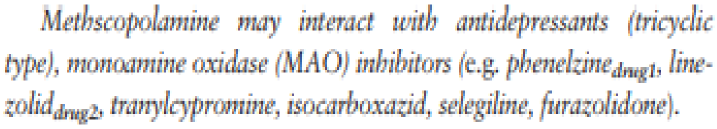
Rule 1: Every drug has a very low probability of interfering with itself, thus we rule out instances where two possible pharmaceuticals have the same name. To be more specific, the following two instances are scrutinized: There are two pharmaceuticals with the same name, and the Supplementary Materials describe where to look for the abbreviation, thus one medicine is really just an abbreviation for another. These are two examples.



Same name: Interactions for Vitamin B2 (Riboflavin$_{drug1}$): Alcohol impairs the intestinal absorption of riboflavin$_{drug2}$

Abbreviation: *Methyldopa does not interfere with measurement of vanillylmandelic_acid$_{drug1}$ (vanillylmandelic (VMA)$_{drug2}$), a test for pheochromocytoma, by those methods which convert VMA to vanillin*

Rule 2: False positives are common, therefore cases where two candidate medications are in coordinate relations are taken out. For instance, rule 2 will eliminate the following case.



Methscopolamine may interact with antidepressants (tricyclic type), monoamine oxidase (MAO) inhibitors (e.g. phenelzine$_{drug1}$, linezolid$_{drug2}$, tranylcypromine, isocarboxazid, selegiline, furazolidone).

Pre-processing

A proper pre-processing can greatly improve the previous performance. Two pre-processing steps are taken in our method: tokenization and converting the numbers into two uniform structures.

Syntax word embedding

In sentence level relation grouping difficulties, syntactic data plays a crucial role. As a result, deep learning techniques also use it to address relation arrangement difficulties. Xu suggested a relation order model based on CNN that makes use of data on the shortest dependency paths. The contribution is not based on the original phrase order but on the word sequence in the shortest path order. Yan found the shortest dependency path solution to the relation characterization problem using extended momentary memory networks. The generated ordered input sequence is based on the data from this solution. This approach uses syntactic data to construct a new ordered input sequence rather than train the word embedding.

A word embedding is a parameterized function that converts words into vectors with high dimensions. Bengio initially proposed word embedding as a way to combat the dimensionality problem when developing language models with neural networks. Thusly, many word embeddings have been recommended for language model learning. Besides, word embeddings were likewise much of the time applied in numerous NLP projects. Utilizing CNN and word implanting as the information, Collobert had the option to accomplish cutting edge execution in POS labeling, piecing, Named Substance Acknowledgment, and Semantic Role Labeling (SRL). To address the connection characterisation issue, Zeng proposed utilizing a convolutional brain network in view of word implanting.

Nevertheless, the aforementioned word embeddings are entirely dependent on linear contexts, or the surrounding words arranged in a sentence's linear sequence. They disregard the syntactic information, which is crucial in sentence-level order issues like DDI extraction.

The dependency-based syntactic contexts for word embedding learning were proposed by Levy and Goldberg. Hashimoto employed predicate-argument structural settings to teach him the word embedding, which he then utilized to gauge the semantic similarity of short sentences. According to their techniques, the training process introduces syntactic data by building syntactic contexts rather than regular linear contexts, which are the surrounding words of a phrase arranged linearly. In contrast to the latter, the syntactic context produces embeddings that are more inclusive and targeted.

Our methodology changes the word2vec gadget's result to present sentence structure word installing, a clever method for putting away syntactic information. This implies that we feed word groupings that are the briefest in the predicate-contention structure, as opposed to the underlying straight request word arrangements. Trimethoprim might keep phenytoin from being processed by the liver. The Enju parser makes the predicate-contention design of the sentence during the underlying parsing of the sentence, as represented in Figure 2. The first and last keywords in the phrase "Trimethoprim impede metabolism of phenytoin" are combined as quickly as feasible to produce the remaining words. Removing superfluous auxiliary words like "may,""the," and "hepatic" leaves only the main parts of the phrase intact, which are necessary for explaining its grammatical structure. This reduces the length of the syntax's word order. Next, word2vec is used to create the syntactic word embedding by entering these shortest way order sequences. The syntax word embedding is represented by the word E word 2 Rmn, where n is the syntax word embedding's dimension size and m is the jargon's size.

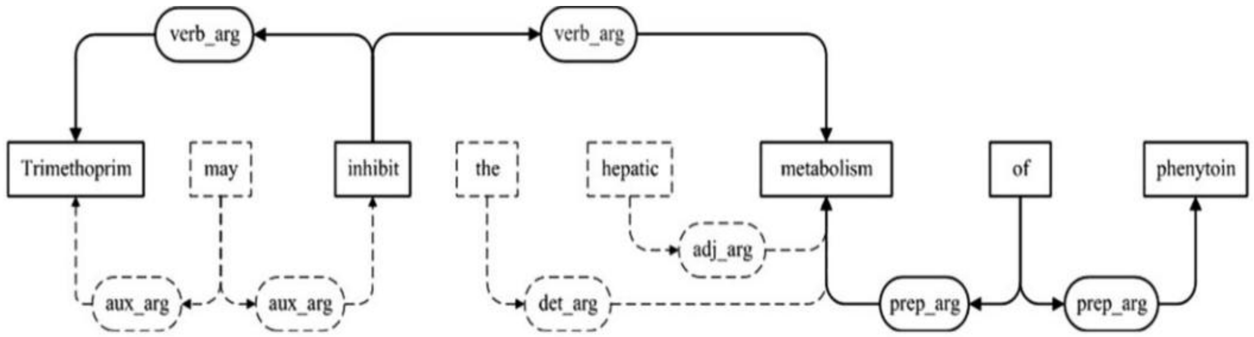


Fig. 2. The sample sentence's predicate-argument structure

Our syntactic context word embedding differs from earlier ones in that it is derived solely from the condensed syntax word sequence, maintaining the syntactic structure of the sentence while eliminating its less significant components. We tested it for DDI extraction and found that, despite its simplicity, it performed effectively.

Feature set

Collobert proposed an architecture based on convolutional neural networks (CNNs) to manage SRL-like tasks at the phrase level. Then, by applying this approach, Zeng addressed issues related to relation grouping. This paper likewise presents a CNN-based method for DDI extraction from the logical writing. In this methodology, the conventional sack of-words highlights (meager 0-1 vectors) are joined with implanting based convolutional highlights (thick genuine worth vectors). The auto-encoder is then used to lessen the distinction between them. We can blend two unique sorts of qualities all the more effectively along these lines.

Classifier training

Once the convolutional and traditional features are combined, the outcome is OutF½½ConvF; TradF2 Rn8, where n8¼n3þn7. What follows is the transmission of Out F to the output layer.:

$$\mathbf{out} = \mathbf{OutF.W_{out}} \qquad (1)$$

The size of the outcome layer (Wout 2 Rn8n9) in the DDI characterisation issue is equivalent to the quantity of DDI sorts (n9). The outcome can be communicated as outµ [out1, out2..., outi..., outn9], where outi is the certainty score of the related DDI type I.

The parameters of the model can be expressed using the equation (h ¼ (M, Wconv, Wtrad, Wout)). Equation (2) is used to conduct the SoftMax operation across all DDI types and acquire the likelihood value for each DDI type.:

$$\mathbf{p(i|x,\theta)} = \frac{\mathbf{e^{out_i}}}{\sum_{j=1}^{n_9} \mathbf{e^{out_i}}} \qquad (2)$$

Equation (3) is then used to determine the log probability of the parameters when all training cases (T ¼ {(x(i), y(i))}) are provided:

$$\mathbf{J(\theta)} = \sum_{\mathbf{i}} \mathbf{log}\left(\mathbf{p}\left(\mathbf{y^{(i)}|x^{(i)},\theta}\right)\right) \qquad (3)$$

Our approach to maximize the log probability makes use of the stochastic gradient descent methodology, just like in Zeng's work.

Two-stage method

Our two-stage SCNN-based technique (SCNN2) comprises the following processing steps, as seen in Figure 1 of the Supplementary Materials:

1. A negative instance filtering process that eliminates potential negative occurrences in order to rebalance the class distribution of the dataset.
2. A pre-processing stage that creates a corpus that is simple to comprehend for classifiers.
3. The learning word embedding step uses word2vec and the Enju parser to construct the syntax word embedding.
4. The fourth step of feature extraction involves extracting the convolutional and conventional features.
5. Educating the classifierUsing the features that were retrieved, a binary classifier is trained in two steps.

6. The DDI detection phase uses the classifier to identify the DDIs in the test set.
7. Educating the classifierUsing the retrieved features as a basis, a four-class classifier is trained in four steps.
8. The DDI classification stage uses the classifier to divide the recovered DDIs from step 6 into four distinct DDI kinds.

Our technique parts the DDI extraction process into two phases: DDI discovery and DDI gathering, very much like existing two-stage strategies. Most quite, our two-stage technique requires preparing two SCNN classifiers with result layer sizes of two and four, individually, while our one-stage strategy just requires preparing one SCNN classifier with an outcome layer size of five.

EXPERIMENTAL RESULTS AND DISCUSSIONS

Experimental settings

Numbapro, a Python compiler from Continuum Investigation, is used to train and develop our SCNN model. It can compile Python code for multicore Central Processing Units or GPUs with CUDA support. It takes several hours to prepare our model using an Nvidia Tesla k20 Graphics Processing Unit (GPU). To generate the syntactic data, the enormous amount of texts must be parsed using the Enju parser, which will take almost a month for the syntax word embedding learning process. The measures of the DDI corpus are shown in Table 1, which compares the proportion of positive to negative instances, all out instances, and the number of positive and negative examples in the original and new training and test sets. There were 23,774 negative and 4,022 positive examples in the initial training set. This resulted in a total of 27,794 examples and a ratio of 1 positive to 5.8 negative, showing a substantial imbalance. In contrast, the new training set has a more equal percentage of 1:2.4 and fewer examples—3,842 positives and 8,991 negatives—for a total of 12,831 occurrences. Regarding the test sets, the initial test set has 5,763 cases with a 1:4.8 proportion, consisting of 981 positive and 4,784 negative occurrences. It's interesting to note that the new test set has 3,057 examples with an improved percentage of 1:2.3, consisting of 973 positive and 2,086 negative instances. This shows that the new training and test sets have a more balanced distribution of positive and negative occurrences, which could improve the model's performance by reducing the class imbalance in the original sets. The measurements of the DDI corpus before and after the negative instance filtering procedure are displayed in Table 1.

Table 1: The DDI corpus's statistics

Corpus	Positives	Negatives	Total	Ratio
Original Training Set	4022	23,774	27,794	1:5.8
New Training Set	3842	8991	12,831	1:2.4
Original Test Set	981	4784	5763	1:4.8
New Test Set	973	2086	3057	1:2.3

The decent F-score measure is utilized by the current DDI extraction strategies to evaluate execution. This metric's equation is F-score ¼ (2PR)/(P ÷ R), where P represents exactness and R for review. We likewise utilize the F-score to think about the exhibition in contrast to these methods.

Performance comparison with other methods

Table 2 shows the performance comparison between our method and others. The two-stage procedures typically perform better than the one-stage methods, as should be evident from it. For instance, the two-stage approaches are used to obtain the top two outcomes in the DDI Extraction Challenge (FBK-irst and WBI). Kim used a two-step process as well later on to get even greater results. It may defy logic to assume that the two-step approach cannot outperform the one-stage method given its flaw of failures in the DDI detection stage transferring to the DDI grouping stage.

Table 2: Comparing performance with the DDI Extraction 2015 test set

Method		Precision	Recall	F-Score	PRF-score (%)	Precision	Recall	F-Score
One-stage	SCNN1	0.693	0.653	0.672	8.8%	0.749	0.770	0.759
	UTurku	0.734	0.501	0.596		0.860	0.587	0.698
	NIL_UCM	0.537	0.503	0.519		0.610	0.571	0.590
Two-stage	SCNN2	0.727	0.653	0.688	2.8%	0.777	0.771	0.774
		–	–	0.672	–	–	–	0.777
	FBK-irst	0.648	0.658	0.653		0.796	0.808	0.802
	WBI	0.644	0.581	0.611		0.803	0.724	0.761

Table 2 presents a performance correlation analysis of various approaches using the DDI Extraction 2015 test set, with particular emphasis on F-score, precision, and recall metrics. One-stage and two-stage approaches are contrasted. With a PRF-score improvement of 8.8%, the SCNN1 model in the one-stage technique achieves a second precision/recall/F-score of 0.749/0.770/0.759. Its precision is 0.693, recall is 0.653, and F-score is 0.672. The UT Urku technique performs better in precision in the second half, with 0.860 precision and 0.698 F-score, but has a lower recall of 0.501, translating to a moderate F-score of 0.596. With an F-score of 0.519, NIL_UCM performs the worst, displaying just modest increases in the second batch. SCNN2 performs better than other two-stage approaches, starting with an F-score of 0.688 and increasing to 0.774. With an F-score of 0.653 in the first measure and a higher 0.802 in the second, FBK-irst demonstrates consistent performance. Despite having a poorer initial recall, WBI obtains a second F-score of 0.761, demonstrating the effectiveness of two-stage approaches in DDI extraction tasks, particularly SCNN2 and FBK-irst.

The characteristics and tactics' impact on performance

Furthermore, we do tests with SCNN2 to assess the performance of the features and techniques of our approach. The findings are displayed in Table 3: we compute the F-score and the equivalent decrease from the prior removal for each feature or technique that was eliminated.

Table 3: The characteristics and tactics' impact on performance

Strategy or Feature Removed	Precision	Recall	F-Score	D (%)
None	0.727	0.653	0.688	–
Negative Instance Filtering	0.687	0.612	0.647	5.2%
Syntax	0.713	0.601	0.652	4.7%
POS	0.709	0.625	0.664	3.5%
POS Encoding	0.692	0.654	0.672	2.7%
Shortest Path	0.673	0.588	0.628	7.1%
Shortest Path Encoding	0.663	0.618	0.640	5.9%
Position	0.682	0.638	0.659	3.0%
Word Embedding	0.641	0.574	0.606	9.3%
Context	0.659	0.601	0.629	6.0%
Convolution Layer 1	0.613	0.578	0.594	9.5%
Convolution Layer 2	0.579	0.650	0.613	8.6%

Table 3 looks at how various characteristics and methods affect the performance of the model by calculating the percentage decrease (D) and evaluating precision, recall, F-score, and each strategy's impact on the model. The model performs best when no strategy or feature is eliminated, with a precision of 0.727, recall of 0.653, and F-score of 0.688.

Performance declines to varied degrees when certain components are removed. The removal of word embeddings, for example, causes the largest performance decline, with a 9.3% drop in the F-score, decreasing recall to 0.574 and precision to 0.641. In essence, there is a noticeable 9.5% decline when the first convolutional layer is removed.

Significant consequences are also shown by other important aspects, such as syntax, shortest route encoding, and negative instance filtering, which show 5.2%, 4.7%, and 5.9% F-score decreases, respectively. A 2.7% decrease in F-score results from the elimination of POS (grammatical form) encoding, which is less effective but still substantial. This table illustrates the fundamental function that each feature—word embeddings, convolution layers, and the shortest path, in particular—plays in reaching optimal model performance.

CONCLUSION

In contemporary healthcare, the use of biomedical informatics techniques to detect drug interactions (DDIs) is a game-changing approach. A unique word embedding is used in a SCNN-based DDI extraction strategy to utilize sentence syntactic data. To introduce location and POS data, position and POS capabilities are introduced to the syntactic word embedding. Prior to combining them with convolutional highlights, autoencoders are utilized to change over meager pack of-words include vectors into thick genuine worth element vectors.

The mix is fed through a SoftMax in order to train the DDI classifier. In experiments, the approach outperforms state-of-the-art methods with an F-score of 0.688. One of the principal objectives of the work is to utilize auto-encoder to change over scanty pack of-words highlights into thick genuine worth component vectors. Learning the syntax word embedding from brief syntactic word sequences and enhancing it with POS features is another objective. Multiclass problems are better addressed by SCNN than by other methods that use multiple concurrent SVMs to build classifiers.

REFERENCES

[1]. Ayvaz, S., Horn, J., Hassanzadeh, O., Zhu, Q., Stan, J., Tatonetti, N. P., ... & Boyce, R. D. (2015). Toward a complete dataset of drug–drug interaction information from publicly available sources. Journal of biomedical informatics, 55, 206-217.

[2]. Bengio,Y. et al. (2013) Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell., 35, 1798–1828.

[3]. Chowdhury, M.F.M. and Lavelli, A. (2013) FBK-irst: A multi-phase kernel-based approach for drug-drug interaction detection and classification that exploits linguistic information. The North American Chapter of the Association for Computational Linguistics, Atlanta, GA, USA, p.53.

[4]. Hashimoto, K. et al. (2014) Jointly learning word representations and composition functions using predicate-argument structures. EMNLP, 14, 1544–1555.

[5]. Ibrahim, H., Abdo, A., El Kerdawy, A. M., & Eldin, A. S. (2021). Signal detection in pharmacovigilance: a review of informatics-driven approaches for the discovery of drug-drug interaction signals in different data sources. Artificial intelligence in the life sciences, 1, 100005.

[6]. Kim, S. et al. (2015) Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. J. Biomed. Inform., 55, 23–30.

[7]. Lai, S., Liu, K., He, S., & Zhao, J. (2015). How to generate a good word embedding? Intelligent Systems IEEE, 1–1.

[8]. Mikolov, T. et al. (2013) Linguistic regularities in continuous space word representations. HLT-NAACL, 13, 746–751.

[9]. Percha, B., & Altman, R. B. (2013). Informatics confronts drug–drug interactions. Trends in pharmacological sciences, 34(3), 178-184.

[10]. Qiu, Y., Zhang, Y., Deng, Y., Liu, S., & Zhang, W. (2021). A comprehensive review of computational methods for drug-drug interaction detection. IEEE/ACM transactions on computational biology and bioinformatics, 19(4), 1968-1985.

[11]. Segura Bedmar, I. et al. (2014) Lessons learnt from the DDIExtraction-2013 shared task. J. Biomed. Inform., 51, 152–164.

[12]. Wan, L. et al. (2013) Regularization of neural networks using drop connect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), The North American Chapter of the Association for Computational Linguistics, Atlanta, GA, USA, pp. 1058–1066.

[13]. Xu, K. et al. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. In: Proceedings of Empirical Methods in Natural Language Processing, SIGDAT, Lisbon, Portugal, pp. 536–540.

[14]. Yan, X. et al. (2015). Classifying Relations via Long Short-Term Memory Networks Along Shortest Dependency Path. In: Proceedings of Empirical Methods in Natural Language Processing. SIGDAT, Lisbon, Portugal, pp. 1785–1794.

[15]. Zeng, D. et al. (2014) Relation classification via convolutional deep neural network. In: Proceedings of COLING, the Association for Computational Linguistics, Dublin, Ireland, pp. 2335–2344.