Drug-Drug Interaction Extraction-Based System: an NLP Approach

José Machado[1*†], Carla Rodrigues[1†], Regina Sousa[1†] and Luis Mendes Gomes[2†]

[1*]Centro Algoritmi/LASI, University of Minho, Portugal.
[2]Centro Algoritmi/LASI, University of the Azores, Portugal.

*Corresponding author(s). E-mail(s): jmac@di.uminho.pt;
Contributing authors: a84710@alunos.uminho.pt;
regina.sousa@algoritmi.uminho.pt; luis.mp.gomes@uac.pt;
†These authors contributed equally to this work.

Abstract

Purpose: Poly-medicated patients, especially those over 65, have increased. Multiple drug use and inappropriate prescribing increase drug-drug interactions, adverse drug reactions, morbidity, and mortality. This issue was addressed with several CDSS alerts. Health professionals have not followed these systems due to their poor alert quality and incomplete databases. **Methods:** Recent research shows a growing interest in using Text Mining via NLP to extract drug-drug interactions from unstructured data sources to support clinical prescribing decisions. NLP text mining and machine learning classifier training for drug relation extraction were used in this process. **Results:** In this context, the proposed solution allows to develop an extraction system for drug-drug interactions from unstructured data sources. The system produces structured information, which can be inserted into a database that contains information acquired from three different data sources. **Conclusion:** The architecture outlined for the drug-drug interaction extraction system is capable of receiving unstructured text, identifying drug entities sentence by sentence, and determining whether or not there are interactions between them.

Keywords: Drug-Drug Interactions; Information Extraction; Natural Language Processing; Text Mining

1 Contextualization and Motivation

A drug-drug interaction (DDI) is a change in the impact of a drug due to the presence of another medication, lowering or neutralizing a drug's efficacy, increasing toxicity, or even causing death[1]. Drug-drug interactions can be classified by several criteria, which are based on the underlying mechanism (for example type of interaction or interaction severity). They can be classified as Pharmacokinetic or Pharmacodynamic [2]. Concerning severity, interactions can be classified as minor, moderate, or severe/major, depending on their impact on the body. In the minor case, they have minimal risk so, they will often have no clinical relevance. In the moderate case, there must be increased vigilance or a reduction in the dose may be necessary. In the severe/major case, a change in therapy is mandatory because it's a case of high clinical significance that will probably lead to serious clinical outcomes [3].

In high-income countries, 40-50% of the elderly are poly-medicated taking an average of seven drugs a day [4, 5]. The concomitant consumption of several drugs and inappropriate prescription of medications potentiates the existence of Adverse Drug Reactions (ADRs), and drug-drug interactions, resulting in a high morbidity and mortality rate [6, 7]. According to the statistics from the US Centers for Disease Control and Prevention, approximately 300,000 people die of ADRs per year in the US and Europe [1, 8].

Thus, expanding our knowledge of interactions between drugs is important to reduce public health safety incidents. Detecting DDIs has become an essential part of public health safety policy because, with rich DDI knowledge, patients can be prevented from harmful drug-drug interactions [1, 9]. However, a large amount of the valuable information on DDIs is unstructured, written in natural language, and hidden in biomedical literature [9, 10]. Hence, text mining technologies that can automatically extract DDIs from unstructured content are necessary to combat text information overload and the process of converting text into machine-understandable knowledge [11–13]. Moreover, in order to keep up with the expanding body of knowledge surrounding drug-drug interactions, the development of automatic information extraction methods is essential, because manual extraction is time-consuming and can lead to outdated data. Recommendation systems have become an integral part of numerous online platforms because they improve user engagement and enrich the user experience [14].

The main goal of this work is to use text mining to develop an intelligent drug-drug extraction system based on pharmacological and toxicological data.

2 DDI Extraction

2.1 Main Approaches

Text mining can be efficient solutions for analyzing biomedical corpora, such as scientific articles, clinical records, and public databases, to support pharmacovigilance and to reduce the time spent by healthcare professionals [15, 16].

Therefore, the use of text mining techniques to extract DDIs from biomedical corpora and help construct drug databases has received attention [1, 17]. DDI extraction is the task of identifying drug-named entities and extracting the potential interaction relations between drug entity pairs from text. It is a special case of binary relation extraction where (subject, predicate, object) triples are extracted from natural language text. Both the subject and object are pharmacological substances, and the predicate is the type of interaction, [18, 19]. Several methods for the extraction of DDIs have been proposed based either on a single task, Drug Named Entity Recognition (DNER) or DDI classification, or DDI extraction in an end-to-end approach [18].

DNER classification approaches aim to recognize drug entity mentions and classify them into predefined categories. For example, [17] presented a feature-based method that identifies and classifies drugs into four classes, achieving an F-score of 57%, and [20] constructed a bi-directional Conditional Random Field and Long Short-Term Memory architecture achieving an F-score of 79.26%, both when evaluated on the DDI corpus. [21], formulated the problem for DNER as a machine reading comprehension problem, achieving state-of-the-art performance on the CHEMDNER corpus with an F-score of 92.92%.

DDI classification approaches focus merely on the task of classifying the relation of drug pairs in biomedical texts. For that, datasets are used, where each entity pair is labeled with the predefined relation types. The types *advice*, *mechanism*, *effect*, åand *int* denote the types of interactions between two drugs and correspond to the positive class. The type *false* indicates the absence of interaction and corresponds to the negative class [18]. For the end-to-end DDI extraction task, existing methods can be divided into two categories: one-stage and two-stage. In one-stage, a multiclass classifier is built to directly classify each candidate DDI instance into one of the four types. In other words, the one-step technique detects and classifies DDIs concurrently. The two-stage split the problem into two steps: first, a binary classifier is built to recognize all candidate instances into positive or negative instances, and then only the positive instances are considered to be classified into one of the four predefined DDI types of the positive class [22, 23].

Several DDI extraction methods have been proposed in scholarly papers, with the earliest methods essentially dividing into two groups: pattern-based methods and machine learning-based methods. With the improvement in computing power and the availability of large datasets, deep learning has emerged as a promising approach and has become a dominant method for DDI extraction tasks [24, 25]. Existing deep learning models for DDI extraction are all based on supervised approaches and although deep learning has many basic architectures, the task of DDI extraction from the literature has mainly used Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [8].

Some works are only focused on the DDI extraction task where a binary classifier is built to recognize all candidate instances into positive or negative instances. That is, the model detects the expressions that indicate interactions

between the mentioned drugs classifying each sentence as interacting or non-interacting. For example, [26] proposed a DDI extraction model with a C4.5 decision tree classifier achieving an F-score of 76.9% when evaluated on the DDI corpus. Also, [27] trained an SVM classifier for the extraction of DDIs achieving an F-score of 83.48% and 59.17% when testing only on the DDI corpus's, DrugBank and Medline documents, respectively.

Alternatively BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) which is a domain-specific language representation model for biomedical text mining pre-trained has been used on large-scale biomedical corpora. With almost the same architecture across tasks, BioBERT largely outperforms BERT (Bidirectional Encoder Representations from Transformers) and previous workd in a variety of biomedical text mining tasks such as Named Entity Recognition (NER) and Relation Extraction (RE), which this domain-specific pre-trained model can be fine-tuned [28, 29]. For example, [30] approached the DDI extraction task using the BERT model achieving an F-score of 81.97% when evaluated on the DDI corpus. Also, [31] proposed a new model for extracting DDIs called Bio-ER-BERT, which combines the BioBERT model and the R-BERT model for relationship extraction, achieving an F-score of 83.88%.

2.2 Drug Knowledge Bases

DrugBank is a free-access online database containing information on drugs and their mechanisms, interactions, and targets [32]. In total about 500000 drugs are described. This database can identify the drug through the common name (USAN), the chemical name (IUPAC), the trademark name, or through the pharmacotherapeutic classifications (AHFS and ATC), among others [33]. The rich, high-quality, primary-sourced content found in DrugBank has allowed it to become one of the world's most widely used reference drug resources [32]. For non-commercial purposes, the DrugBank database is available free of charge in XML format.

DDInter is a DDI database that contains about 240,000 DDI associations connecting 2,051,833 approved drugs, each of them annotated with basic chemical and pharmacological information and their interaction network. For each DDI, the severity level (major, moderate, minor, and unknown), accepted as suggested by DRUGDEX, mechanism description, strategies for managing potential risks, and alternative medications based on the third level of the Anatomical Therapeutic Chemical (ATC) code, are provided [2, 34].

Infomed is a Portuguese online platform provided by INFARMED, with 37531 drudgs, whose data are similar to those presented in the national drug database but in a more accessible and user-friendly format for the user to search for the desired information about the drugs. The information available can be highlighted the therapeutic indications, contraindications, special warnings and precautions for use, drug interactions and other forms of interaction, use during pregnancy and lactation, and undesirable effects [].

3 DDI Extraction Architecture Proposal

This section aims to present an architecture, illustrated in figure 1, designed to be incorporated into DDI extraction-based systems.

The architecture has as input a text extracted from unstructured data (for example, document files) that is first splitted into sentences using the Natural Language Toolkit (NLTK) library's *sent_tokenize* function. Each sentence is given as input to the DNER model, to extract drug/chemical entities that appear in the sentence. Notice that one sentence can have more than two drug mentions, so it is necessary to take all the possible drug pair combinations.

Following the text transformation pipeline, the feature extraction step takes place, where all sentences are represented by a numerical vector as a result of the combination of the following features: 1/0 BOW representation with a 1-2-gram model, Word2Vec (CBOW), and custom features. Finally, each sentence, now represented as a numerical vector, can be given as input to the classifier model previously trained, classifying each sentence as *interaction* or *no interaction*. At the end of the process, each sentence should be associated with the following structure information, *(pair of medical entities, classification result)*, and the newly extracted DDI information can be inserted into the database.

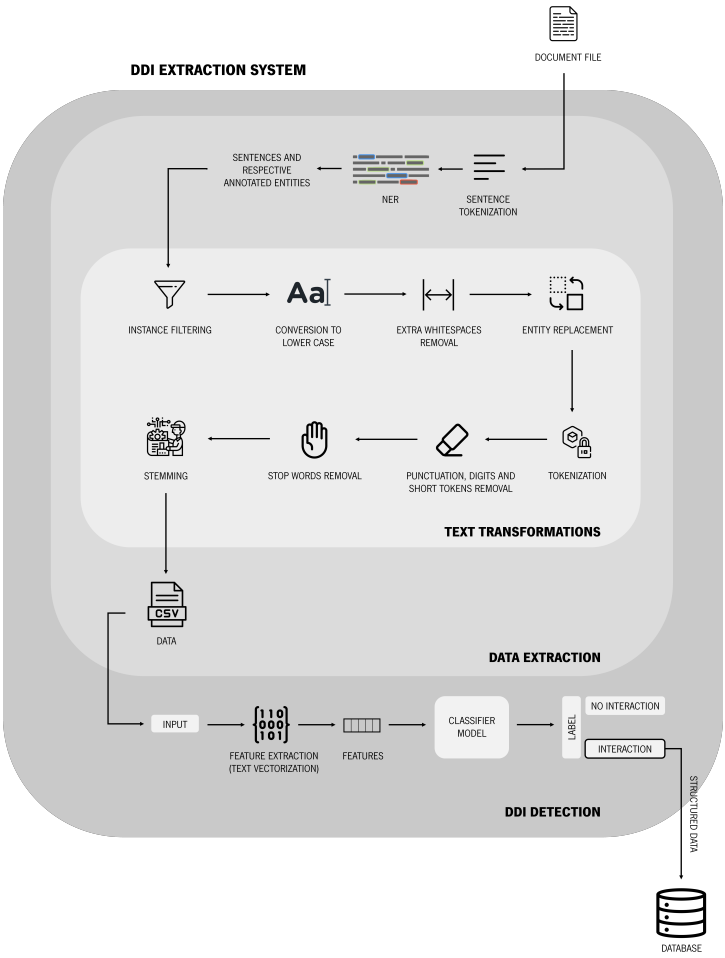

Fig. 1 Architecture proposal: DDI Extraction System.

In next section, the phases will be described in detail.

3.1 Data Extraction

This section describes the datasets used in the development of the DDI extraction system in terms of their main characteristics, the information presented, and statistics: the DDI and the CHEMDNER corpus.

NLP techniques rely mostly on the availability of annotated corpora so that is possible to train models that can be used to extract information from raw text [35]. Therefore, in [36] a manually annotated corpus of XML documents is presented, the DDI corpus, with pharmacological substances as well as the interactions between them. The corpus comprises 792 texts selected from the DrugBank database and 233 Medline abstracts, and was developed for the

SemEval DDIExtraction Challenge 2013, whose main goal was to provide a common framework for the evaluation of information extraction techniques applied to the recognition of pharmacological substances and the detection of DDIs from biomedical texts [36, 37]. Four entity types were proposed to annotate pharmacological substances: *drug*, *brand*, *group* and *drug_n*. As regards the relationships, five different types of DDI relationships are proposed [36]: *mechanism*, *effect*, *advice*, *int* and *false* where an interaction between the two drugs is not shown in the sentence. The DDI corpus comprises 905 XML documents with a total of 26,965 annotated drug relations for train, 3,194 of them corresponding to actual DDIs, including both pharmacokinetic and pharmacodynamic. For testing, the dataset has a total of 10,232 annotated drug relations, 5,495 of them representing an interaction.

When developing supervised NER systems, the availability of a large, manually annotated text corpus can be appropriated. The CHEMDNER corpus is a public available collection of 10,000 PubMed abstracts that contain a total of 84,355 chemical entity mentions corresponding to 19,805 unique chemical name strings. Each of the chemical entity mentions was manually labelled by an expert chemistry literature curators according to its structure-associated chemical entity mention class: abbreviation, family, formula, identifier, multiple, systematic, and trivial [38].

3.2 DDI Extraction System

The main goal of this study is the creation of a system that can extract drug-drug interactions from unstructured text and insert them into a database, as illustrated in Figure 1. In short, the development of the DDI extraction system mainly comprised the:

1. Development of a relational database on drugs and their interactions;
2. Training of a Drug Named Entity Recognition model for the identification of drug entities;
3. Training of a Machine Learning Classifier for Relation Extraction between two drug entities.

A relation database on drugs and their interactions was created as a result of the integration of three different sources of information: DrugBank, DDInter and Infomed. This database contains information about ATC classification, pharmacological activity, synonyms, chemical identifiers, external identifiers from other databases, and interactions with other active substances. The database aggregates data on 4,031 active substances, 7,722 medicines, and 1,341,086 drug-drug interactions, with 17,850 classified as major, 53,072 as moderate, 2,714 as minor, and 710,423 as unknown.

To train a DNER model for the identification of drug/chemical entities in plain text, the chosen approach was to fine-tune a BioBERT model on the CHEMDNER corpus. For that, was used the pre-trained BioBERT model by Data Mining and Information Systems (DMIS) Laboratory (Korea University)

from the Hugging Face Transformers library as the base and the Simple Transformers library on top of it to make it possible to train the NER model with just a few lines of code. The dataset used includes train, test, and validation sets with 30,682, 26,364, and 30,639 sentences, respectively, annotated at the token level in the CoNLL IOB-type format.

The task of identifying relationships between entities from unstructured text is known as Relation Extraction, part of the larger task of Information Extraction. Formally, the task receives an unstructured text and a set of entities as input and returns a set of triplets, each triplet taking the following form: (First Entity, Second Entity, Relation Type).

In this approach, the focus is only on the DDI detection task, reducing it to a binary classification problem of determining whether or not two drugs present in a sentence are related. That is, train a machine learning classifier to identify and distinguish whether the candidate entity pair has a semantic relation, classifying each sentence as *interaction* (label 1) or *no interaction* (label 0). It consists of a supervised training process on the labelled dataset, the DDI corpus, already divided into train and test sets.

First, it was necessary to perform some cleaning transformations on the DDI Corpus, such as (i) eliminating sentences that have discontinuous entities and corresponding pairs; (ii) eliminating sentences that do not have labelled entities; (iii) deleting empty XML documents; (iv) separate sentences labelled as *interaction* from sentences labelled as *no interaction*.

After running the text transformation pipeline, all the sentences and respective labels are saved in a CSV file containing a total of 19,695 sentences for training and 3,915 sentences for testing, which corresponds, to approximately, 67% of the data going to training and 33% going to testing. The test data has 682 sentences labelled as *interaction* (class 1) and 3,233 sentences labelled as *no interaction* (class 0). In the training data, 2,845 sentences are labelled as *interaction* and 16,850 labelled as *no interaction*. Being an unbalanced dataset and the class *no interaction* the majority one, the training set underwent the Random Undersampling process.

The feature extraction step comes next, in which a set of features is computed for each sentence. The feature vectors are then used to train a machine learning classifier. Eight different types of text vectorization techniques were applied: 1/0 BOW representation with a 1-gram model, 1/0 BOW representation with a 2-gram model, 1/0 BOW representation with a 3-gram model, 1/0 BOW representation with a 1-2-gram model, TF-IDF, custom features, Word2Vec and Doc2Vec.

The *ContVectorizer* function from scikit-learn was used for the 1-gram, 2-gram, 3-gram, and 1-2-gram representations, with the *ngram-range* parameter set to (1,1), (2,2), (3,3), and (1,2), respectively. The *binary* parameter was set to True for all n-gram representations, while the other parameters were left at their default values. As a result, each sentence was represented by a 3343, 18225, 33977, and 21568-dimensional vector for the 1-gram, 2-gram, 3-gram,

and 1-2-gram representations, respectively. For the TF-IDF text vectorization technique, was used the scikit-learn TF-IDF *vectorizer* implementation, resulting in a 3343-dimensional vector representation.

In terms of word embeddings-based features, the Gensim library's Word2Vec (CBOW) and Doc2Vec (PV-DM) implementations were used by proceeding with the generation of aggregated sentence vectors based on the averaging of the word vectors for the sentences' words. Each sentence was thus represented by a 10-dimensional vector that can be easily fed into the classifier.

Custom features were defined using domain knowledge, where a representative set of features is computed for each sentence about a drug pair. An 11-dimensional feature vector was created for each sentence by combining the following features:

F1 - number of positive keywords in the sentence;
F2 - positive keywords exist between drug names;
F3 - positive keywords exist within scope but not between names;
F4 - number of negative keywords in sentence;
F5 - negative keywords exist between drug names;
F6 - negative keywords exist within scope but not between drug names;
F7 - number of special keywords in the sentence;
F8 - number of words between drugs;
F9 - number of verbs between drugs;
F10 - drug keywords exist between drug names;
F11 - drug keywords exist within scope but not between names.

Features 1 to 9 were implemented by [26] in their work on extracting information about drug-drug interactions from biomedical literature. Features 10 and 11 were added to encompass sentences containing more than two drug entities. To extract those features was used the trigger list of positive (interact, inhibit, not recommended simultaneously, contradict, influence, affect, have effect, increase, enhance, decrease, diminish, carefully monitored, examined, effect, effects, initiated, with), negative (no, not, without, neither, nor, lack, cannot, absence, unchanged, unlikely), and special (concomitant, concomitantly, concurrently, simultaneously, co-administration) tokens by [26]. Positive and negative keywords refer to the most common words used to describe two interacting and non-interacting drugs, respectively. By increasing the meaning of positive keywords, the presence of special keywords in a sentence increases the likelihood that it will present an interacting drug pair.

Once the text transformations and feature extraction are completed, the next step is the selection and evaluation of the classification model, available on Scikit Learning. For that, different combinations of features and classifiers (Logistic Regression, Support Vector Machines, Decision Trees, Random Forest, Naïve Bayes, and K-Nearest Neighbor) were tested to determine the best model by comparing the values of the F1-score.

4 Results and Discussion

Several features sets are tested, namely: (set 1) 1-gram; (set 2) 2-gram; (set 3) 3-gram; (set 4) 1-2-gram; (set 5) tf-idf; (set 6) custom features; (set 7) word2vec; (set 8) doc2vec; (set 9) 1-2-gram + custom features; (set 10) 1-2-gram + tf-idf; (set 11) 1-2-gram + word2vec; (set 12) 1-2-gram + doc2vec; (set 13) 1-2-gram + word2vec + custom features; (set 14) 1-2-gram + word2vec + tf-idf; (set 15) 1-2-gram + doc2vec + custom features; (set 16) 1-2-gram + doc2vec + tf-idf.

The Precision, Recall, and F1-score values presented in the next tables correspond to the Weighted Average value, evaluating the overall performance of each model. The 1/0 BOW representation with a 1-gram, 2-gram, 3-gram, and 1-2-gram model, corresponding to feature sets 1, 2, 3, and 4, was the first vectorization technique tested. When comparing the performance of feature sets 1, 2, 3, and 4 (table 1), it can be seen that the 1-2-gram representation (set 4) produces better results in terms of precision, recall, and F1-score, although the difference is not very significant, especially when compared to the feature set 2.

Table 1: Comparing the performance of feature sets 1, 2, 3, and 4

Features	Algorithm	Precision	Recall	F1-score
bow_1_gram	Logistic Regression	0.814072	0.657216	0.698321
	SVM	0.810943	0.628097	0.672976
	Decision Trees	0.788407	0.528991	0.580785
	Random Forest	0.822825	0.675096	0.713797
	Naïve Bayes	0.795321	0.496041	0.545506
	K-Nearest Neighbor	0.777361	0.627842	0.672064
bow_2_gram	Logistic Regression	0.866947	0.763985	0.790186
	SVM	0.861220	0.744317	0.773607
	Decision Trees	0.835010	0.638059	0.681513
	Random Forest	0.851284	0.631418	0.674869
	Naïve Bayes	0.840125	0.662069	0.702861
	K-Nearest Neighbor	0.788430	0.791571	0.789963
bow_3_gram	Logistic Regression	0.839111	0.514943	0.559706
	SVM	0.840871	0.581354	0.627818
	Decision Trees	0.814360	0.778289	0.792300
	Random Forest	0.807536	0.785696	0.794988
	Naïve Bayes	0.845926	0.653384	0.695080
	K-Nearest Neighbor	0.749594	0.771137	0.759235
bow_1_2_gram	Logistic Regression	0.856963	0.785951	0.806675
	SVM	0.851810	0.758621	0.784441
	Decision Trees	0.840179	0.758621	0.783046
	Random Forest	0.863307	0.732056	0.763514
	Naïve Bayes	0.837579	0.666411	0.706666
	K-Nearest Neighbor	0.772656	0.740485	0.754326

Next, feature sets 5 and 6 (table 2) and 7 and 8 (table 3) were evaluated. Comparing their performance reveals that sets 5 and 6 produce superior results compared to the embeddings representations. Nonetheless, when compared to the 1/0 BOW representation, both of the four text vectorization techniques yield the worst results.

Table 2: Comparing the performance of feature sets 5 and 6

Features	**Algorithm**	**Precision**	**Recall**	**F1-score**
tf_idf	Logistic Regression	0.820748	0.646999	0.689604
	SVM	0.817280	0.638059	0.681749
	Decision Trees	0.785395	0.596679	0.645299
	Random Forest	0.808995	0.681992	0.718715
	Naïve Bayes	0.791310	0.495530	0.545542
	K-Nearest Neighbor	0.811956	0.482759	0.528369
custom_features	Logistic Regression	0.839291	0.754534	0.779735
	SVM	0.838158	0.754789	0.779785
	Decision Trees	0.835848	0.749936	0.775689
	Random Forest	0.836233	0.741507	0.769105
	Naïve Bayes	0.835837	0.755045	0.779666
	K-Nearest Neighbor	0.809527	0.793870	0.800770

Table 3: Comparing the performance of feature sets 7 and 8

Features	**Algorithm**	**Precision**	**Recall**	**F1-score**
word2vec	Logistic Regression	0.801749	0.627842	0.672754
	SVM	0.797009	0.606641	0.654028
	Decision Trees	0.776844	0.647254	0.687902
	Random Forest	0.803672	0.672797	0.710809
	Naïve Bayes	0.795198	0.568582	0.618973
	K-Nearest Neighbor	0.784931	0.540230	0.592347
doc2vec	Logistic Regression	0.795897	0.576501	0.626415
	SVM	0.793995	0.581609	0.631288
	Decision Trees	0.763246	0.595658	0.644329
	Random Forest	0.805750	0.696296	0.729891
	Naïve Bayes	0.797898	0.573180	0.623163
	K-Nearest Neighbor	0.787107	0.556833	0.608312

As a result, the 1-2-gram representation was combined with each of these features (tf-idf, custom, word2vec, and doc2vec). Combinations of different 1/0 Bow representations were not carried out to avoid obtaining vectors with excessively high dimensions. Looking at Table 4, some of the tested combinations outperformed the 1/0 BOW 1-2-gram representation (F1-score = 0.806675). The feature set 11 (1-2- gram + word2vec) with the Logistic Regression classifier produced a higher F1-score value of 0.814581.

Table 4: Some of the tested combinations

Features	Algorithm	Precision	Recall	F1-score
bow_1_2_gram + custom_features	Logistic Regression	0.869658	0.798723	0.818398
	SVM	0.865656	0.786207	0.807983
	Decision Trees	0.857392	0.785441	0.806335
	Random Forest	0.871101	0.777778	0.801772
	Naïve Bayes	0.837994	0.670243	0.710008
	K-Nearest Neighbor	0.828374	0.592593	0.639639
bow_1_2_gram + tf_idf	Logistic Regression	0.857807	0.787739	0.808189
	SVM	0.851068	0.762708	0.787618
	Decision Trees	0.826623	0.744061	0.769852
	Random Forest	0.841491	0.722095	0.753943
	Naïve Bayes	0.830293	0.672031	0.711424
	K-Nearest Neighbor	0.777274	0.656705	0.695470
bow_1_2_gram + word2vec	Logistic Regression	0.858114	0.795913	0.814581
	SVM	0.851736	0.768327	0.792149
	Decision Trees	0.821093	0.592593	0.640103
	Random Forest	0.854757	0.732567	0.763471
	Naïve Bayes	0.837841	0.667688	0.707782
	K-Nearest Neighbor	0.788252	0.625798	0.670768
bow_1_2_gram + doc2vec	Logistic Regression	0.858147	0.795147	0.813994
	SVM	0.852636	0.769349	0.793066
	Decision Trees	0.824667	0.694253	0.729883
	Random Forest	0.844646	0.704725	0.739688
	Naïve Bayes	0.837894	0.667944	0.708005
	K-Nearest Neighbor	0.802244	0.651341	0.692923

As the results obtained above (Table 4) are all relatively similar, combinations of three features were tested, with the 1-2-gram representation as constant. The results are shown in Tables 5 and 6.

Table 5: Combinations of three features

Features	Algorithm	Precision	Recall	F1-score
bow_1_2_gram + word2vec + custom_features	Logistic Regression	0.869411	0.809706	0.826990
	SVM	0.864706	0.782886	0.805229
	Decision Trees	0.852898	0.767050	0.791276
	Random Forest	0.867064	0.768327	0.793735
	Naïve Bayes	0.837994	0.670243	0.710008
	K-Nearest Neighbor	0.827696	0.586718	0.634061
bow_1_2_gram + word2vec + tf_idf	Logistic Regression	0.856483	0.794381	0.813145
	SVM	0.852392	0.774713	0.797260
	Decision Trees	0.805986	0.638059	0.681682
	Random Forest	0.846337	0.719796	0.752369
	Naïve Bayes	0.830406	0.672542	0.711865

Table 5 continued from previous page

Features	Algorithm	Precision	Recall	F1-score
	K-Nearest Neighbor	0.783563	0.597957	0.646460

Table 6: Combinations of three features

Features	Algorithm	Precision	Recall	F1-score
bow_1_2_gram + doc2vec + custom_features	Logistic Regression	0.868438	0.805875	0.823859
	SVM	0.864197	0.777778	0.801090
	Decision Trees	0.861002	0.782631	0.804590
	Random Forest	0.871101	0.777778	0.801772
	Naïve Bayes	0.837994	0.670243	0.710008
	K-Nearest Neighbor	0.826640	0.571648	0.619510
bow_1_2_gram + doc2vec + tf_idf	Logistic Regression	0.856849	0.794636	0.813400
	SVM	0.851789	0.772669	0.795577
	Decision Trees	0.808146	0.676117	0.713850
	Random Forest	0.845322	0.709323	0.743582
	Naïve Bayes	0.830406	0.672542	0.711865
	K-Nearest Neighbor	0.797973	0.619668	0.665580

The best result was obtained with the Logistic Regression classifier and the feature set 13 (1-2- gram + word2vec + custom features) achieving an F1-score of 0.826990 while ensuring acceptable values for precision and recall.

The main goal was to create a system capable of extracting drug-drug interaction information from unstructured data. To that end, a model was created that can read a sentence containing two or more drugs and determine whether or not two drugs interact by examining all of the possible drug pair combinations in the sentence. The proposed information extraction system implied the training of a DNER model for identifying drug/chemical entities and a machine learning classifier for relation extraction between two drug entities.

The resulting NER model can identify drug/chemical entities in plain text as B-Chemical and I-Chemical, achieving an F1-score, Precision and Recall of 89.32%, 88.53% and 90.13%, respectively.

Concerning the relation extraction model, it consists of an NLP and machine learning-based DDI detection system. For the positive class, the model achieved Precision, Recall, and F1-score of 45%, 80%, and 58%, respectively. That is, of all sentences classified as positive only 45% represented an actual drug-drug interaction, whereas 80% of all sentences representing a drug-drug interaction were successfully predicted. Concerning the overall performance of the classifier, the model achieved a Macro Average F1-score of 72% and a Weighted Average F1-score of 82%.

By analyzing the contribution of each feature, it was noticed that from all individual types of text vectorization techniques explored, the n-gram model approach was the one to achieve better results as this approach maintains

word order being able to capture the sentence's semantic meaning. The word embedding approaches, Word2Vec and Doc2Vec, did not perform as well as expected, although they have the capability of capturing semantic and syntactic relationships between words and also the context of words in a document, preserving most of the relevant information about a text corpus. This can be explained by the relatively small dataset size that was used to train the embedding model. Thus, it would be more appropriate to use pre-trained word embeddings like Google's Word2Vec and Stanford's Glove, as these are trained on large datasets, saved, and then used for solving other tasks.

When comparing the obtained results with other works focused only on the DDI detection task, the model achieved a close result, slightly lower than the F1-score of 76.9% achieved by [26]. However, the authors only used sentences with two drug mentions from the DDI corpus; thus, the dataset used for the experiment was not the same. Also, in [27] a work where an SVM classifier trained for the detection of DDIs achievied an F1-score of 83.48% and 59.17%, when testing only on the DDI corpus's DrugBank and MedLine documents, respectively. Thus, the dataset used by the authors for the experiment was also not the same, as in the context of this work was used the benchmark DDI corpus with its original division for train and test.

5 Conclusions and Future Work

The severity of polypharmacy and the development of new drugs, combined with a large amount of information on DDIs being unstructured, is the motivation for developing a solution that allows keeping an up-to-date database about drug-drug interactions. As a result, the information extraction system developed is intended to keep up with the growing volume of knowledge surrounding DDIs. The traditional manual extraction is time-consuming and can result in outdated information.

We propose a system that intents to receive unstructured text, identify in it, sentence by sentence, drug entities and then determine whether there are interactions between them. This system has two major limitations. The first one is in the input, it is only possible to introduce two drug names. Then, the classification model will only provide a binary output identifying if it exists interaction between the drugs, with an F1-score of 0.826990. This result was obtained with a logistic regression classifier which corresponds to the best precision and recall values (*cf.* Table 5). This process involved the application of text mining through NLP techniques and the training of a machine learning classifier for relation extraction between drugs. One of the challenges in developing the classification model was the low number of sentences labeled as *interaction*. Only about 30.5% of the annotated relations had an identified interaction. The system generates structured data in the form (drug1, drug2, label) that can be inserted into a database containing information acquired from three distinct data sources (DrugBank, DDinter and Infomed). Following

the DNER step, only the most recent occurrence of DRUG1 or DRUG2 is considered in sentences containing multiple references to these terms. Therefore, we assume that this system can be incorporated into a recommendation system that supports the Drugs Prescription Act in order to determine whether or not there is an interaction between the two drugs.

Other text vectorization techniques, such as pre-trained word embeddings, must be evaluated despite the model's promising performance. Other methods, including the application of deep learning models, specifically transformers models, may also be explored.

We are developing a recommendation system with the intention of using it to enhance a prescription software with the ability to identify interactions in real time, which can be useful during the medical act.

Acknowledgments

This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020

References

[1] Zhao, Z., Yang, Z., Luo, L., Lin, H., Wang, J.: Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. Bioinformatics **32**(22), 3444–3453 (2016)

[2] Xiong, G., Yang, Z., Yi, J., Wang, N., Wang, L., Zhu, H., Wu, C., Lu, A., Chen, X., Liu, S., *et al.*: Ddinter: an online drug–drug interaction database towards improving clinical decision-making and patient safety. Nucleic acids research **50**(D1), 1200–1207 (2022)

[3] Kaski, J.C., Kjeldsen, K.P.: The ESC Handbook on Cardiovascular Pharmacotherapy. European Society of Cardiology. Oxford University Press, ??? (2019)

[4] Gomes, D., Placido, A.I., Mó, R., Simões, J.L., Amaral, O., Fernandes, I., Lima, F., Morgado, M., Figueiras, A., Herdeiro, M.T., *et al.*: Daily medication management and adherence in the polymedicated elderly: A cross-sectional study in portugal. International Journal of Environmental Research and Public Health **17**(1), 200 (2020)

[5] Pinto, A., Rodrigues, T., Mendes, J., Bação, F., Lobo, V.: Medication and polymedication in portugal, 59–68 (2013)

[6] Sarmento, E., Leal, C., Monteiro, M.J.: Information technology in the process of managing polypharmacy in elderly patients. Procedia computer science **121**, 322–328 (2017)

[7] McIntosh, J., Alonso, A., MacLure, K., Stewart, D., Kempen, T., Mair, A., Castel-Branco, M., Codina, C., Fernandez-Llimos, F., Fleming, G., *et al.*: A case study of polypharmacy management in nine european countries: Implications for change management and implementation. PloS one **13**(4), 0195232 (2018)

[8] Zhang, T., Leng, J., Liu, Y.: Deep learning for drug–drug interaction extraction from the literature: a review. Briefings in bioinformatics **21**(5), 1609–1627 (2020)

[9] Zheng, W., Lin, H., Luo, L., Zhao, Z., Li, Z., Zhang, Y., Yang, Z., Wang, J.: An attention-based effective neural model for drug-drug interactions extraction. BMC bioinformatics **18**(1), 1–11 (2017)

[10] Tsoukalas, A., Albertson, T., Tagkopoulos, I., *et al.*: From data to optimal decision making: a data-driven, probabilistic machine learning approach to decision support for patients with sepsis. JMIR medical informatics **3**(1), 3445 (2015)

[11] Cheerkoot-Jalim, S., Khedo, K.K.: A systematic review of text mining approaches applied to various application areas in the biomedical domain. Journal of Knowledge Management (2020)

[12] Liu, S., Chen, K., Chen, Q., Tang, B.: Dependency-based convolutional neural network for drug-drug interaction extraction. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1074–1080 (2016). IEEE

[13] Silva, A., Portela, F., Santos, M.F., Machado, J., Abelha, A.: Text Mining Models to Predict Brain Deaths Using X-Rays Clinical Notes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10089 LNAI, pp. 153–163 (2017)

[14] Ferreira, D., Silva, S., Abelha, A., Machado, J.: Recommendation system using autoencoders. Applied Sciences (Switzerland) **10**(16) (2020)

[15] Neto, C., Brito, M., Lopes, V., Peixoto, H., Abelha, A., Machado, J.: Application of data mining for the prediction of mortality and occurrence of complications for gastric cancer patients. Entropy **21**(12) (2019)

[16] Martins, B., Ferreira, D., Neto, C., Abelha, A., Machado, J.: Data mining for cardiovascular disease prediction. Journal of medical systems **45**(1) (2021)

[17] Abacha, A.B., Chowdhury, M.F.M., Karanasiou, A., Mrabet, Y., Lavelli, A., Zweigenbaum, P.: Text mining for pharmacovigilance: Using machine

learning for drug name recognition and drug–drug interaction extraction and classification. Journal of biomedical informatics **58**, 122–132 (2015)

[18] Zaikis, D., Vlahavas, I.: Tp-ddi: Transformer-based pipeline for the extraction of drug-drug interactions. Artificial Intelligence in Medicine **119**, 102153 (2021)

[19] Kavuluru, R., Rios, A., Tran, T.: Extracting drug-drug interactions with word and character-level recurrent neural networks. In: 2017 IEEE International Conference on Healthcare Informatics (ICHI), pp. 5–12 (2017). IEEE

[20] Zeng, D., Sun, C., Lin, L., Liu, B.: Lstm-crf for drug-named entity recognition. Entropy **19**(6), 283 (2017)

[21] Sun, C., Yang, Z., Wang, L., Zhang, Y., Lin, H., Wang, J.: Biomedical named entity recognition using bert in the machine reading comprehension framework. Journal of Biomedical Informatics **118**, 103799 (2021). https://doi.org/10.1016/j.jbi.2021.103799

[22] Wang, W., Yang, X., Yang, C., Guo, X., Zhang, X., Wu, C.: Dependency-based long short term memory network for drug-drug interaction extraction. BMC bioinformatics **18**(16), 99–109 (2017)

[23] Sahu, S.K., Anand, A.: Drug-drug interaction extraction from biomedical texts using long short-term memory network. Journal of biomedical informatics **86**, 15–24 (2018)

[24] Zhang, Y., Zheng, W., Lin, H., Wang, J., Yang, Z., Dumontier, M.: Drug–drug interaction extraction via hierarchical rnns on sequence and shortest dependency paths. Bioinformatics **34**(5), 828–835 (2018)

[25] Sun, X., Dong, K., Ma, L., Sutcliffe, R., He, F., Chen, S., Feng, J.: Drug-drug interaction extraction via recurrent hybrid convolutional neural networks with an improved focal loss. Entropy **21**(1), 37 (2019)

[26] Mahendran, D., Nawarathna, R.: An automated method to extract information in the biomedical literature about interactions between drugs. In: 2016 Sixteenth International Conference on Advances in Ict for Emerging Regions (icter), pp. 155–161 (2016). IEEE

[27] Bui, Q.-C., Sloot, P.M., Van Mulligen, E.M., Kors, J.A.: A novel feature-based approach to extract drug–drug interactions from biomedical text. Bioinformatics **30**(23), 3365–3371 (2014)

[28] Lewis, P., Ott, M., Du, J., Stoyanov, V.: Pretrained language models for biomedical and clinical tasks: Understanding and extending the

state-of-the-art. In: Proceedings of the 3rd Clinical Natural Language Processing Workshop, pp. 146–157. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.clinicalnlp-1.17

[29] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics (2019). https://doi.org/10.1093/bioinformatics/btz682

[30] Datta, T.T., Shill, P.C., Al Nazi, Z.: Bert-d2: Drug-drug interaction extraction using bert. In: 2022 International Conference for Advancement in Technology (ICONAT), pp. 1–6 (2022). IEEE

[31] Wen, A., Sun, X., Yu, K., Wu, Y., Zhang, J., Yuan, Z.: Drug-drug interaction extraction using pre-training model of enhanced entity information. In: 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), pp. 527–532 (2020). IEEE

[32] Wishart, D.S., Feunang, Y.D., Guo, A.C., Lo, E.J., Marcu, A., Grant, J.R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., *et al.*: Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic acids research **46**(D1), 1074–1082 (2018)

[33] Barros, R.L.F.: Interacções medicamentosas e definição de modelo de controlo de risco. PhD thesis, Faculdade de Ciências e Tecnologia (2010)

[34] DesignGroup, C.B..D.: DDInter (2020). http://ddinter.scbdd.com/

[35] Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: State of the art, current trends and challenges. Multimedia tools and applications, 1–32 (2022)

[36] Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., Declerck, T.: The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. Journal of biomedical informatics **46**(5), 914–920 (2013)

[37] Segura-Bedmar, I., Martínez Fernández, P., Herrero Zazo, M.: Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). (2013). Association for Computational Linguistics

[38] Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D.M., Sayle, R.A., Batista-Navarro,

R.T., Rak, R., Huber, T., Rocktäschel, T., Matos, S., Campos, D., Tang, B., Xu, H., Munkhdalai, T., Ryu, K.H., Ramanan, S., Nathan, S., Žitnik, S., Bajec, M., Weber, L., Irmer, M., Akhondi, S.A., Kors, J.A., Xu, S., An, X., Sikdar, U.K., Ekbal, A., Yoshioka, M., Dieb, T.M., Choi, M., Verspoor, K., Khabsa, M., Giles, C.L., Liu, H., Ravikumar, K.E., Lamurias, A., Couto, F.M., Dai, H.-J., Tsai, R.T.-H., Ata, C., Can, T., Usié, A., Alves, R., Segura-Bedmar, I., Martínez, P., Oyarzabal, J., Valencia, A.: The CHEMDNER corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics **7**(S1) (2015). https://doi.org/10.1186/1758-2946-7-s1-s2