A Rich Feature-based Kernel Approach for Drug-Drug Interaction Extraction

ANASS RAIHANI
Computer, Networks, Mobility and Modeling Laboratory
FST, Hassan 1st University
Settat, MOROCCO

NABIL LAACHFOUBI
Computer, Networks, Mobility and Modeling Laboratory
FST, Hassan 1st University
Settat, MOROCCO

***Abstract*—Discovering drug-drug interactions (DDIs) is a crucial issue for both patient safety and health care cost control. Developing text mining techniques for identifying DDIs has attracted a great deal of attention in the last few years. Unfortunately, state-of-the-art results didn't exceed the threshold of 0.7 F1 score, which calls for more efforts. In this work, we propose a new feature-based kernel method to extract and classify DDIs. Our approach consists of two steps: identifying DDIs and assigning one of four different DDI types to the predicted drug pairs. We demonstrate that by using new groups of features non-linear kernels can achieve the best performance. When evaluated on the DDIExtraction 2013 challenge corpus, our system achieved an F1-score of 71.79%, as compared to 69.75% and 68.4% reported by the top two state-of-the-art systems.**

Keywords—Drug–drug interaction; Feature-based approach; Nonlinear kernel; Biomedical informatics; Natural Language Processing

I. Introduction

"A drug-drug interaction is a modification of the effect of a drug when administered with another drug" [1]. Unexpected side effects of DDIs are generally dangerous and can lead to deaths. Understanding these DDIs and their side effects is of great importance, leading to reduced healthcare costs and reduced number of drug-safety incidents. New DDIs are always reported in new scientific publications and technical reports, but extracting those DDIs by hand is expensive and time consuming. Therefore, automatic DDI extraction, which detects DDIs in unstructured text and classifies them into predefined categories, has become an urgent need in medical text mining.

DDI extraction has attracted a special attention in the last few years. With the organization of DDIExtraction challenges in 2011 and 2013 [2,3] and the creation of the DDIExtraction 2013 challenge corpus [4] several approaches to manage this task have been proposed. Zheng et al [5] used context vectors with a graph kernel to build the second best system. There system achieved an F1-score of 68.4%. Convolutional Neural Networks (CNN) have been used by [6] to build the top performing system. Word embeddings and position embeddings are used to represent DDI instances. This system gets 69.75% in F1-score and outperforms the graph kernel system by 1.35%. Feature-based linear kernel has been used by [7] to build a simple system that uses few types of features. The results was encouraging and the system achieved 67% F1-score. We think that the DDI extracting task can't be solved only by a linear kernel because of the high complexity of the task. Non-linear Feature-based kernel can be more powerful to perform this task especially if combined with intelligently chosen features.

With the goal to build an intelligent and powerful system, we develop a DDI extraction system based on a non-linear SVM classifier. Interacting drugs are identified first, and then classified into specific DDI types. We define five types of features to represent the complexity of data: "word features" with position information, "one-drug features" to represent features related to each drug, "pair features" to represent features related to the drug pair, "main-verb features" to represent features related to the main verb of the sentence, and finally the negation features. This system separates candidate drug pairs into five groups based on their syntactic structures then features are optimized for each group.

When evaluated on the standard corpus [4], our system achieved an overall F1-score of 71.79%, which outperforms the current best system by 2%. We believe that the strength of our method comes from combining intelligent features with non-linear kernel. In addition, the cascade strategy [8], used to perform the classification, contributes to the higher performance.

In this section, we describe our method for extracting drug–drug interactions from biomedical texts. Fig 1 illustrates the general architecture of our system. Candidate drug pairs are separated into five groups based on their syntactic structures. For each group a binary classifier is trained to extract interacting drug pairs. Extracted drug pairs are then grouped before being classified into predefined relation categories by a DDI type classifier.

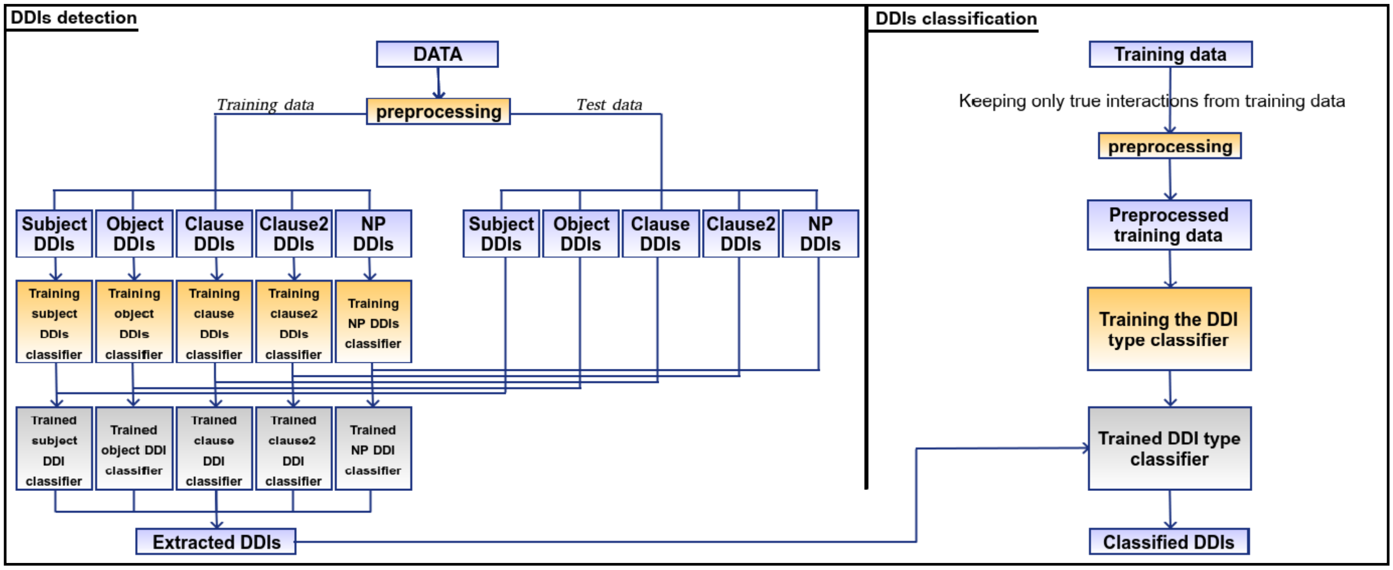

DDIs detection
DATA
Training data
preprocessing
Test data
Subject DDIs
Object DDIs
Clause DDIs
Clause2 DDIs
NP DDIs
Training subject DDIs classifier
Training object DDIs classifier
Training clause DDIs classifier
Training clause2 DDIs classifier
Training NP DDIs classifier
Trained subject DDI classifier
Trained object DDI classifier
Trained clause DDI classifier
Trained clause2 DDI classifier
Trained NP DDI classifier
Extracted DDIs
DDIs classification
Training data
Keeping only true interactions from training data
preprocessing
Preprocessed training data
Training the DDI type classifier
Trained DDI type classifier
Classified DDIs

Fig. 1. The general architecture of our system

II. METHODS

A. DDIs Extraction

1) Text preprocessing

Several preprocessing steps are completed on both training and test data. To ensure generalization of the input sentences, drugs are blinded in the following way: the two drugs of interest are replaced by “ARG” while all the other drugs are replaced by “DRUGi” where i is the drug index. Sentences that have no trigger word or contain only one drug are filtered out. We use the same list of trigger words created by [8]. After drug blinding, we use LingPipe NLP toolkit [9] to tokenize and tag sentences with POS tags, OpenNLP shallow parser [10] is then used to produce chunks. Dependency graphs and constituent parse trees are also generated for all sentences using Stanford parser [11, 12].

2) Candidate drug pairs partitioning

In a previous study, [13] demonstrated that partitioning candidate DDI pairs based on their syntactic properties then using specific group of features for each partition improves the performance of the DDI extraction system. Following this strategy, we classify candidate pairs into different groups based on their positions into the sentence. Every sentence will be divided into clauses. Each clause consists of a subject phrase, a verb chunk and an object phrase as shown in Fig 2.

Candidate DDI pairs will be classified into one of the following groups based on their syntactic containers:

Subject: If the candidate DDI pair belongs to the same subject phrase.

Object: If the candidate DDI pair belongs to the same object phrase.

Clause: If the candidate DDI pair belongs to the same clause.

Clause_2: If the candidate DDI pair is separated by two verb-chunks.

NP: If the sentence contains only one phrase.

We filtered out all candidate DDI pairs that are separated by more than two verb-chunks. We build a classifier for each group. Different combination of features are used for each classifier.

3) Features

In this section we describe all features used by our DDI extraction system. Table I shows the optimal combination of features used by all classifiers built for the 5 groups. For each classifier, the selection of features is based on a 10-fold cross-validation over training data. Fig 4 shows an example of features generated for a DDI pair.

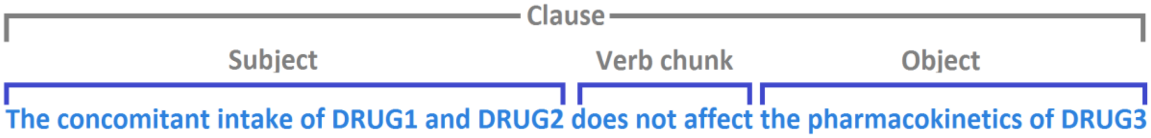

Clause
Subject
Verb chunk
Object
The concomitant intake of DRUG1 and DRUG2 does not affect the pharmacokinetics of DRUG3

Fig. 2. Example of a sentence containing one clause

a) Word-features

In previous studies [14, 15, 16, 7] individual words and sequences of words in a sentence have been used successfully in extracting relational knowledge like protein–protein interactions or drug-drug interactions. Hence, in our system, we use unigrams and bigrams of lemmatized tokens as features. Many studies [14] [7] have shown the importance of appending position information to word features. In our system the position information can takes 3 values:

BF (before): If the concerned word is before the investigated drug pair.

BE (between): If the concerned word is between the investigated drug pair.

AF (after): If the concerned word is after the investigated drug pair.

We use four form of windows to select words for the generation of word-features:

W1: All words of the sentence containing the candidate pair are selected.

W2: All words of the clause containing the candidate pair are selected.

W3: Words between the beginning of the clause containing the candidate pair and the end of the sentence are selected.

W4: words between the beginning of the chunk containing the first drug and the end of the chunk containing the second drug are selected.

For each classifier the best window is selected using a 10 fold cross validation, results are shown in Table I.

b) One-drug features

Are designed to capture relations between each drug of the candidate DDI pair and the phrase to which it belongs. One-drug features comprises three groups of features:

Surrounding words: Words before and after each drug are added as features.

Surrounding triggers: Trigger words that belong to the same phrase of the concerned drug are added as features. We append position information (before or after) and all prepositions between the trigger words and the concerned drug. If no trigger word exist in the phrase containing the drug, then "no_rel_word" is add as feature.

Succeeding drugs: Relations between the concerned drug and the succeeding drugs in the same phrase are added as features. Punctuation, coordination and prepositions are also captured by this group of features. If no drug exist then the "no_other_drug" is added as feature.

c) Pair features

Pair features consist of three groups of features:

Same_chunk features: Detect if the two investigated drugs are within the same chunk.

Between_drugs features: Detect trigger words, connectors (because, since, until ….), prepositions and negations between the two investigated drugs.

Trigger_DRUG_Position features: Like Bui et al [13] we determine the relative position of each trigger word within the phrase by checking the following cases:

- Trigger [prep] DRUG1 [prep] DRUG2
- DRUG1 [prep] trigger [prep] DRUG2
- DRUG1 [prep] DRUG2 [prep] trigger

Where DRUG1 and DRUG2 are drugs of the candidate DDI pair and "prep" are prepositions connecting chunks that contain the trigger word and the drug.

Depending on the obtained case, features are generated to represent:

The position of the trigger relative to the candidate pair (before, between, after).

Prepositions that connect the trigger with the candidate pair.

Prepositions that connect chunks between drugs of the candidate pair.

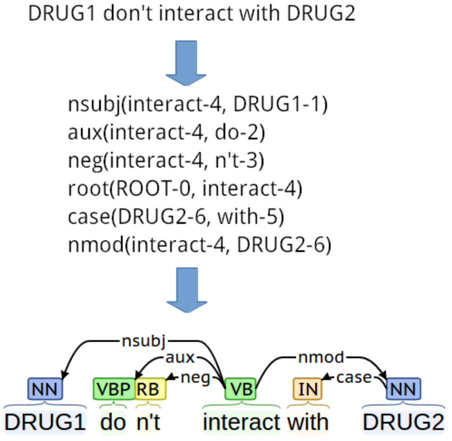
d) Main-verb features

Are designed to indicate how DRUG1 in the subject phrase and DRUG2 in the object phrase are related. To perform this task unigrams, bigrams, negations and trigger-words are extracted from the verb chunk to be used as features. Connectors before the verb chunk and adverb phrase after the verb chunk are also used as features.

e) Negative-sentence feature

In some cases, the sentence deny the existence of a relation between two drugs, and it is important to detect those cases to avoid any miss-classification. For example, adding "Negative-sentence" as feature to the sentence in Fig 2 can be very helpful to avoid the classification of DRUG1 and DRUG2 as interacting drug pair.

To perform this task we generate first the dependency graph of the sentence by the Stanford parser [11, 12]. This graph uses nodes to represent words and edges to describe governor-dependent relations between them. One of the important governor-dependent relations is the negative dependency relation which describe the relation between a negation word and the word it modifies.



DRUG1 don't interact with DRUG2

```
nsubj(interact-4, DRUG1-1)
aux(interact-4, do-2)
neg(interact-4, n't-3)
root(ROOT-0, interact-4)
case(DRUG2-6, with-5)
nmod(interact-4, DRUG2-6)
```

Fig. 3. An example of a dependency graph

For example in the sentence on Fig. 3 the Stanford parser will generate a negative dependency relation between "not" and "interact" [neg(interact-4, not-3)] where "not" is the governor and "interact" is the dependent word.

To exploit the negative dependency relation we have developed a list of trigger words. If the dependent word of any negative dependency relation belongs to this list, the "NEGATIVE_SENTENCE" feature will be added to the vector of features. For example, "NEGATIVE_SENTENCE" feature will be generated for the sentence on Fig 3 because its list of dependences contains a negative dependency and the dependent word ("interact") belongs to the trigger words list.

4) *Machine learning*

For DDI detection, we use LIBSVM, the popular SVM library [17], with a default radial basis function (RBF) kernel. For each candidate pair, individual features generated are normalized and added to a single vector as proposed by [18].

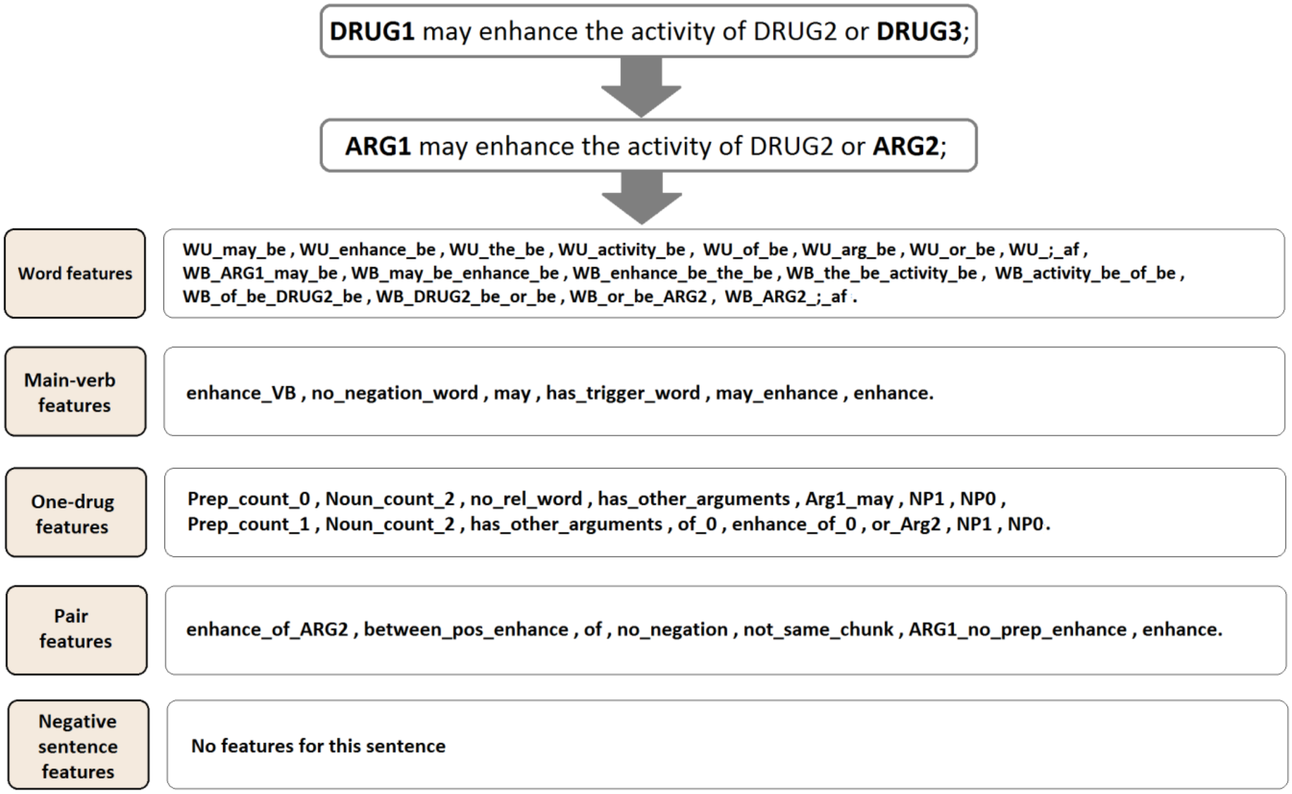

DRUG1 may enhance the activity of DRUG2 or DRUG3;
ARG1 may enhance the activity of DRUG2 or ARG2;
Word features
WU_may_be , WU_enhance_be , WU_the_be , WU_activity_be , WU_of_be , WU_arg_be , WU_or_be , WU_;_af , WB_ARG1_may_be , WB_may_be_enhance_be , WB_enhance_be_the_be , WB_the_be_activity_be , WB_activity_be_of_be , WB_of_be_DRUG2_be , WB_DRUG2_be_or_be , WB_or_be_ARG2 , WB_ARG2_;_af .
Main-verb features
enhance_VB , no_negation_word , may , has_trigger_word , may_enhance , enhance.
One-drug features
Prep_count_0 , Noun_count_2 , no_rel_word , has_other_arguments , Arg1_may , NP1 , NP0 , Prep_count_1 , Noun_count_2 , has_other_arguments , of_0 , enhance_of_0 , or_Arg2 , NP1 , NP0.
Pair features
enhance_of_ARG2 , between_pos_enhance , of , no_negation , not_same_chunk , ARG1_no_prep_enhance , enhance.
Negative sentence features
No features for this sentence

Fig. 4. An example of of feature extraction

TABLE I. THE OPTIMAL COMBINATION OF FEATURES FOR EACH GROUP BASED ON 10-FOLD CROSS-VALIDATION RESULTS OVER TRAINING DATA.

Features			Subject	Object	Clause	Clause2	NP
Word features	***Unigrams***	W1		X	X		X
		W2				X	
		W3	X				
	Bigrams	W1				---	X
		W3	X		X		
		W4		X			
One-drug features	***Surrounding triggers***		X		X	X	
	Succeeding drugs		X		X	X	
	Surrounding words		X	X	X	X	X
Pair features	***Same_Chunk***		X	X	X	X	X
	Between_Drugs		X	X	X	X	X
	Trigger_DRUG_Position		X	X	X		X
Main-verb features			X	X	X	X	X
Negative-sentence features			X	X	X	X	

B. *DDIs Classification*

The objective of the classification task is to assign one of four DDI types (Mechanism, Effect, Advice and Int) to interacting pairs extracted in the detection step. Kim et al (2015) [7] have shown that the one against one strategy gives better performance in DDIs classification comparing to the one against all strategy. Raihani et al (2016) [8] have built a new DDIs classifier that exploits the lexical field particularity of each DDI type. When compared to a one-against-one strategy classifier, the new classifier gets better results. Thus we will use the same classification system developed by [8].

III. Results and Discussion

A. Dataset

To evaluate our system we use the corpus from the DDIExtraction 2013 challenge [3,4]. This corpus includes 905 manually annotated documents from MEDLINE abstracts and DrugBank database, which are split into 714 and 191 documents for training and test sets, respectively. The corpus provides examples by sentences, for each sentence all drug pairs are annotated.

Four types of DDI relationships is used to annotate interacting drug pairs in the set: Mechanism, Effect, Advice, and Int. Mechanism is used if the interaction is described by the pharmacokinetic mechanism. Effect is used when the effect of the interaction is described. Advice is used if an advice or recommendation concerning the DDI is given. Int is used when the sentence doesn't provide any information about the type of the DDI.

Table II shows statistics of training data and test data before and after preprocessing. The removed negative pairs constitute 18.5% of the negative set while the removed positive pairs constitute only 2.3% of the positive set. This portion is negligible compared to the advantage of not showing 18.5% of negative pairs to SVM classifiers.

B. Performance Comparison

Compared with state-of-the-art systems, including the best existing system our feature-based kernel system shows much better performance. It outperforms the current best system [6] by 2%, mainly due to much higher recall.

Table III compares our system with top three performing systems based on F1 scores. Our system achieves 71.79% F1-score for detection and classification performance ("CLA"), whereas [6], [5] and [7] produced 69.75%, 68.4% and 67% F1-score respectively. For DDIs detection performance ('DEC') the current best system [5] achieved 81.8% F1-score while our system gets better results by achieving 82,4% F1-score.

Shengyu Liu et al [6] define a range criteria for filtering out some negative instances, then they use Convolutional Neural Networks with word embeddings and position embeddings, which capture the semantic information of words and relative distances between words and the two drugs of interest, to perform the detection and the classification tasks. Zheng et al [5] apply context vectors to a graph kernel to detect and classify DDIs from biomedical texts. There method focuses on the effective use of types of contexts and relations among words with different distances. In addition, they use the one-against-all strategy to perform the classification task. Kim et al [7] use linear kernel with a simple binary SVM classifier for identifying DDIs and use the one-against-one strategy for assigning DDI types to the extracted pairs. They use the one-against-one strategy to handle the bad effect of unbalanced classes.

On the other side, our method exploits new groups of features and uses a binary SVM classifier with RBF kernel to identify DDIs. In addition we use a system of 4 binary SVM classifiers, work in cascade, to perform the classification task. Previous study [8] has shown that this method gives the best performance in DDIs classification.

In Table III, our system performs best for advice, mechanism and effect types. In contrast, the same system does not perform well for Int. Int type is defined as DDIs which cannot be assigned to other types. We think that the general description of Int or the small number of training and test sets (188 and 96 instances for training and test data respectively) may be the cause of the poor results for this type.

Table IV shows the separate performance of our system on DrugBank and MEDLINE test documents. The DDI detection and classification performance on the DrugBank set shows 74.7% F1, while the performance on the MEDLINE set is substantially lower (44,5% F1). This difference is consistent with the results reported by state-of-the-art systems [5, 6, 7] and the results from the DDIExtraction 2013 challenge [4].

One reason may be the small number of training examples provided for MEDLINE. MEDLINE training set is significantly smaller than DrugBank training set and constitute only 7% of the overall training data. Another reason can be the scientific language used in MEDLINE abstracts, which use long and complex sentences to describe relations. In contrast, sentences used in DrugBank are usually short and concise.

TABLE II. Statistics of Training and Test Data Before and After Preprocessing and Filtering.:

	Original set			**Preprocessed set**	
	Positive	**Negative**		**Positive**	**Negative**
training	4020	23772		3922	19194
test	979	4737		960	4016
all	4999	28509		4882	23210

TABLE III. Performance Comparison between Our System and the Top-Ranking Systems on the DDI2013 Test Data. 'CLA' Refers to Detection and Classification Performance. 'DEC' Refers to Detection Performance. the Performance is Measured based on F1 Scores.

Method	CLA	DEC	Mechanism	Effect	Advice	Int
Our method	**71.79**	**82.46**	**74.26**	69.81	**78.68**	**51.74**
Shengyu Liu et al [6]	69.75	-	70.24	69.33	77.75	46.38
Zheng et al [5]	68.4	81.8	66.9	**71.3**	71.4	51.6
Kim et al [7]	67	77.5	69.3	66.2	72.5	48.3

C. Feature Analysis

Using lexical features without position information as a baseline, position information, one-drug features, main-verb features, pair features, and n*egative-sentence features* are added respectively to the system and evaluated one the test dataset. Table V shows the contribution of each group of features.

Adding the position information to the word-features improves the F1 performance by 4.5%. This significant improvement is understandable because relative position help the system to understand if an individual word is used in describing an interaction between two drugs or not. These results confirms the conclusion of [7] about the importance of attaching position information to the words.

One-drug features, main-verb features, pair features, and negative-sentence features contribute on performance by increasing F1-score by 0.56%, 0.44%, 0.58% and 0.44% respectively. One-drug features, pair features and negative-sentence features help get higher recall while main-verb features help to improve the precision. While one-drug features cover neighbouring words, pair features and main-verb features seem to help with the overall picture of a relation between two drugs into the sentence. It is remarkable that using words with position information alone achieves such high performance for detecting DDIs. Integrating position information into word features helps differentiate the context of interacting pairs from that of non-interacting ones in sentences that involve multiple drug mentions.

TABLE IV. COMPARISON BETWEEN PERFORMANCE RECORDED ON DRUGBANK AND MEDLINE TEST SETS. 'CLA' REFERS TO DETECTION AND CLASSIFICATION PERFORMANCE. 'DEC' REFERS TO DETECTION PERFORMANCE. THE PERFORMANCE IS MEASURED BASED ON F1 SCORES.

Dataset	CLA	DEC	Mechanism	Effect	Advice	Int
DrugBank	**74.73**	**84.88**	**77.78**	**73.44**	**79.33**	**55.07**
MEDLINE	44.57	58.7	40	46.15	61.54	22.22

TABLE V. IMPROVEMENT OF DDIS DETECTION AND CLASSIFICATION PERFORMANCE WHEN ADDING FEATURES 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.

Features	Precision	Recall	F1	Improvement
Baseline (w/o pos)	0.6491	0.6557	0.6524	-
Baseline	0.7358	0.6629	0.6974	+4.50%
+ One-drug features	0.7108	0.6956	0.7031	+0.56%
+ Main-verb features	0.7482	0.6710	0.7075	+0.44%
+ Pair feature	0.7439	0.6853	0.7134	+0.58%
+ *negative sentence* features	0.7360	0.7007	0.7179	+0.44%
All features	0.7360	0.7007	0.7179	+6.55%

IV. CONCLUSION

We present a two-step classification approach to extract DDIs from biomedical literature. Interacting drug pairs are first identified by a single SVM classifier then the cascade strategy [8] is used to assigning DDI types to drug pairs. The main factors of our approach are the partition of the datasets and the combination of novel feature sets. Based on many syntactic properties, the original dataset is partitioned into 5 subsets to obtain more consistent sub datasets, then feature sets are optimized for each sub dataset. When evaluated on the DDIExtraction 2013 challenge corpus, our system achieved an overall F1-score of 71.79%, which outperforms the current state-of-the-art system by 2%. As future work, we plan to complete this system by a named entity recognition module. The system is initially built to extract DDIs, but it can effortlessly be adapted to other relation extraction tasks such as gene-disease relations and protein-protein interactions.

ACKNOWLEDGMENT

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 taken 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] A. Raihani, N. Laachfoubi, "Extracting drug-drug interactions from biomedical text using a feature-based kernel approach". Journal of Theoretical and Applied Information Technology 15th October 2016 -- Vol. 92.

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

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

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

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

[13] 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.

[14] 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.

[15] J. Björne, T. Salakoski, Generalizing biomedical event extraction, in: Proceedings of BioNLP Shared Task 2011 Workshop, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 183–191.

[16] S. Kim, W.J. Wilbur, Classifying protein–protein interaction articles using word and syntactic features, BMC Bioinformatics 12 (Suppl 8) (2011) S9, http://dx.doi.org/10.1186/1471-2105-12-S8-S9.

[17] 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].

[18] 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, pp. 121–130.