EXTRACTING DRUG-DRUG INTERACTIONS FROM BIOMEDICAL TEXT USING A FEATURE-BASED KERNEL APPROACH

[1]ANASS RAIHANI, [2]NABIL LAACHFOUBI

[1,2] Computer, Networks, Mobility and Modeling Laboratory, FST, Hassan 1st University, Settat, MOROCCO

E-mail: [1]an.raihani@uhp.ac.ma, [2]n.laachfoubi@hotmail.fr

ABSTRACT

Discovering unknown drug interactions is of great importance for healthcare professionals since these interactions can become extremely dangerous and can affect patient's safety. Since newly discovered drug interactions are reported in scientific papers, developing text mining techniques to automatically extract those interactions from unstructured texts is of great importance. All state-of-the-art systems evaluated on the standard DDIExtraction 2013 challenge corpus didn't exceed the threshold of 70%, which means that developing more powerful systems to manage this task still very important. In this paper we present a new feature-based kernel method to extract and classify drug interactions described in biomedical literature. Like many previous works, our method consists of two steps. First we detect interacting drug pairs, and then we classify each extracted pair into one of four interaction categories. To perform the first step, we have enhanced an existing feature-based system by adding new features, correction patterns, and trigger words. To perform the second step, we have built a new feature-based kernel classifier that exploit the lexical field particularity of each interaction type. This classifier is composed of 4 binary classifiers work sequentially. When evaluated on the DDIExtraction 2013 challenge corpus, our system achieved an F1-score of 71.14%, as compared to 69.75% and 68.4% reported by the top two state-of-the-art systems based respectively on Convolutional Neural Networks and graph kernel with context vectors methods.

Keywords: *Drug–drug interaction, Biomedical literature, Feature-based kernel approach , Biomedical Informatics , Natural Language Processing*

1. INTRODUCTION

"A drug-drug interaction is a modification of the effect of a drug when administered with another drug" [1]. Detecting drug-drug interactions (DDIs) and their effects is essential to avoid undesirable drug reactions and to ensure the safety of medical prescriptions. For example, a report shows that deaths caused by unexpected DDIs rose between 1999 and 2004 by 68% [2]. Many websites offer on-line services to check possible drug Interactions [3]. Those services use DDIs databases which can be populated automatically or by experts. But the enormous amount of documents describing DDIs makes populating those databases by hand expensive and time consuming. On the other hand, new drug interactions, discovered by professionals, are always reported in new scientific publications and technical reports. MEDLINE [4] size, for example, has grown by 51% between 2007 and 2016, and now contains more than 24 million documents. This means that regular updates of DDIs databases are indispensable. Therefore, developing text mining techniques to automatically extract DDIs from unstructured text has become an urgent need for health care professionals.

Segura-Bedmar et al [5] used pattern matching and linguistic constructions resolution to build the first DDI extraction system. After the organization of DDIExtraction challenges in 2011 and 2013 [6,7] several machine learning approaches have been proposed. In the DDIExtraction 2011 challenge corpus, no information about the types of interactions is provided, while, in DDIExtraction 2013 challenge corpus [8], true interactions are classified into the following interaction types: "Mechanism", "Effect", "Advice", and "Int". In both challenges, systems that use machine learning approaches achieved the best results.

The top performing system [9] on the DDI2013 corpus uses Convolutional Neural Networks (CNN) with Word embeddings and position embeddings to detect and classify DDIs. This system achieves an F1-score of 69.75%. Kernel methods are also exploited to perform the detection and classification task. For example, the best kernel system [10] uses context vectors with a graph kernel to achieve 68.4% F1-score. But what was surprising is that Kim et al [11], by using only a rich feature-based linear kernel, builds the third best system and perform 67% F1-score. These encouraging results lead us to suppose that by using novel features, new improvement in performance can be achieved especially if combined with a non linear kernel.

In this paper we describe a new feature-based kernel system where interacting drug pairs are identified first, and then classified into a specific DDI type. We believe that in many cases, enhancing an existing system can be more advantageous than entirely build a new one. Bui et al [12] developed a feature-based kernel system that performs only the first step. This system was tested separately on DrugBank [13] and MEDLINE [4] documents of the DDI2013 corpus each time with different parameters and training data, but it never be tested entirely on DDI2013 corpus. After analyzing we found that three types of enhancement can be introduced to this system: adding new features, adding new trigger words, and enhancing preprocessing. So to accomplish the first step, the enhanced system is adopted. For the classification step, we developed a new method that uses 4 binary classifiers in sequence to exploit the lexical field particularity of each DDI type.

When evaluated on the DDIExtraction 2013 challenge corpus (DDI2013 corpus), our system achieved an overall F1-score of 71.14%, which outperforms the current best system [9] by 1.39%.

2. METHODS

In this system we manage the detection and classification task in two step. First we use a binary classifier to detect interacting drug pairs. Then we use a system of 4 successive classifiers to assign predefined interaction types to the extracted pairs. Conforming to the Task 9.2 of the DDIExtraction 2013 challenge [7], we assume that drugs have been annotated, and the interactions are expressed within the boundaries of a single sentence and concern a pair of drugs.

This section is divided in two subsections. In the first one we detail the new enhancements introduced after giving a brief description of the baseline system [12]. In the second subsection we describe methods, features, and kernels used by the new interaction-type classifier to perform the classification step.

2.1. DDIs Detection

2.1.1. Baseline

This section gives a brief description of the baseline system. For more details please see the original work [12].

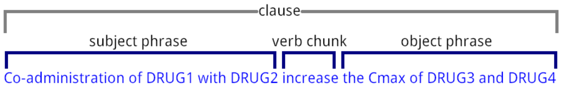
2.1.1.1. *Text preprocessing*

Each drug is renamed as DRUGi where i is the index of the drug. LingPipe NLP toolkit [14] is used to tokenize and tag sentences with POS tags. OpenNLP shallow parser [15] takes the tokens and their tags as input and produces chunks. A list of 298 trigger words has been created. Any sentence that has no trigger word or contains just one drug is filtered out.

2.1.1.2. *Sentence representation and candidate drug pairs partitioning*

Each sentence will be segmented into clauses. Each clause contains one subject phrase, one verb chunk and one object phrase as shown in figure 1. Depending on their positions in the sentence, drug pairs will be classified into one of the following groups:

Subject: If the drug pair belongs to the same subject phrase.
Object: If the drug pair belongs to the same object phrase.
Clause: If the drug pair belongs to the same clause.
Clause_2: If exactly 2 verb chunks exist between the drug pair.
NP: If the input sentence contains only one phrase (does not contain any clause).
Drug pairs separated by more than two verb chunks are filtered out.
5 different classifiers were built (a classifier for each group). Each classifier uses different combination of features.



Co-administration of DRUG1 with DRUG2 increase the Cmax of DRUG3 and DRUG4

Figure 1: Example of a sentence containing one clause, the clause is composed of one subject phrase, one verb chunk and one object phrase.

2.1.1.3. *Features*

The following features are used optionally by the 5 classifiers:

Lexical features: Designed to detect relations between each drug of the candidate pair and its closest tokens.
Phrase features: Are used to describe relations between the drug pair and the trigger words within the boundaries of one phrase.
Verb features: Unigrams and bigrams extracted from the verb chunk of the clause containing the candidate drug pair.
Syntactic features: Describe the syntactic structure around each drug of the candidate pair within the boundaries of one phrase.
Auxiliary features: Describe if the names of the candidate drug pair are real names or pronouns, if they are identical, and if they belongs to the same chunk.

2.1.1.4. *Machine learning*

The LIBSVM [16] classifier is used with RBF kernel. For each candidate drug pair all individual features generated are normalized and added to a single vector as proposed by Miwa et al. (2009).

2.1.2. Enhancements and new features

In this section, we present the enhancements introduced to the preprocessing step and to the trigger words list, thereafter new features are presented.

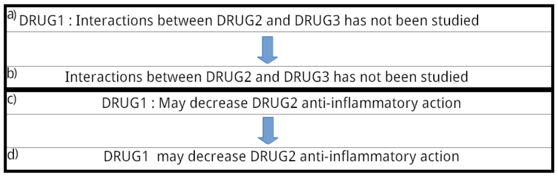
2.1.2.1. *Preprocessing*

Kim et al [11] reported that concatenating the title with the next sentence is a serious problem in the DDI2013 corpus. This problem can prevent systems from correctly identifying sentences structure. For example the baseline system failed to recognize that the drug pair in the figure 2.c belongs to the Clause group. To alleviate this problem the following rules will be implemented:

1) Titles composed of just one drug are removed.

2) If the two tokens after the title match a set of patterns, the title will not be removed, but the two points between the title and the sentence will be removed, and the first word before the title will be converted to lowercase.

These patterns use 3 trigger words lists to check if the title is related syntactically to the next sentence. This correction module will be added before the baseline preprocessing step. The addition of this module help removing negative DDI instances (Like DRUG1-DRUG2 pair in figure 2.a), and identifying correctly the sentence structure. Figure 2.c and figure 2.d show an example of title correction. The sentence in the figure 2.d is classified correctly on Clause category after the addition of the correction module.



DRUG1 : Interactions between DRUG2 and DRUG3 has not been studied

Interactions between DRUG2 and DRUG3 has not been studied

DRUG1 : May decrease DRUG2 anti-inflammatory action

DRUG1 may decrease DRUG2 anti-inflammatory action

Figure 2: Examples of title corrections. The title in the first sentence will be removed while it will be kept in the third sentence.

Finally the Stanford parser [17, 18] is used to generate constituent parse trees and dependency graphs for all sentences. We will use these graphs to generate new features.

2.1.2.2. *Trigger words*

As previously described, the baseline system uses trigger words to filter out none informative sentences and to generate features.

After analyzing the list of trigger words, we found that many important trigger words are not included. For example this sentence: {The mixing of DRUG1 with an DRUG2 in vitro can result in substantial inactivation of the DRUG3} is removed in the filtering phase because the trigger word “inactivation” doesn't exist in the list. To handle this problem a set of 203 new trigger words has been added after analyzing the training data and the old list. This increases the number of trigger words to 501.

2.1.2.3. *New features*

The following features are added optionally to the set of features used by each classifier. Table 1 shows the optimal combination of the new features for each group based on 10-fold cross-validation results over training data.

Lexical features: In the baseline system, lexical features are limited within the boundaries of 3 chunks before and 3 chunks after each drug. Furthermore the tokens between the drug pair are not well exploited. To address this shortfall, we used unigrams and bigrams of lemmatized tokens as features. Similar to the works of He et al [19] and Kim et al [11], the position information, appended to each lemmatized token, depends on its relative position to the drug pair: before (bf), between (be) or after (af). For example, unigram features generated from the sentence in figure 2.b are: [interaction_bf, between_bf, and_be, have_af, not_af, be_af, study_af]. All tokens in the sentence are considered, but only for NP group, the window size will be defined as 4 tokens before the first drug and 4 tokens after the second drug because this window size shows better performance.

NEGATIVE_SENTENCE feature: Dependency graph generated by the Stanford parser [17, 18] uses nodes to represent words and their position in the sentence (e.g. in Figure 3, DRUG2-6 represents the word DRUG2 in the sixth position), and edges to describe governor-dependent relations between the words. We iterate over all dependent words in the dependences list of each sentence, and If any dependent word belongs to a trigger words list, developed especially to generate this feature, and the type of relation between this dependent word and it's governor is negation (neg), the “*NEGATIVE_SENTENCE*” feature will be added to the vector of features. For example, in the figure 3, the list of dependences contains a negative dependency (neg(interact-4, n't-3)), and the dependent word "interact" belongs to the trigger words list, so the “*NEGATIVE_SENTENCE*” feature will be generated.

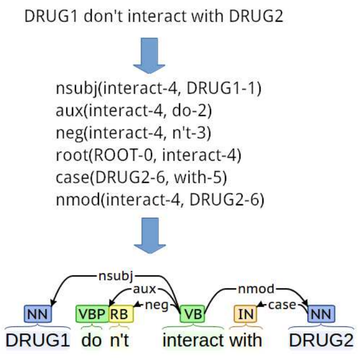


*Figure 3: An example of a dependency graph.**

COORDINATED_DRUGS feature: For each sentence we remove all dependences except those who have a drug as dependent and a drug as governor (like: “conj(DRUG2-6, DRUG3-8)). We will call the graph described by the remaining dependences the COORDINATED_DRUGS graph. If the two drugs of the candidate drug pair are related in the COORDINATED_DRUGS graph (i.e., a path exist between the two candidate drugs in this graph), the "*CORDINATED_DRUGS"* feature will be added to the vector of features. For example, in this sentence: {Since DRUG1 is a DRUG2, it is possible that DRUG3 diminish his effectiveness}, The nsubj(DRUG1-2, DRUG2-5) dependency exists between DRUG1 and DRUG2, so the "*CORDINATED_DRUGS"* feature will be generated.

Grammatical tags count features: Combining information from both sub-sequential and graph

*This graph has been generated by the service available at: http://nlp.stanford.edu:8080/corenlp/process.

Table 1: The optimal combination of the new features for each group based on 10-fold cross-validation results over training data.

Group	Lexical	NEGATIVE_SENTENCE	COORDINATED_DRUGS	Grammatical tags count	SAME_BLOK
Subject	x	x			x
Object		x			
Clause	x	x	x		
Clause_2	x	x	x	x	
NP	x		x	x	

representations improves the overall performance [19]. To exploit a part of information given by the parse tree, we extract the shortest path connecting the drug pair in the parse tree, then we count the number of appearances of each grammatical tag, this number will be appended to the corresponding tag then added as a feature. For example the features extracted from this path ‘‘NP VP S VP NP VP S VP PP NP’’ are: [NP_3, VP_4, S_2, PP_1].

SAME_BLOK feature: Two drugs belong to the same block if they appear in the same coordinate structure. For example, All drugs in the following sentence except DRUG7 belong to the same block: {DRUG1 such as DRUG2 (e.g., DRUG3), DRUG4 such as the DRUG5 or DRUG6 can interact with DRUG7}.It can be very helpful for the machine learning to now if the candidate drugs belong to the some block, so we have developed a rule-based module to decide if the drug pair appears in the same block, if so, we add the “SAME_BLOK” feature to the vector of features.

2.1.2.4. *Parameters selection*

The RBF kernel uses two parameters: C and gamma. We apply a grid-search to find the best parameters using the 10-fold cross-validation technique over the training data.

2.2. DDI Classification

2.2.1. Classifiers architecture

The objective of the classification phase is to classify drug pairs into 4 classes: Advice, Mechanism, Effect, and Int.
Advise: If an advice or recommendation concerning the DDI is given.
Mechanism: If the interaction is described by the pharmacokinetic mechanism.
Effect: If the effect of the interaction is described.
Int: If the sentence doesn't provide any information about the type of the DDI.

By analyzing the data, one can notice that interactions of Advice class are described by a special lexical field comparing to interactions of other classes. For this reason we have built a binary classifier (Classifier_A) that separates interactions of Advice class (output1) from interactions of other classes (output2). Along the same lines, interactions of Mechanism class are described by a special lexical field comparing to interactions of Int and Effect classes, so we have built a second binary classifier (Classifier_B) specialized on separating interactions of Mechanism class (output1) from interactions of Effect and Int classes (output2). The input of this classifier is the output2 of the Classifier_A. The third classifier (Classifier_C) separates interactions of Effect class (output1) from interactions of Int class (output2). The Classifier_A+Classifier_B+Classifier_C system will be called System_1 in this paper.

- *Int class recall*

By analyzing the results reported by the System_1 (see section 3.4), we noticed that all classes get good results except the Int class. The main cause behind the poor results recorded for the Int class is the low recall. To alleviate this problem, we decided to add a fourth classifier (Classifier_D) with a very high precision (but low recall) before the System_1 to separate some DDIs of Int class (output1) from the other DDIs. The output2 of this classifier is now the input of the System_1.

The Classifier_D prevents a set of drug pairs that belong to Int class from entering into the System_1, and thus from being classified on a wrong class. These pairs will be classified directly as Int class. The Classifier_D+System_1 system will be called System_2 in this paper.

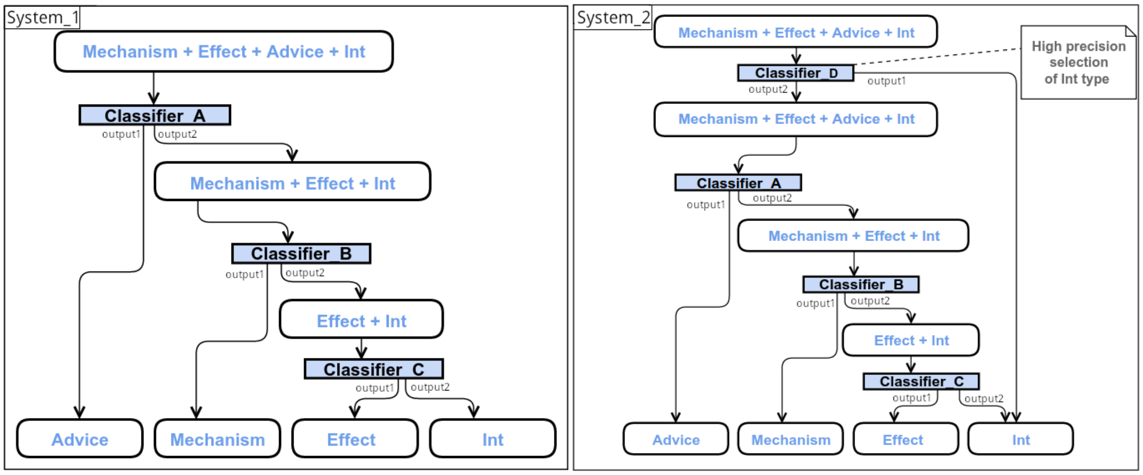


Figure 4: Architectures of System_1 and System_2 developed to perform the classification step.

Figure 4 shows the architectures of the new classifiers described in this section.

- *A classifier for comparison*

Kim et al [11] shown that the one against one strategy give better performance than the one against all strategy in DDIs classification. So, by way of comparison with our new classifiers, we built a multi-classifier that uses the one against one strategy. This classifier will be called System_3. Comparison details are presented in the section 3.4.

2.2.2. Preprocessing

We use the same preprocessing steps used by the baseline DDI detection system (see section 2.1.1).

2.2.3. Features

Lexical features, such as unigrams and bigrams, have been used successfully to extract relational knowledge [11, 19]. Their direct exploitation of the lexical field makes of them strong indicators for the machine learning. Our system exploits the lexical level using unigrams and bigrams combined with position information.

Beside the lexical features, we have tested a wide set of features like dependency features and parse tree features, but experiments in the training data have been shown that using lexical features alone gives the best performance in the classification task. Next we present the lexical features used by each classifier.

2.2.3.1. *Features used by all classifiers except Classifier_D*

Unigrams of lemmatized tokens: All tokens of the sentence except the interacting drugs tokens will be lemmatized then added as features. For example the features generated from the sentence in the figure 2.b are: [interaction, between, and, have, not, be, study].

Bigrams of lemmatized tokens with position information: All tokens of the sentence will be lemmatized then concatenated as bigrams. Similar to the works of He et al [19] and Kim et al [11], the position information, appended to each lemmatized token, depends on its relative position to the drug pair: before (bf), between (be) or after (af). For example features generated from the sentence in the figure 2.b are: [interaction_bf_between_bf, between_bf_DRUG, DRUG_and_be, and_be_DRUG, DRUG_have_af, have_af_not_af, not_af_be_af, be_af_study_af].

2.2.3.2. *Features used by the Classifier_D*

Classifier_D uses only unigrams of lemmatized tokens between the drug pair as features because they show a very high precision in experiments when they are used alone. For example the features generated from the sentence in the figure 1.d are: [may, decrease].

2.2.4. Machine learning

Our system use LIBSVM, the popular SVM library [16]. Two kernels are used optionally by the classifiers composing this system, radial basis function (RBF) and linear kernel. For the RBF kernel, two parameters need to be selected, C and gamma, while only C must be selected for the linear kernel. We apply a grid-search to find the best parameters and the best kernel for each classifier using the 10-fold cross-validation technique over the training data. The Classifier_A uses binary classification with RBF kernel, while Classifier_B, Classifier_C and Classifier_D use binary classification with linear kernel. The System_3 uses multi-classification with RBF kernel.

3. RESULTS AND DISCUSSION

3.1. Dataset

We use the corpus from the DDIExtraction 2013 challenge [7] to evaluate our system. This corpus contains 905 annotated documents from the DrugBank [13] database and the MEDLINE [4] abstracts. Drug pairs are annotated as interacted or not interacted. True interactions are classified into the following four classes: Mechanism, Effect, Advice, and Int (For definitions see the section 2.2.1). Table 2 shows statistics of training and test data before and after preprocessing and filtering steps. Removed negative pairs represent 38.05% of the negative set while removed positive pairs represent only 3.42% of the positive set. The statistics of DDI types are shown in Table 3.

For the classification subtask, all negative pairs are removed from the training data, then positive pairs are used to build a special training data for each classifier as follow:
Classifier_A: class 1=DDIs of Advice type, class 2=DDIs of Mechanism+Effect+Int types.
Classifier_B: class 1=DDIs of Mechanism type, class 2=DDIs of Effect+Int types (DDIs of Advice type are removed).
Classifier_C: class 1= DDIs of Effect type, class 2=DDIs of Int type (DDIs of Advice and Mechanism types are removed).
Classifier_D: class 1= DDIs of Int type, class 2=DDIs of Advice+Mechanism+Effect types.
System_3: A class for each type.

3.2. Performance Comparison

In this section, we compare our best system, which is the enhanced DDI detection system+System_2 (see sections 2.1.2 and 2.2.1), with the state-of-the-art systems.

Our feature-based kernel system shows the best performance when compared with state-of-the-art systems. It outperforms the current best system [9] by 1.39% F1-score.

Table 4 compares our system with the best three systems based on the standard F1-score

Table 2: Statistics of training and test data before and after preprocessing and filtering.

	Original set		Preprocessed set	
	Positive	Negative	Positive	Negative
training	4020	23772	3878	14369
test	979	4737	950	3290
all	4999	28509	4828	17659

Table 3: Statistics of DDI types in training and test data.

	Advice	Mechanism	Effect	Int
training	826	1319	1687	188
test	221	302	360	96

Table 4: Performance comparison between our system and the top-ranking systems on the DDI2013 test data. The standard F1-score evaluation measure is used as unit (%). 'CLA' refers to detection and classification performance. 'DEC' refers to detection performance.

Method	CLA	DEC	Mechanism	Effect	Advice	Int
Our method	**71.14**	81.5	**73.57**	69.55	77.43	**52.35**
Shengyu Liu et al [9]	69.75	-	70.24	69.33	**77.75**	46.38
Zheng et al [10]	68.4	**81.8**	66.9	**71.3**	71.4	51.6
Kim et al [11]	67	77.5	69.3	66.2	72.5	48.3
Baseline [12]	-	79.4	-	-	-	-

evaluation measure. Our system achieves 71.14% F1-score for detection and classification performance ('CLA'), whereas Shengyu Liu et al [9], Zheng et al [10] and Kim et al [11] produced 69.75%, 68.4% and 67% F1-score respectively. For DDIs detection ('DEC') the current best system [10] achieved 81.8% F1-score while our system gets comparable results by achieving 81.5% F1-score.

Beside top performing systems, Table 4 shows the performance of the baseline system for two reasons, first as reference to measure the improvement performed by the new system, and second because the baseline system had never been tested on the entire DDIExtraction 2013 challenge corpus.

Shengyu Liu et al [9] uses Convolutional Neural Networks with word embeddings and position embeddings to perform the detection and classification task after filtering a set of negative instances by relying on a range of criteria. Zheng et al [10] uses context vectors with a graph kernel to detect and classify DDIs. He exploits different types of contexts and relations among words with different distances. To perform the classification subtask he uses the one-against-all strategy. Kim et al [11] uses linear kernel with a binary classification to identify DDIs, and uses the one-against-one strategy to assign types to the extracted pairs. This strategy is used to handle the bad effect of unbalanced classes.

On the other side, our method uses a binary SVM classifier with RBF kernel for identifying DDIs and 4 binary SVM classifiers in cascade to assign DDI types. We use those classifiers to exploit the lexical field particularity of each type. Our approach gets its best performance for advice, mechanism and effect types, while it does not perform well for int. This may be due the small number of training and test data for this type (188 and 96 instances for training and test data respectively).

The results in Table 5 show that our system performs well on the DrugBank test documents with F1-score of 73.95%. However, its performance decreases on the MEDLINE test documents to 43.02%. Similar difference in performance was reported by the teams that participated in the DDIExtraction 2013 challenge [7] and by state-of-the-art systems[9, 10, 11]. One issue that might affects performance on the MEDLINE test set is its size, which is significantly smaller than DrugBank test set. Another reason can be the scientific language of the MEDLINE documents, which use more complex sentences to describe relations.

3.3. Enhancements Analysis

Table 6 presents improvement of DDI detection performance when adding enhancements one by one to the baseline system.10-fold cross validation was performed over the training data to obtain the results.

Title corrections module improves the F1 performance by 0.59%. This improvement is understandable because this module removes negative DDI instances and helps the system to correctly recognize the sentence structure. New trigger words in their side improve the performance by 0.72% F1. This shows that our observation about the insufficient of the baseline trigger words list was correct. For the new features, Lexical and NEGATIVE_SENTENCE features contribute the most by increasing F1-score by 0.5% and 0.51% respectively. The rest of features have less impact on performance and their total contribution on improvement is 0.5%.

The improvement recorded by lexical features shows that the baseline system doesn't benefited sufficiently from the lexical level, and confirms the results reported by Kim et al [11] about the importance of attaching position information to the words. In the other hand, NEGATIVE_SENTENCE feature helps the system to differentiate between positive DDIs and negative DDIs expressed by negation. Since this feature is generated using grammatical dependences, we think that these dependences can be source of other powerful features.

Table 5: Comparison between performance recorded on DrugBank and MEDLINE test sets. The standard F1-score evaluation measure is used as unit (%). 'CLA' refers to detection and classification performance. 'DEC' refers to detection performance.

Dataset	CLA	DEC	Mechanism	Effect	Advice	Int
DrugBank	**73.95**	**84.07**	**76.34**	**73.54**	**77.94**	**55.07**
MEDLINE	43.02	55.81	33.33	46.43	61.54	18.18

Table 6: Improvement of DDI detection performance when adding enhancements one by one to the baseline system. 'Improvement' column shows the F1-score difference between each row and its previous row. The last row shows the total improvement. 10-fold cross validation was performed over the training data to get the scores.

Enhancements and new features	Precision	Recall	F1	Improvement
Baseline	0.7809	0.7077	0.7425	-
Baseline + title corrections(TC)	0.7773	0.7216	0.7484	+0.59%
Baseline + TC + new trigger words	0.7737	0.7383	0.7556	+0.72%
+New lexical features	0.8088	0.7179	0.7606	+0.5%
+ NEGATIVE_SENTENCE feature	0.8144	0.7226	0.7657	+0.51%
+ Grammatical tags count feature	0.8157	0.7268	0.7687	+0.3%
+ *CORDINATED_DRUGS* feature	0.8112	0.7313	0.7692	+0.05%
+ SAME_BLOK feature	0.8119	0.7335	0.7707	+0.15%
Enhanced DDI detection system	0.8119	0.7335	0.7707	+2.82%

Table7: Results reported by the three classification systems on the test corpus for the detection and classification task when they are added to the enhanced DDI detection system (EDDIDS).

System	Precision	Recall	F1
EDDIDS+System_1	0.7317	0.6823	0.7061
EDDIDS+System_2	**0.7371**	**0.6874**	**0.7114**
EDDIDS+System_3	0.7273	0.6782	0.7019

3.4. Classifiers Analysis

Table 7 shows the results reported by the three classification systems (described in the section 2.2.1) on the test corpus for the detection and classification task when they are added to the enhanced DDI detection system (EDDIDS).

The EDDIDS+System_2 gets the best performance by achieving 71.14% F1 in the detection and classification task, while EDDIDS+System_1 and EDDIDS+System_3 ranked second and third by achieving respectively 70.61% F1 and 70.19% F1.

Table 8 details why the EDDIDS+System_2 achieves better performance. Data in this table shows that adding the Classifier_D (see section 2.2.1) improves the recall of Int type by 5.2% and the F1-score of the same type by 4.46% which lead to get the best performance for this type by recording 52.35% F1. The EDDIDS+System_2 achieves also the best performance for Mechanism and Effect types, this is due to preventing a set of drug pairs that belong to Int type to be wrongly classified as Mechanism or Effect type. This is why only precisions of those two types show an improvement while recalls maintain the same values.

3.5. Erroneous Data and Title Section Problem

During system development, we found that the training corpus contains erroneous data. In fact, some true interactions are wrongly annotated as false and vice versa. Figure 5.a shows a true relation between 'Simvastatin' and 'Amiodarone' drugs that was wrongly annotated as negative instance (See Amiodarone_ddi.xml in DrugBank training data sentence id :DDI-DrugBank.d143.s21). SVM classifiers can manage erroneous training data to some degree, but if such data are present on the test corpus, systems will be wrongly evaluated.

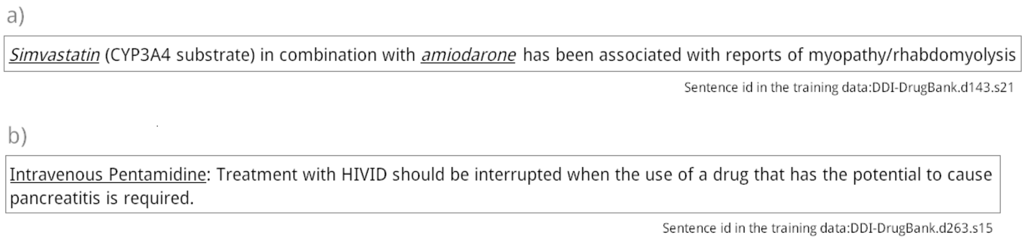
Title section can be another source of noise to systems that require a grammatically well-formed text. In our case if the sentence is incorrectly segmented, drug pairs may be classified on false syntactic groups, thus, inappropriate features will be generated. Even a module was developed to handle some title problems, others still with no solution. For example Figure 4.b shows an instance where the title can't be deleted by the title correction module.

4. CONCLUSION

In this paper, we have proposed a feature-based kernel method with SVM classification to extract DDIs from biomedical literature. New features and enhancements are introduced to a feature-based kernel system before use it to perform the detection step. For the classification step, we have built a new classifier that exploits the lexical field particularity of each DDI type. When compared to a one-against-one strategy classifier, our new classifier gets better results. Applied to the DDIExtraction 2013 challenge corpus, the proposed method gets the best performance when compared to the state-of-the-art systems by achieving 71.14% F1.

Table 8: The results reported on the test corpus by the three classification systems when they are added to the enhanced DDI detection system (EDDIDS). 'ADV', 'MEC', 'EFF', and 'INT' are for Advice, Mechanism, Effect, and Int types respectively.

System	ADV			MEC			EFF			INT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
EDDIS+System_1	**0.815**	0.737	**0.774**	0.786	0.682	0.730	0.654	**0.736**	0.692	0.739	0.354	0.478
EDDIS+System_2	**0.815**	0.737	**0.774**	**0.798**	0.682	**0.735**	0.659	**0.736**	**0.695**	0.735	**0.406**	**0.523**
EDDIS+System_3	0.79	**0.751**	0.77	0.742	**0.705**	0.723	**0.672**	0.697	0.684	**0.790**	0.354	0.489



a)

Simvastatin (CYP3A4 substrate) in combination with amiodarone has been associated with reports of myopathy/rhabdomyolysis

Sentence id in the training data:DDI-DrugBank.d143.s21

b)

Intravenous Pentamidine: Treatment with HIVID should be interrupted when the use of a drug that has the potential to cause pancreatitis is required.

Sentence id in the training data:DDI-DrugBank.d263.s15

Figure 5: The first example shows a wrongly annotated instance in the training data (drugs are underlined) while the second example presents a sentence concatenated with the title section (the title section is underlined).

The results show that our new architecture, developed to exploits the particularity of each group, gives the best performance. This technique can be used to manage similar classification problems like protein-protein interaction classification or other classification problems. The results show also that using new features and correction patterns improves the performance of DDI extraction, and that using a complete set of trigger words is crucial to get good results.

Authors of the DDI2013 corpus assumed that interactions are expressed within the boundaries of a single sentence and concern one drug pair. This assumption represents a limitation because interactions can concern more than two drugs and can be defined by multiple sentences. Another limitation is that our current system gets drug entities from external source. We think that completing this system by a module to automatically detect drugs entities is of great importance.

Shengyu Liu et al [9] achieved the second best results by using Convolutional Neural Networks (CNN) to perform the detection and the classification in one step. A previous study [20] has shown that performing this task in two steps (detection then classification) gives better results. We think that building a CNN based method that takes two steps to conduct the task can gives better results. We think also that developing a module to handle the badly formatted sentences may be another source of improvement.

ACKNOWLEDGMENTS

The authors thank Quoc-Chinh Bui, Peter M.A. Sloot, Erik M. van Mulligen and Jan A. Kors for making their system available for academic use. We have take advantage of the source code of their system available at http://www.biosemantics.org/uploads/DDI.zip.

REFERENCES

[1] Drug-Drug interaction. (n.d.) *Mosby's Medical Dictionary, 8th edition*. (2009). Retrieved April 13 2016 from http://medical-dictionary.thefreedictionary.com/drug-drug+interaction.

[2] JW Payne.2007. A Dangerous Mix [Published in The Washington Feb 27 2007] washingtonpost.com/wp-dyn/content/article/2007/02/23/AR2007022301780.html.

[3] http://umm.edu/health/medical/drug-interaction-tool [accessed June 20 2016].

[4] MEDLINE, https://www.nlm.nih.gov/bsd/pmresources.html [accessed June 20 2016].

[5] I. Segura-Bedmar, P. Mart´ınez, and C. de Pablo-S´anchez, "A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents," BMC Bioinformatics, vol. 12,supplement 2, no. 1, 2011.

[6] I. Segura-Bedmar, P. Mart´ınez, and D. S´anchez-Cisneros, "The1st DDIExtraction-2011 challenge task: extraction of drug-drug interactions from biomedical texts ," in Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction (DDIExtraction '11), pp. 1–9, Huelva, Spain, September 2011.

[7] I. Segura-Bedmar, P. Mart´ınez, and M. Herrero-Zazo,"SemEval-2013 task 9: extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval '13), pp. 341–350, Atlanta, Ga, USA, June 2013.

[8] M. Herrero-Zazo, I. Segura-Bedmar, P. Martnez, T. Declerck, The DDI corpus: an annotated corpus with pharmacological substances and drugdrug interactions,J. Biomed. Inform. 46 (5) (2013) 914–920. http://dx.doi.org/10.1016/j.jbi.2013.07.011.

[9] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang, "Drug-Drug Interaction Extraction via Convolutional Neural Networks," Computational and Mathematical Methods in Medicine, vol. 2016, Article ID 6918381, 8 pages, 2016. doi:10.1155/2016/6918381.

[10] W. Zheng ,H. Lin ,Z. Zhao ,B. Xu ,Y. Zhang ,Z. Yang ,J. Wang , "A Graph Kernel Based on Context Vectors for Extracting Drug-Drug Interactions," Journal of Biomedical Informatics, Volume 61, June 2016, Pages 34-43.

[11] S. Kim, H. Liu, L. Yeganova, and W. J. Wilbur, "Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach," Journal of Biomedical Informatics, vol.55, pp. 23–30, 2015.

[12] Q. Bui, P. M. Sloot, E. M. van Mulligen, and J. A. Kors, "A novel feature-based approach to extract drug-drug interactions from biomedical text," Bioinformatics, vol. 30, no. 23, pp. 3365–3371, 2014.

[13] V. Law, C. Knox, Y. Djoumbou et al., "DrugBank 4.0: shedding new light on drug metabolism," Nucleic Acids Research, vol. 42,no. 1, pp. D1091–D1097, 2014.

[14] alias-i: LingPipe, a tool kit for processing text using computational linguistics. http://alias-i.com/lingpipe/ [accessed June 20 2016].

[15] apache: OpenNLP,a machine learning based toolkit for the processing of natural language text. https://opennlp.apache.org/ [accessed June 20 2016].

[16] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm [accessed June 20 2016].

[17] Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. Proceedings of ACL 2013.

[18] Danqi Chen and Christopher D Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. Proceedings of EMNLP 2014.

[19] L. He, Z. Yang, Z. Zhao, H. Lin, Y. Li, Extracting drug–drug interaction from the biomedical literature using a stacked generalization-based approach, PLoS One 8 (6) (2013) e65814.

[20] A. Bokharaeian, B.and D´ıaz. 2013. NIL UCM: Extracting Drug-Drug interactions from text through combination of sequence and tree kernels. In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013).