BIOINFORMATICS ORIGINAL PAPER
Vol. 30 no. 23 2014, pages 3365–3371
doi:10.1093/bioinformatics/btu557

Data and text mining

Advance Access publication August 20, 2014

A novel feature-based approach to extract drug–drug interactions from biomedical text

Quoc-Chinh Bui[1,*], Peter M.A. Sloot[2,3,4], Erik M. van Mulligen[1] and Jan A. Kors[1]

[1]Department of Medical Informatics, Erasmus University Medical Center Rotterdam, [2]Informatics Institute, University of Amsterdam, The Netherlands, [3]Complexity Institute, Nanyang Technological University, Singapore and [4]ITMO University, St. Petersburg, Russian Federation

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Knowledge of drug–drug interactions (DDIs) is crucial for health-care professionals to avoid adverse effects when co-administering drugs to patients. As most newly discovered DDIs are made available through scientific publications, automatic DDI extraction is highly relevant.

Results: We propose a novel feature-based approach to extract DDIs from text. Our approach consists of three steps. First, we apply text preprocessing to convert input sentences from a given dataset into structured representations. Second, we map each candidate DDI pair from that dataset into a suitable syntactic structure. Based on that, a novel set of features is used to generate feature vectors for these candidate DDI pairs. Third, the obtained feature vectors are used to train a support vector machine (SVM) classifier. When evaluated on two DDI extraction challenge test datasets from 2011 and 2013, our system achieves F-scores of 71.1% and 83.5%, respectively, outperforming any state-of-the-art DDI extraction system.

Availability and implementation: The source code is available for academic use at http://www.biosemantics.org/uploads/DDI.zip

Contact: q.bui@erasmusmc.nl

Received on April 9, 2014; revised on August 11, 2014; accepted on August 12, 2014

1 INTRODUCTION

Drug–drug interaction (DDI) is a situation when one drug increases or decreases the effect of another drug (Tari *et al.*, 2010). Information about DDIs is crucial for drug administration to avoid adverse drug reactions or therapeutic failure (van Roon *et al.*, 2009). For example, a recent study reports that DDIs are a significant cause of hospital admissions (Dechanont *et al.*, 2014). While specialized databases are available for finding known DDIs, such as DrugBank (http://www.drugbank.ca) or Micromedex (http://micromedex.com), their coverage is limited and there are discrepancies in DDI listing between existing databases (Wong *et al.*, 2008). As a consequence, most of newly discovered DDIs need to be extracted from scientific publications (Herrero-Zazo *et al.*, 2013). Text-mining techniques such as automatic relation extraction have been applied successfully in large-scale experiments to extract various types of relations [e.g. protein–protein interactions (PPIs), gene-disease] efficiently (Hahn *et al.*, 2012; Rebholz-Schuhmann *et al.*, 2012). Therefore, automatic DDI extraction methods can be particularly relevant to effectively extract DDIs and corresponding evidence from the scientific literature.

To develop and evaluate automatic DDI extraction methods, a DDI corpus has been created by Herrero-Zazo *et al.* (2013). This corpus was manually annotated with 18 502 pharmacological substances, mainly consisting of generic and brand names, and 5028 DDIs. With the availability of this corpus and the introduction of two DDI extraction challenges in 2011 and 2013 (Segura-Bedmar *et al.*, 2011a, 2013), several approaches have been proposed to extract DDIs from biomedical text. In both challenges, systems built on machine learning (ML) approaches were dominant and achieved the best results (Segura-Bedmar *et al.*, 2011a, 2013). In these systems, the DDI extraction tasks are modeled as classification problems where each candidate DDI pair is classified as an interacting pair or not. To build the classification models, data from annotated DDI corpora are often transformed into more structural representations using various natural language processing (NLP) tools. Among these ML-based systems, support vector machine (SVM) methods are the most popular (Segura-Bedmar *et al.*, 2013). In general, ML-based DDI extraction systems can be categorized into two groups, namely feature- and kernel-based methods.

In feature-based systems, each data instance is represented as a feature vector in an n-dimensional space. The main focus in these systems is to define features that potentially best represent the data characteristics. For DDI extraction tasks, various feature types have been used ranging from lexical to syntactic and semantic information. For example, Segura-Bedmar *et al.* (2011b) developed a system using bag-of-words and local context features. To improve the performance of feature-based systems, some authors combine multiple types of features with the hope that these features can complement each other. He *et al.* (2013) introduced a system that uses lexical, semantic and domain knowledge features. Chowdhury and Lavelli (2013a) proposed a system that combines heterogeneous features. Their system comprises lexical, syntactic, semantic and negation features derived from sentences and their corresponding parse trees.

In kernel-based systems, the structural representations of data instances, e.g. syntactic parse trees or dependency graphs, are exploited. Various kernels have been proposed to quantify the similarities between two instances by computing the similarities of their representations. These kernels differ from each other

*To whom correspondence should be addressed.

based on how syntactic representations are used and how similarity functions are calculated (Tikk *et al.*, 2013). For the DDI extraction challenges, the use of kernels varies between the participating systems. Among them, the most commonly used kernels are all-paths graph kernel (Airola *et al.*, 2008), shallow linguistic kernel (Giuliano *et al.*, 2006) and path-enclose tree kernel (Moschitti, 2004). As the proposed kernels exploit different types of structural representations and similarity functions, they all have pros and cons. To compensate for the weakness of each individual kernel, kernel combination is often used. For example, Chowdhury and Lavelli (2013b) proposed a hybrid kernel, which combines three different kernels. Their system achieved the best results in the DDI extraction 2013 challenge (Task 2). Furthermore, the combination can take place at the output level (ensemble approach) where the output of multiple systems is combined using a voting scheme. Thomas *et al.* (2011) developed a system that combines the output of two kernel-based systems and a case-based reasoning system using a majority voting scheme. This system yielded the best result in the DDI extraction 2011 challenge.

Although systems using feature-based kernels alone did not yield the best performance in the DDI extraction challenges, feature-based kernels still play an important role in relation extraction tasks. In fact, the winning teams of the DDI extraction 2011 and 2013 challenges both incorporate feature-based kernels proposed by Giuliano *et al.* (2006) as part of their systems. Furthermore, Miwa *et al.* (2009) have shown that their feature-based PPI extraction system achieved state-of-the-art results on five PPI corpora. A recent study by Tikk *et al.* (2013) on the performance of various types of kernels for PPI extraction tasks also suggests that to improve the performance of the current PPI extraction systems, novel feature sets should be explored over novel kernel functions. This suggestion may also apply to the DDI extraction tasks, as most current approaches to extract DDI pairs have also previously been used to extract PPI pairs.

In this article, we propose a novel feature-based approach to extract DDIs from biomedical text. Our approach differs from existing approaches in two ways. First, we partition candidate DDI pairs into five groups based on their syntactic structures. Second, we apply a set of novel features that is optimized for each group based on the syntactic properties. Our results show that the proposed system achieves the best results in terms of F-scores and performance efficiency when compared with the state-of-the-art DDI extraction systems.

2 METHODS

Our method consists of three steps. First, we apply text preprocessing to convert input sentences into structured representations. Second, a feature vector for each candidate DDI pair is extracted from the corresponding structured representation using predefined feature sets. In the last step, the obtained feature vectors are used to train an SVM classifier to generate a predictive model, which is used to classify candidate DDI pairs of the test dataset.

2.2 Text preprocessing

The text preprocessing step consists of filtering out irrelevant sentences, entity blinding, word tokenizing, part-of-speech (POS) tagging and parsing sentences with a shallow parser. We manually created a list of 292 trigger words by combining a list of trigger words previously used to extract PPIs (Bui *et al.*, 2011) and some trigger words specific to DDI taken from the training dataset. Sentences that contain one drug or have no trigger word are filtered out. Next, to improve generalization of the input sentences, all drug names are blinded by assigning names as DRUGi where i is the drug index. Each sentence is then tokenized and POS tagged with the LingPipe NLP toolkit (http://alias-i.com/lingpipe). Finally, the tokens and their tags are used as input for the OpenNLP shallow parser (https://opennlp.apache.org/) to produce chunks.

2.3 Structured representation

We adapt the structured representation proposed by Bui and Sloot (2012) to express candidate DDI pairs. This structured representation, which consists of three syntactic layers (chunk, phrase and clause), is generated based on the chunks outputted from the shallow parser. As there are many cases where DDI pairs span into more than one single clause, we represent these cases using multiple single clauses. We modify the structured representation as follows:

Phrase: consists of a list of chunks (i.e. the output of the shallow parser). Figure 1b shows examples of phrases (dashed boxes), which consist of noun chunks (NCs; plain boxes) connected by preposition chunks (PCs; shadowed boxes).

Clause: consists of a verb chunk and two phrases that are located in the left and in the right of the verb chunk. Complex sentences are represented by multiple clauses. For example, Figure 1a shows a clause that has a verb chunk connected with the left phrase (subject) and the right phrase (object). Figure 1c shows a complex sentence that consists of three clauses. Furthermore, to reduce the number of clauses generated for each input sentence, only verb chunks that belong to the main clauses are used to construct the structured representation.

With the proposed structured representation, we can express relationship of almost all drug pairs. Figures 1a and b show examples of drug

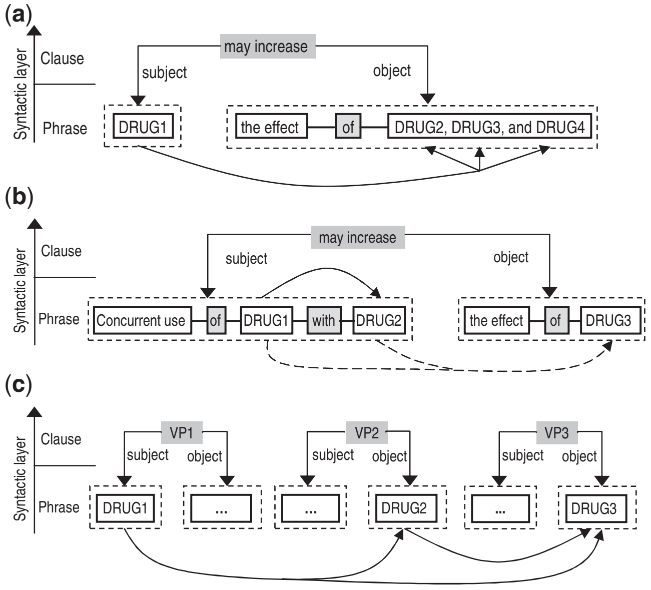

Fig. 1. Structured representation for DDI pairs. (**a**) Examples of positive DDI pairs expressed by a clause. (**b**) Examples of a positive DDI pair expressed by a phrase (subject) and of negative DDI pairs, indicated by dashed lines, expressed by a clause. (**c**) An example of a complex sentence, which consists of multiple clauses. DRUG1–DRUG2 and DRUG2–DRUG3 pairs span over two clauses, whereas the DRUG1–DRUG3 pair spans over three clauses

pairs that interact (positive DDI) and that do not interact (negative DDI) expressed by the structured representations.

2.3 Features

In this section, we describe a set of novel features that are specifically designed to exploit the strength of the structured representations. To generate features for each candidate DDI pair, we find the smallest syntactic container (e.g. a phrase, a clause or clauses) from the structured representation containing that pair. For example, the smallest syntactic container of the DRUG1–DRUG2 pair in Figure 1b is a phrase, whereas the smallest syntactic container of the DRUG2–DRUG3 pair in Figure 1c encloses two clauses. Given a candidate DDI pair and its syntactic container, we check whether the syntactic container contains any trigger words. If the syntactic container functions as a subject, we also check its right verb chunk for trigger words, as there are cases in which trigger words do not belong to the subjects but to their right verb chunks. If no trigger word is detected, then the candidate DDI pair is skipped, otherwise the following features are generated depending on its container type (e.g., subject, clause):

Lexical features: are used to capture relations between each drug of the candidate DDI pair and its surrounding tokens. These relations might reveal the syntactic role of the drug within the phrase containing it, such as whether the drug is a part of the coordination or is an abbreviation of another drug. Lexical features of each drug are three tokens on the left and three tokens on the right of that drug. Left and right tokens are distinguished by adding _L and _R suffixes, respectively. In addition, if a token is a drug (e.g. DRUG1 or DRUG2) then that token is replaced by 'arg'. For example, lexical features of the DRUG2 in Figure 1b are: of_L, arg_L, with_L. As DRUG2 is the last token of that phrase, there is no feature extracted from the right side.

Phrase features: are applicable for a candidate DDI pair of which the syntactic container is a phrase. These features are designed to capture relations of the candidate DDI pair and trigger words that belong to the phrase containing that pair. For each trigger word, we determine its relative position within the phrase by checking the following cases:

- Trigger [prep]* arg1 [prep]* arg2 (case 1)
- Arg1 [prep]* trigger [prep]* arg2 (case 2)
- Arg1 [prep]* arg2 [prep]* trigger (case 3)

Here *prep* are prepositions connecting chunks that contain the trigger and the DDI pair. Arg1 and arg2 are drugs of the (ordered) candidate DDI pair. The '*' indicates that zero or more prepositions are required. Based on the obtained case, corresponding features are generated to represent the position between the trigger and the candidate DDI pair (i.e. left, middle or right) and to indicate which prepositions are used to connect the trigger and the target pair as well as the chunks between the drugs of the target pair. For example, features generated for the DRUG1–DRUG2 pair in Figure 1b are *use_of_arg1* and *arg1_with_arg2_case1*. Furthermore, if there is a negative modifier (e.g. no, not) which belongs to the same chunk that contains a trigger, we insert the modifier as the prefix for that trigger.

As it is non-trivial to automatically determine which trigger actually has a relation with (i.e. governs) the candidate DDI pair, all detected triggers are used to generate phrase features.

Verb features: are bag-of-words (unigrams and bigrams) generated from the verb chunk of the clause to which the candidate DDI pair belongs. The verb features indicate how the drug in the left phrase (subject) and the drug in the right phrase (object) are related.

Syntactic features: are designed to capture the surrounding syntactic structure of each drug of the candidate DDI pair within the phrase to which it belongs. To do this, we assign indices for all preceding noun and preposition chunks which connect to the noun chunk containing that drug. Furthermore, we also check whether there is any drug succeeding that drug and which prepositions are used to connect them. For example, the syntactic features generated for DRUG1 in Figure 1b are *NC1*, *PC2*, *has_more_args* and *with_arg*. Together with verb features, syntactic features particularly help to distinguish between DDI pairs that have a drug governed by its preceding noun chunks and DDI pairs that have drugs spanning into two phrases (i.e. subject and object) of a clause. For example, consider the positive DRUG1–DRUG2 pair in Figure 1a and the negative DRUG2–DRUG3 pair in Figure 1b. Although both pairs have the same sequence of tokens, if the syntactic structure is used then DRUG1 in Figure 1a and DRUG2 in Figure 1b have completely different syntactic features.

Auxiliary features: consist of three features that capture information related to the drugs of the target pair. In particular, the first feature keeps track if drug names of the pair are real names versus pronouns (e.g. these drugs, this drug). The second feature denotes whether the drugs have the same name, and the third feature indicates whether the target drugs are in the same chunk.

2.4 Partitioning DDI pairs

In a previous study, Bui *et al.* (2011) showed that partitioning candidate PPI pairs based on syntactic properties and selecting partition-specific feature improved the performance of their PPI extraction system. Following this strategy, we categorize candidate DDI pairs into different groups based on their syntactic containers. To reduce the number of syntactic groups being generated, we only consider candidate DDI pairs that span over at most two clauses. For example, the DRUG1–DRUG3 pair in Figure 1c is ignored, as it spans over three clauses. This partitioning process results in five syntactic groups, namely subject, object, clause, clause_2 and NP. Here clause_2 denotes a syntactic structure that spans over two clauses, and NP denotes an input sentence that contains only a phrase.

Owing to space limitations, we refer to the Supplementary source code for more details on text preprocessing and feature generation.

2.5 Machine learning

Recent relation extraction competitions have shown that the use of SVMs in relation extraction systems is dominant and systems that use SVMs achieved the best performance (Nédellec *et al.*, 2013; Segura-Bedmar *et al.*, 2011a, 2013). In this study, we use the LIBSVM classifier with a default RBF kernel (http://www.csie.ntu.edu.tw/ ~ cjlin/libsvm/) for classification of DDI pairs. All individual features extracted for each DDI pair are normalized and combined into a single feature vector as proposed by Miwa *et al.* (2009). To find the best parameter C and gamma for each model, we use the CVParameterSelection function from the WEKA toolbox (http://www.cs.waikato.ac.nz/ml/weka/).

3 RESULTS AND DISCUSSION

3.1 Datasets

We use the DDI extraction 2011 and 2013 datasets (hereafter referred to as DDI-2011 and DDI-2013) provided by the DDI extraction 2011 and 2013 challenges to evaluate our extraction method. Each dataset consists of two parts, a training dataset and a test dataset. There are differences between the two challenge datasets. The DDI-2011 datasets contain documents selected from the DrugBank database, whereas the DDI-2013 datasets consist of documents selected from the DrugBank database and Medline abstracts. Furthermore, in the DDI-2011 dataset, each drug pair was annotated either as a true interaction (positive instance) or no interaction (negative instance), whereas the DDI-2013 datasets have more fine-grained annotations with

Table 1. Statistics of the DDI-2011 and DDI-2013 training and test datasets

Corpus	Training			Testing		
	Sen.	Pos.	Neg.	Sen.	Pos.	Neg.
DDI-2011	4267	2402	21 425	1539	755	6271
DB-2013	5675	3788	22 217	973	884	4426
ML-2013	1031	232	1555	326	95	365

Notes. The DDI-2013 datasets are split into two subsets (DB-2013 and ML-2013) based on document types. Sen., Pos. and Neg. denote numbers of input sentences, positive instances and negative instances, respectively.

Table 2. Statistics of the transformed training datasets after applying text preprocessing steps

Group	DB-2013		ML-2013		DDI-2011	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Subject	876	4301	29	250	600	4488
Object	356	4797	13	271	203	3770
Clause	1852	2238	121	211	1240	3212
Clause_2	341	871	12	85	163	1324
NP	197	1039	27	102	74	713
Total (known cases)	3622 (96%)	13 246 (60%)	202 (87%)	919 (59%)	2280 (95%)	13507 (63%)
Ignored cases	60	479	10	50	24	676
Filtered out/ skipped	106 (2.80%)	8492 (38.20%)	20 (8.60%)	586 (37.70%)	98 (4.10%)	7242 (33.80%)

Notes. Pos. and Neg. denote positive and negative instances, respectively.

different interaction types. Statistics of the datasets are shown in Table 1.

3.2 Transformation of datasets

When applying the text preprocessing and partitioning steps for each dataset, we obtain a transformed dataset where irrelevant DDI pairs are filtered out and the original dataset is split into five groups. Tables 2 and 3 show statistics of the transformed datasets for training and test datasets, respectively. The data in these tables indicate that the text preprocessing has effectively filtered out significant numbers of negative instances (TNs) with a small cost of missing positive instances (FNs). Overall, numbers of filtered instances vary from 2.5 to 4.1% for FNs and from 27.9 to 33.8% for TNs on the DrugBank datasets. However, numbers of FNs on the Medline dataset are unexpectedly high, ranging from 8.6 to 17.9.0%. Furthermore, a small number of positive instances are ignored during the partition step owing to their complex syntactic structures. These numbers are shown in Tables 2 and 3 as ignored cases.

In addition, the data from Tables 2 and 3 show that the numbers of instances vary significantly between groups of each dataset and across datasets. This indicates that the performance on each group might also differ accordingly.

3.3 Evaluation settings

We use the standard evaluation measures (Precision, Recall and F-score) proposed by the DDI extraction challenge to evaluate the performance of our system (Segura-Bedmar *et al.*, 2013). As our method mainly focuses on the detection of interaction pairs, we ignore the interaction types annotated in the DDI-2013 dataset. (The detection of DDI pairs is an important step in the extraction pipeline of most of the systems that participated in the DDI extraction 2013 challenge, including the top two systems). In addition, because we partition each dataset into five groups, we need to train the classifier separately for each group. To find the optimal feature sets for these groups, we tried various combinations of the proposed features. The best feature sets for each group are shown in Table 4. These features were determined based on the DB-2013 training set but used for all evaluations.

We evaluate the performance of our system on each test dataset after training on the corresponding training dataset, except for the ML-2013 test dataset. For this test dataset, the system is trained on the combined DB-2013 and the ML-2013 training datasets as suggested by Chowdhury and Lavelli (2013b) and Thomas *et al.* (2013).

3.4 Performance of DDI extraction

Table 5 shows the results of our system evaluated on the DDI-2011 and DDI-2013 test datasets. To understand its performance on different document types (i.e. DrugBank and Medline abstracts), we present the results of the DDI-2013 sub datasets separately. Furthermore, to calculate recall, all positive instances missed by the previous preprocessing steps are considered as FNs. Besides reporting the overall performance of the whole dataset, we also present the performances of individual groups. Recall for each group is calculated using data from Table 3 (which do not take into account filtered and ignored instances), whereas the recall for the overall performance for each test dataset is calculated using data from Table 1.

The results in Table 5 show that our system performs well on the DB-2013 and DDI-2011 test datasets with F-scores of 83.5 and 71.1%, respectively. However, its performance decreases on the Medline test dataset with an F-score of 59.2%, which is 24.3 points lower than that of the DB-2013 test dataset. This performance decrease stems from the low recall, which can partly be explained by the loss of positive instances during the preprocessing steps. In addition, for each dataset, the performance on each group also differs significantly. These performance differences might be due to three factors. First, the ratio of the positive and negative instances varies among all groups (see Tables 2 and 3). This causes the performance degradation for groups that have smaller positive/negative ratios (Van Hulse *et al.*, 2007). Second, the selection of different feature sets for various syntactic groups may also account for the differences in performance. Third, the annotation quality of the DB-2013 is better than that of DB-2011, which was annotated automatically without any manual revision (Herrero-Zazo *et al.*, 2013).

Table 3. Statistics of the transformed test datasets after applying text preprocessing steps

Group	DB-2013		ML-2013		DDI-2011	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Subject	156	782	21	20	179	1000
Object	90	1174	16	77	78	1429
Clause	504	429	36	58	376	997
Clause_2	37	229	2	26	54	280
NP	61	367	3	23	34	567
Total (known cases)	848 (96%)	2981 (68%)	78 (82%)	204 (57%)	721 (96%)	4273 (68%)
Ignored cases	14	178	0	4	4	131
Filtered out/skipped	22 (2.50%)	1222 (27.90%)	17 (17.90%)	148 (41.60%)	30 (4.00%)	1867 (29.80%)

Notes. Pos. and Neg. denote positive and negative instances, respectively.

Table 4. Optimized features for each syntactic group

Group	Lexical	Phrase	Verb	Syntactic	Auxiliary
Subject	X	X	X	X	X
Object	X	X	X		X
Clause	X		X	X	X
Clause-2	X		X	X	X
NP	X	X			X

Table 6 shows the performance comparison between our system (BioSem) and the top-performance systems participating in the DDI-2013 extraction challenge (Task 2). The data show that our system outperforms the top five systems on the DB-2013 test dataset with an F-score increase ranging from 0.8 to 13.2 points. While the recall of our system is lower than the best system (81.2 versus 83.8%), its precision is significantly higher (85.9 versus 81.6%). Furthermore, our system also yields better results when compared with these systems on the ML-2013 test dataset. The results in Table 7 show that the BioSem achieves an F-score of 59.2%, which is higher than the other systems 6.2–17.1 points. It is worth noting that the systems that participated in the challenges had to be developed under strict time constraints, which may have affected their performance. Nevertheless, the authors of the top-performing systems have participated in the DDI-2011 extraction challenge and thus were familiar with the task and could fine-tune their systems using the DDI-2011 test dataset.

To provide a fair performance comparison, we present the evaluation results of the best known systems that run on the DDI-2011 post-challenge test dataset in Table 8. We also provide the results of the best system of the DDI-2011 extraction challenge for reference. The data show that post-challenge systems achieve higher performance in terms of F-scores as compared with the best system of the DDI-2011 extraction challenge. These performance improvements might stem from the fact that these systems have a better design and/or could be fine-tuned on the available test dataset. Compared with these post-challenge systems, our system yields better results with F-score improvements ranging from 1.9 to 2.2 points. It is worth noting that the system proposed by Chowdhury and Lavelli (2013b) is the same system that achieved the best results in DDI-2013 challenge.

Table 5. Evaluation results on the DDI-2011 and DDI-2013 test datasets

Group	DB-2013		ML-2013		DDI-2011	
	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
Subject	83.92	76.92	86.67	61.90	75.65	81.56
Object	84.72	67.78	54.55	37.50	81.43	73.08
Clause	86.08	94.44	71.79	77.78	65.77	84.84
Clause_2	91.67	59.46	100.00	50.00	76.19	29.62
NP	88.64	63.93	25.00	66.67	64.29	26.47
Overall performance						
Precision (%)	85.88		67.57		69.85	
Recall (%)	81.22		52.63		72.45	
F-score (%)	83.48		59.17		71.13	

Notes. The DDI-2013 test datasets are split into two subsets (DB-2013 and ML-2013) based on document types. P and R denote precision and recall, respectively.

Table 6. Performance comparison with the top five systems participating in the DDI-2013 extraction challenge on the DB-2013 test dataset

Team	Precision (%)	Recall (%)	F-score (%)
FBK-irst	81.6	83.8	82.7
WBI	81.4	75.5	78.3
SCAI	79.6	68.1	73.4
UTurku	84.3	63.8	72.6
UC3M	65.6	75.8	70.3
BioSem	85.9	81.2	83.5

3.5 Performance analysis

In this section, we address some issues related to the performance of the proposed system as well as discuss its complexity with respect to the state-of-the-art systems.

3.5.1 Performance variation on different datasets In the previous section, we have mentioned that the ratio of positive and negative instances might directly contribute to the differences in performance between syntactic groups (e.g. subject, object, etc.) of each dataset. This phenomenon can also be observed in the same groups across different datasets. For example, on the DB-2013 dataset, the ratios of positive/negative instances of the *clause* group are 0.83 and 1.20 for training and test datasets, whereas on the DDI-2011 dataset these values are 0.39 and 0.38, respectively (see Tables 2 and 3). These differences might explain why precision and recall of the *clause* group differs between these two datasets: 86.1 versus 65.8% for precision and 94.4 versus 84.8% for recall. Furthermore, this might also explain the high precision of the *subject* group on the ML-2013 test

Table 7. Performance comparison with the top five systems participating in the DDI-2013 extraction challenge on the ML-2013 test dataset

Team	Precision (%)	Recall (%)	F-score (%)
FBK-irst	55.8	50.5	53.0
WBI	62.5	42.1	50.3
UWM-TRIADS	38.7	63.0	47.9
SCAI	43.1	52.6	47.4
UC3M	31.3	64.2	42.1
BioSem	67.6	52.6	59.2

Table 8. Performance comparison of systems on the post-challenge DDI-2011 test dataset

Team	Precision (%)	Recall (%)	F-score (%)
WBI (1st 2011)	60.5	71.9	65.7
Chowdhury and Lavelli (2013b)	63.5	75.2	68.9
He *et al.* (2013)	66.2	72.6	69.2
BioSem	69.9	72.5	71.1

dataset, as the positive/negative ratios between training and test datasets are 0.11 and 1.05, respectively.

Another issue that might affect the system performance is the size of the datasets. This is clearly visible for the ML-2013 dataset, which is significantly smaller (14 times) than the DB-2013 dataset. Moreover, learning a model from a small training set is one of the challenges of an ML-based approach. This problem is even harder in our case since we further split the training set into five sub datasets. For example, when we used the ML-2013 datas*et al*one for training, our system achieved an F-score of 35.4% on the ML-2013 test dataset (data not shown). However, when trained on the combined DB-2013 and ML-2013 training datasets and evaluated on the ML-2013 test set, the F-score increases to 59.2%. This indicates that even though there are differences in structure between the document types (Cohen *et al.*, 2010) of two datasets, increasing the size of the ML-2013 training set by adding training instances from the DB-2013 set, to some extent, helps improving the performance of our system on this test dataset.

3.5.2 Contribution of the proposed feature sets When applying an ML-based approach for relation extraction tasks, each candidate pair is classified independently as being a true interaction pair or not. The benefit of this approach is that it can easily be used with any (binary) classifier. However, when each candidate DDI pair is considered independently, it is taken out of context. In other words, the dependencies between the drugs of the candidate DDI pair and their neighboring drugs might be missed, which might lead to a wrong classification. For example, consider a positive DRUG1–DRUG2 pair and two negative DRUG1–DRUG3 and DRUG2–DRUG3 pairs in the sentence ‘Concurrent use of DRUG1 with DRUG2 may increase the effect of DRUG3’ as shown in Figure 1b. For the DRUG2–DRUG3 pair, if only lexical features are used then one may miss the information that DRUG2 has already participated in a relation with DRUG1. For the DRUG1–DRUG3 pair, even if a dependency tree is used, one might still miss the information that DRUG1 has a relation with DRUG2. To address this problem, previous systems usually combine various types of features so that they can complement each other. In our system, we explicitly tackle this problem by introducing three novel feature sets, namely verb, phrase and syntactic features.

Table 9. Contribution of phrase, syntactic and verb features to the performance of our system

Features	P (%)	R (%)	F (%)
Lex + Aux + Phrase + Syntactic + Verb (1)	85.64	79.63	82.53
(1) - Verb	81.9	76.24	78.97
(1) - Phrase	84.63	79.75	82.12
(1) - Syntactic	83.06	81	82.01
(1) - Phrase - Syntactic	81.7	80.32	81
Optimized feature sets	85.88	81.22	83.48

Notes. The results are evaluated on the DB-2013 test dataset. Verb features are not applicable to NP group, and phrase features are not applicable to clause and clause-2 groups. Lex, Aux, P, R and F denote lexical, auxiliary, precision, recall and F-score, respectively.

Table 9 shows the contributions of the phrase, syntactic and verb features on the performance of our system when evaluated on the DB-2013 test dataset. The data show that when the verb features are removed, the performance in terms of F-score degrades 3.56% compared with that of the whole feature set. While removing the phrase or syntactic feature alone decreases the performance slightly, removing both phrase and syntactic features results in the performance decreases 1.53%. This means that one of these features may only be suitable for certain groups. This phenomenon is clearly visible when we apply the optimized feature sets from Table 4 to the test dataset, resulting in an increase of 0.95% on the F-score compared with that of the whole feature sets.

In addition, by mapping each candidate DDI pair into a syntactic container before generating features, we can enhance the lexical features by not generating unnecessary tokens surrounding each drug of the candidate DDI pair. For example, the number of lexical features generated for DRUG2 in Figure 1b is three features instead of six features for systems that use a flat structure.

3.5.3 Computational performance and complexity To increase the performance of DDI extraction systems, most of top-performing systems use either ensemble approaches (Thomas *et al.*, 2011, 2013) or kernel combination approaches (Chowdhury and Lavelli, 2013b; He *et al.*, 2013). While they manage to increase the performances, the computational resources and the complexity of their systems also increase. Furthermore, some systems also incorporate domain knowledge (He *et al.*, 2013; Thomas *et al.*, 2013) to enhance the performance, but this hinders the adaptation of these systems to new relation extraction tasks.

In contrast, our proposed feature-based system uses a small set of features to generate feature vectors from a simple syntactic representation. It uses a shallow parser for analyzing input sentences and requires only a single kernel to build predictive models. Therefore, it is simpler and requires less computational time compared with the other ML-based systems. For example, our system requires 51 s to process the DB-2013 dataset (22 s for the text preprocessing step and 29 s for training and classifying instances). This experiment was performed on a laptop with an Intel Core i7-2640 M, 2.8 GHz processor.

3.5.4 Error analysis To identify the main sources of error of our system, we analyze all errors [118 false positives (FPs), 130 FNs] produced by our system when evaluated on the DB-2013 test dataset. Overall, these errors (both FPs and FNs) can be categorized into four groups. The first group of errors (22 FPs, 39 FNs) is caused by parser errors or incorrect construction of structured representations. These errors lead to the wrong categorization of candidate DDI pairs. The second error group (34 FPs) is caused by a non-deterministic context, where the syntactic containers of the candidate DDI pairs alone are not enough to determine the outcome. The third error group (42 FPs, 91 FNs) is caused by unusual syntactic structures of the input sentences, anaphora problems and the long distance between two drugs (measured by the number of chunks) of the candidate DDI pairs. The fourth error group (20 FPs) consists of cases where candidate DDI pairs syntactically seem to be true DDI pairs.

While most of the errors are non-trivial, the errors caused by input sentences with special syntactic structures can be tackled if rules are defined to convert these input sentences into a form that can be handled by the structured representation. For the other errors, substantial changes in the system are needed to further improve the current performance.

4 CONCLUSIONS

In this study, we have proposed a novel feature-based approach to extract DDIs from text. The key factors of our approach are the combination of the novel feature sets and the partition of the datasets. By partitioning the original dataset into subsets based on their syntactic properties, we obtain more consistent sub datasets and can optimize feature selection for each sub dataset. Furthermore, by combining the strength of various types of features, our system is robust and generalizes well on different datasets. The evaluation results show that our system achieves better performance than the state-of-the-art systems on various test datasets.

Our approach is simple and more efficient in terms of computational time than other ML-based systems, as it uses a small set of features and a default SVM kernel. Furthermore, the proposed feature sets are generic, except for the auxiliary feature set. While the system is initially proposed to extract DDIs, it can easily be adapted to other binary relation extraction tasks, such as PPIs and gene–disease relations.

Funding: PMAS is partially supported by Russian Scientific Foundation, proposal #14-21-0037.

Conflict of interest: none declared.

REFERENCES

Airola,A. *et al.* (2008) All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, **9** (**Suppl. 11**), S2.

Bui,Q.-C. *et al.* (2011) A hybrid approach to extract protein-protein interactions. *Bioinformatics*, **27**, 259–265.

Bui,Q.-C. and Sloot,P.M.A. (2012) A robust approach to extract biomedical events from literature. *Bioinformatics*, **28**, 2654–2661.

Chowdhury,M. and Lavelli,A. (2013a) Exploiting the scope of negations and heterogeneous features for relation extraction: a case study for drug-drug interaction extraction. In: *Proceedings of NAACL-HLT*. Atlanta, GA, USA, pp. 765–771.

Chowdhury,M. and Lavelli,A. (2013b) FBK-irst: a multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In: *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, GA, USA, pp. 351–355.

Cohen,K.B. *et al.* (2010) The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, **11**, 492.

Dechanont,S. *et al.* (2014) Hospital admissions/visits associated with drug-drug interactions: a systematic review and meta-analysis. *Pharmacoepidemiol. Drug Saf.*, **23**, 489–497.

Giuliano,C. *et al.* (2006) Exploiting shallow linguistic information for relation extraction from biomedical literature. In: *Proceedings of ACL 2006*. Trento, Italy, pp. 401–408.

Hahn,U. *et al.* (2012) Mining the pharmacogenomics literature—a survey of the state of the art. *Brief. Bioinform.*, **13**, 460–494.

He,L. *et al.* (2013) Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PLoS One*, **8**, e65814.

Herrero-Zazo,M. *et al.* (2013) The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions. *J. Biomed. Inform.*, **46**, 914–920.

Miwa,M. *et al.* (2009) A rich feature vector for protein-protein interaction extraction from multiple corpora. In: *Proceedings of the 2009 Conference on Empirical Methods in NLP*. ACL, Singapore, pp. 121–130.

Moschitti,A. (2004) A study on convolution kernels for shallow semantic parsing. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL'04*. ACL, Morristown, NJ, USA, pp. 335–342.

Nédellec,C. *et al.* (2013) Overview of BioNLP shared task 2013. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. Sophia, Bulgaria, pp. 1–7.

Rebholz-Schuhmann,D. *et al.* (2012) Text-mining solutions for biomedical research: enabling integrative biology. *Nat. Rev. Genet.*, **13**, 829–839.

Segura-Bedmar,I. *et al.* (2013) Semeval-2013 task 9: extraction of drug-drug interactions from biomedical texts. In: *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, GA, USA, pp. 341–350.

Segura-Bedmar,I. *et al.* (2011a) The 1st DDIExtraction-2011 challenge task: extraction of drug-drug interactions from biomedical texts. In: *Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDI Extraction 2011)*. Huelva, Spain.

Segura-Bedmar,I. *et al.* (2011b) Using a shallow linguistic kernel for drug-drug interaction extraction. *J. Biomed. Inform.*, **44**, 789–804.

Tari,L. *et al.* (2010) Discovering drug-drug interactions: a text-mining and reasoning approach based on properties of drug metabolism. *Bioinformatics*, **26**, i547–i553.

Thomas,P. *et al.* (2011) Relation extraction for drug-drug interactions using ensemble learning. In: *Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDI Extraction 2011)*. Huelva, Spain.

Thomas,P. *et al.* (2013) WBI-DDI: drug-drug interaction extraction using majority voting. In: *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, GA, USA, pp. 628–635.

Tikk,D. *et al.* (2013) A detailed error analysis of 13 kernel methods for protein-protein interaction extraction. *BMC Bioinformatics*, **14**, 12.

Van Hulse,J. *et al.* (2007) Experimental Perspectives on Learning from Imbalanced Data. In: *Proceedings of the 24th International Conference on Machine Learning*, ICML'07. ACM, New York, NY, USA, pp. 935–942.

Van Roon,E.N. *et al.* (2009) An evidence-based assessment of the clinical significance of drug-drug interactions between disease-modifying antirheumatic drugs and non-antirheumatic drugs according to rheumatologists and pharmacists. *Clin. Ther.*, **31**, 1737–1746.

Wong,C.-M. *et al.* (2008) Clinically significant drug-drug interactions between oral anticancer agents and nonanticancer agents: profiling and comparison of two drug compendia. *Ann. Pharmacother.*, **42**, 1737–1748.