Journal of Biomedical Informatics 55 (2015) 23–30

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach

Sun Kim [1], Haibin Liu *,[1], Lana Yeganova [1], W. John Wilbur

National Center for Biotechnology Information (NCBI), Bethesda, MD, USA

ARTICLE INFO

Article history:
Received 21 October 2014
Revised 21 January 2015
Accepted 11 March 2015
Available online 19 March 2015

Keywords:
Drug–drug interaction
Biomedical literature
Linear kernel approach

ABSTRACT

Identifying unknown drug interactions is of great benefit in the early detection of adverse drug reactions. Despite existence of several resources for drug–drug interaction (DDI) information, the wealth of such information is buried in a body of unstructured medical text which is growing exponentially. This calls for developing text mining techniques for identifying DDIs. The state-of-the-art DDI extraction methods use Support Vector Machines (SVMs) with non-linear composite kernels to explore diverse contexts in literature. While computationally less expensive, linear kernel-based systems have not achieved a comparable performance in DDI extraction tasks. In this work, we propose an efficient and scalable system using a linear kernel to identify DDI information. The proposed approach consists of two steps: identifying DDIs and assigning one of four different DDI types to the predicted drug pairs. We demonstrate that when equipped with a rich set of lexical and syntactic features, a linear SVM classifier is able to achieve a competitive performance in detecting DDIs. In addition, the *one-against-one* strategy proves vital for addressing an imbalance issue in DDI type classification. Applied to the DDIExtraction 2013 corpus, our system achieves an *F*1 score of 0.670, as compared to 0.651 and 0.609 reported by the top two participating teams in the DDIExtraction 2013 challenge, both based on non-linear kernel methods.

1. Introduction

New drugs are generally studied on relatively small and homogeneous patient populations. As a result, pharmaceuticals often have side effects that remain unnoticed until they are already available to the public. This is especially true of side effects that emerge when two drugs are co-administered. A change in the effect of one drug in the presence of another drug is known as a drug–drug interaction (DDI) [1]. It is characterized as an increase or decrease in the action of either substance, or it may be an adverse effect that is not normally associated with either drug. Understanding these drug–drug interactions and their downstream effects is of significant importance, leading to reduced number of drug-safety incidents and reduced healthcare costs.

To address the DDI problem, a number of drug databases such as DrugBank [2] and Stockley's Drug Interactions [1] have been created. Yet, they cover only a fraction of knowledge available. A large amount of up-to-date information is still hidden in the text of journal articles, technical reports and adverse event reporting systems, and this body of unstructured published literature is growing rapidly. MEDLINE®, for example, has doubled in size within the last decade and currently contains about 23 million documents. This creates an urgent need for text mining techniques to extract DDI information.

Using text mining techniques for DDI extraction has received less attention compared to other biomedical relation extraction tasks (e.g., protein–protein interactions), possibly due to the lack of gold standard sets [3–6]. The DDIExtraction challenges are the first community-wide competition addressing the DDI extraction problem [7,8] and a series of studies have been reported at the 2011 and 2013 challenge workshops [9–11].

Top performing systems in the DDIExtraction challenges use Support Vector Machines (SVMs) with non-linear kernels [12,13]. To handle structural representations of input instances, such as dependency graphs, non-linear kernels directly calculate similarities between two graphs by comparing embedded subgraphs [14]. While non-linear kernels are theoretically capable of implicitly searching a high dimensional feature space of subgraphs, existing methods generally exploit only a partial feature space because of the exponential number of subgraphs [15]. In addition, non-linear kernels are frequently combined into composite kernels [12,11]. Composite kernels, however, incur more computational cost because the complexity of the underlying kernels accumulates

* Corresponding author.
E-mail addresses: sun.kim@nih.gov (S. Kim), haibin.liu@nih.gov (H. Liu), lana.yeganova@nih.gov (L. Yeganova), wilbur@ncbi.nlm.nih.gov (W.J. Wilbur).
[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.jbi.2015.03.002
1532-0464/Published by Elsevier Inc.

and additional learning is required to optimize the weights for individual kernels.

Despite the popularity of non-linear kernel methods, linear kernels are a good alternative for relation extraction tasks [16–18]. Linear kernels with word-level features alone provide a strong baseline performance [11,12]. Moreover, they can explicitly include nodes, edges and path structures of the dependency graphs [17]. Also, the straightforward representation of linear kernels enables the intuitive interpretation of obtained results. Most importantly, when training large-scale datasets, it has been demonstrated that often linear kernels are the only practical choice [19,20]. However, the performance of linear kernel systems in DDI extraction tasks has a noticeable gap from that of the top systems using non-linear kernels [7,8,21].

We conjecture that linear kernel-based systems may benefit from a rich set of lexical and syntactic features. With the goal to build a simple and scalable system, we develop a DDI extraction system based on a single linear SVM classifier. We define five types of features to capture the complexity of data: word features with position information, pairs of non-adjacent words, dependency relations, parse tree structures and tags for differentiating DDI pairs within the same noun phrase. Unlike other state-of-the-art systems [13,21] which incorporate external, domain-specific resources, our features originate exclusively from training data.

We evaluate our system on the DDIExtraction 2013 corpus [22]. Consistent with other studies [11–13], we adopt a two-phase approach, where DDI pairs are identified first, and then classified into specific DDI types. The proposed method achieves an overall *F*-score of 67% which outperforms the best performing system by 1.9%. We believe that the strength of our method comes from using a diverse set of features. In addition, the *one-against-one* strategy [23] used in the DDI type classification contributes to the higher performance. As the first linear kernel method that achieves the state-of-the-art performance on both DDI detection and classification tasks, we consider it a strong alternative to the nonlinear, composite kernel-based approaches. The inherent simplicity of the method adds transparency to the overall system, which could be especially beneficial if the system is used as a part of a more complicated schema. The source code for generating the features proposed in this article is available at http://www.ncbi.nlm.nih.gov/IRET/DDI.

2. Methods

Fig. 1 illustrates the overall architecture of our DDI extraction system. A binary classifier is trained first to extract interacting drug pairs from all candidate interactions. A DDI type classifier is then built to classify the interacting pairs into predefined relation categories. Our approach focuses on interactions expressed within the boundaries of a single sentence, and also assumes that drug entities involved in the target interactions have been annotated.

In this section, we first elaborate the five types of features used, including two novel features proposed for the DDI problem: word pair and noun phrase-constrained coordination (NPC) features. Then, we briefly introduce the preprocessing steps completed on both training and test data. Next, we describe our linear SVM classifier with a modified Huber loss function [24]. In the end, we compare our method with existing DDI extraction systems.

2.1. Features

2.1.1. Word features

Word-level features such as individual words in a sentence and sequences of words have been demonstrated to provide a strong performance baseline in extracting relational knowledge

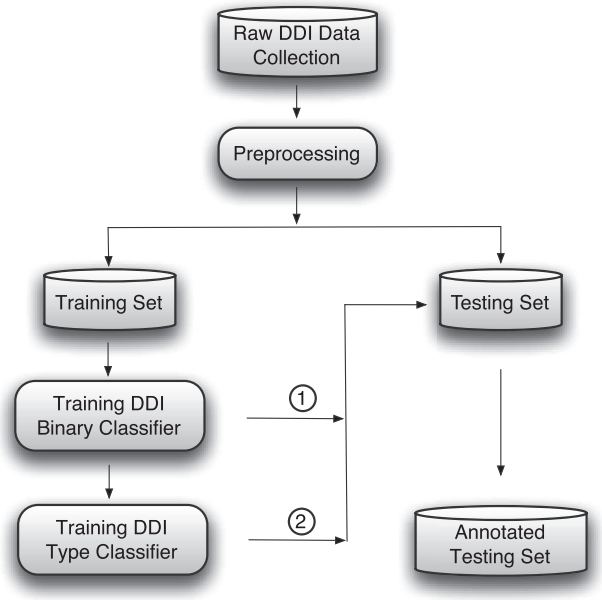

Fig. 1. Two-phase DDI extraction framework. DDI detection (①) decides whether a drug pair interacts. DDI type classification (②) assigns DDI types to interacting pairs.

[11,17,25]. Hence, in our system, we use *n*-gram features of size up to 3, i.e., unigrams, bigrams and trigrams. Including *n*-grams of larger size does not always lead to a performance increase due to the data sparseness problem [25]. Similar to the works of He et al. [11] and Giuliano et al. [26], the position information is appended to each word feature according to positions of words in a sentence relative to an investigated drug pair: *before* (BF), *between* (BE) and *after* (AF). For instance, "Interaction_BF of_BF **ketamine** and_BE **halothane** in_AF rats_AF" where "ketamine" and "halothane" are two drug names.

2.1.2. Word pair features

While word features may capture repetitive expression patterns in neighboring words, they are not able to discover patterns involving distant words in a sentence. A simple solution to capture distant word patterns is to extract all possible word combinations from a training set. However, this approach increases the number of features considerably, and it also may degrade classification performance. To address this issue, we here propose a novel technique for selecting significant word pairs.

First, unigram word features are paired and only those pairs with a minimum frequency k are selected. Second, for selected word pairs, p-values are calculated using the hypergeometric distribution [27]. The p-value reflects how strongly a feature is represented in the positive set as compared to the negative set. It relates to the null hypothesis that the co-occurrence of two words is randomly distributed between positive and negative sets. If the co-occurrence is randomly distributed, the word pair will have a high p-value. If the p-value is low, this indicates a $1 - p$ probability that the co-occurrence is not random and is likely indicative for positive DDIs.

To obtain the most useful word pairs, we need the least restrictive frequency and the most restrictive p-value. In this work, we set $k = 200$ and p-value = 0.01 based on $F1$ scores via 10-fold document-level cross validation on the training set. This significant p-value helps select 588 word pairs from a total of 449,826 pairs with k above 200. This feature set contains certain informative word pairs such as "$drug_1 \cdots drug_2 \cdots$ **increase** $\cdots$ **level**" and

"**interaction** $\cdots drug_1 \cdots$ **with** $\cdots drug_2$", which provide a strong signal indicating DDIs.

2.1.3. Dependency graph features

Dependency graphs use nodes to represent words in a sentence and edges to describe governor-dependent relations between the words. Thus, they can capture long-range dependencies among sentential constituents by considerably narrowing the linear order distance between target entities [19]. Since the syntactic dependencies closely approximate the underlying semantic relationships, they have been effectively used by biomedical knowledge extraction systems [28,16,29–31].

While some approaches use an all-inclusive approach to explore paths of all possible lengths between any two nodes in a dependency graph [32,15], the shortest path between two nodes is particularly likely to carry the most valuable information about their mutual relationship [33–36]. Given the dependency graph of each sentence, therefore, the shortest dependency path connecting the target drugs in the undirected version of the graph is selected. If there exist multiple shortest paths, we randomly choose one. The extracted path is then transformed into an ordered sequence of individual dependency relations, in which original relation labels and edge orientations are appropriately preserved. For instance, a dependency path "ketamine $\overset{\textbf{nsubj}}{\longleftarrow}$ interacts $\overset{\textbf{prep\_with}}{\longrightarrow}$ halothane" is encoded as "nsubj(interacts,ketamine); prep_with(interacts,halothane)". We further split a dependency path into *n*-grams of up to size 3. Compared to the vertex-walks based *q*-grams proposed by Kuboyama et al. [37], our dependency features are equivalent to *q*-bigrams in their work.

2.1.4. Parse tree features

We have observed in the training data of the DDIExtraction 2013 challenge [8] that the textual descriptions of more than 25% of the total 4023 interacting drug–drug pairs involve different subordinate clauses of a sentence such as "*If additional* ***adrenergic drugs*** *are to be administered by any route, they should be used with caution because the pharmacologically predictable sympathetic effects of* ***BROVANA*** *may be potentiated.*", or appear in the main sentence and its subordinate clause for instance "*When* ***carbamazepine*** *is withdrawn from the combination therapy,* ***aripiprazole*** *dose should then be reduced.*" Capturing these grammatical patterns is thus important to the successful extraction of these interactions. Compared to constituent parse trees which inherently retain phrasal and clausal structures, dependency graphs do not explicitly preserve this rich syntactic information.

It has been shown that systems exclusively relying on parse trees obtain inferior results to using dependency graphs in information extraction tasks [31,16]. However, combining information from both representations improves the overall performance [38,11]. To supplement our dependency features, we extracted the shortest path connecting the two investigated drugs in the parse tree. The resulting path is a sequence of grammatical tags such as "*NP S VP VP SBAR S VP PP NP*", representing a concise syntactic traverse from one drug to the other. To capture frequent syntactic patterns, *n*-grams over individual tags of size 3 are used as our parse tree features. Unlike word features and dependency graph features, unigrams and bigrams are not used for parse tree features because these patterns are too short to represent syntactic structures.

2.1.5. Noun phrase-constrained coordination features

Linguistically, relationships are rarely discussed among entities in syntactic constituents where 3 or more target entities appear in a coordination. For instance, in the sentence "***Clidinium*** *may decrease the effect of* ***phenothiazines***, ***levodopa***, *and* ***ketoconazole***.", the coordination structure "phenothiazines, levodopa, and ketoconazole" is used to enumerate a list of drugs that potentially interact with "Clidinium', with no indication of interactions among the drugs inside the coordination.

In this work, we use base noun phrases to constrain the scope of the coordinated drug mentions, and propose a novel, noun phrase-constrained coordination feature to indicate if the target drugs are coordinated in a noun phrase. Suppose the total number of drug mentions in a base noun phrase NP_b is n, the new feature f_c for each candidate DDI pair (d_1, d_2) is defined as follows:

$$f_c(d_1, d_2) = \begin{cases} 1, & \text{if } (d_1, d_2) \in NP_b \text{ and } n \geqslant 3, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Because of the cascaded structure, candidate noun phrases are recursively extracted from the constituent parse trees. A base noun phrase NP_b is defined to be the longest noun phrase that does not contain any prepositional phrases (PP), verb phrases (VP), subordinate clauses (SBAR) or sentences (S). We observed only 16 interacting drug–drug pairs (0.2%) among the total 8045 pairs satisfying $f_c = 1$ in the DDIExtraction 2013 training data.

2.2. Preprocessing

Several standard preprocessing steps are first completed on both training and test data. These include sentence segmentation and tokenization, Part-of-Speech (POS) tagging and syntactic parsing that produces constituent parse trees and dependency graphs for sentences [39,40]. To ensure generalization of the features, drug mentions are anonymized using "DRUG" for target drugs and "DRUG_OTHER" for other drugs. Numbers are replaced by a generic tag "NUM", and other tokens normalized into their corresponding lemmas by the BioLemmatizer [41].

The same drug mentions can appear multiple times in a sentence. Considering that drugs are unlikely to interact with themselves, candidate pairs with both drugs referring to the same name are removed [42,12]. This helps reduce candidate drug pairs. In addition, we notice that drug names are sometimes separated by a colon from the detailed description on their interactions with other drugs. For instance, "***Morphine***: *Combination* ***hormonal contraceptives*** *may increase the clearance of* ***morphine***." In such cases, as the description itself is an independent sentence, pairing "Morphine" on the left of the colon with drug mentions in the description may interfere with the narrative flow of the description. Thus, we remove the drug mentions on the left of the colon from further consideration.

Fig. 2 shows the preprocessing step and the feature vector obtained for an example sentence.

2.3. SVM classifier

For DDI detection and classification, we use an SVM classifier with the modified Huber loss function [24]. We have observed that the modified Huber loss function has consistently achieved better performance than the hinge loss function used in traditional SVMs for biomedical classification problems [43,25,44]. Let T denote the size of the training set. Let the binary feature vector of the ith pair in the training set be denoted by X_i. Let $y_i = 1$ if the pair is annotated as positive and $y_i = -1$ otherwise. Let w denote a vector of feature weights, of the same length as X_i. Let θ denote a threshold parameter, and let λ denote a regularization parameter. Then the cost function is given by:

$$C = \frac{1}{2}\lambda|w|^2 + \frac{1}{T}\sum_{i=1}^{T} h(y_i(\theta + w \cdot X_i)), \quad (2)$$

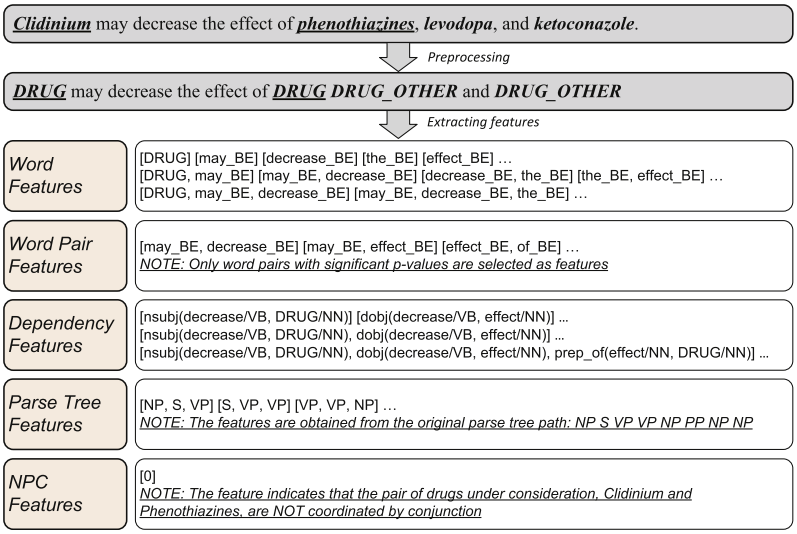

Fig. 2. An example of preprocessing and feature extraction. The underlined drug pair, *clidinium* and *phenothiazines*, is the candidate DDI. 'NPC' means noun phrase-constrained coordination and 'BE' denotes *between* candidate drugs.

where the function h is the modified Huber loss function defined as follows:

$$h(z) = \begin{cases} -4z, & \text{if } z \leqslant -1, \\ (1-z)^2, & \text{if } -1 < z < 1, \\ 0, & \text{if } 1 \leqslant z. \end{cases} \quad (3)$$

The values of the parameters, w and θ minimizing C are determined using a gradient descent algorithm. The regularization parameter λ is computed from the training set as follows:

$$\lambda = \lambda' \langle |x| \rangle^2, \quad (4)$$

where $\langle |x| \rangle$ is the average Euclidean norm of the feature vectors in the training set. The parameter λ' is set to 0.00001 for the DDI task.

2.4. Comparison with existing DDI extraction systems

Our linear SVM classifier relies on a set of general features to achieve the state-of-the-art performance. Word-level features, dependency graphs and parse trees are commonly used by relation extraction systems [31,17,12,11]. Compared to the implicit use in non-linear kernel systems [11,12], features extracted from dependency graphs and parse trees are represented as linear n-grams in our system. As the only linear kernel system of the top 3 DDIExtraction 2013 teams, UTurku [21] makes intensive use of dependency graph features, but does not take advantage of the rich syntactic information in parse trees. In our system, both are used to complement each other for multiple aspects of structural analysis of sentences.

The noun phrase-constrained coordination and word pair features are novel in our DDI extraction approach. He et al. [11] aimed for general conjunction structures around investigated drugs by encoding relative distances into features. However, our coordination feature is able to explicitly capture the coordination structure of enumerated drugs, and semantically constrain the scope of the structure by noun phrases. Also, even though Bobić et al. [45] combined non-adjacent tokens across sections ("BF", "BE" and "AF"), they used all the combinations without any feature selection. We find that the feature selection using p-values is effective for reducing the data complexity and for improving the DDI extraction performance.

In addition, Björne et al. [21] took advantage of domain knowledge derived from DrugBank [2] and MetaMap [46]. Besides domain-specific resources, He et al. [11] additionally asked domain experts to manually compile keyword and semantic type features. Considering that our features originate exclusively from training data and their extraction is domain independent, our feature types may be more generalizable to other relation extraction tasks.

3. Results and discussion

3.1. Dataset

We train and evaluate the proposed approach on the DDI corpus from the DDIExtraction 2013 challenge [8]. The DDI corpus includes 905 manually annotated documents from the DrugBank database and MEDLINE abstracts, which are split into 714 and 191 documents for training and test sets, respectively. The DDI set provides examples by sentences and, for each sentence, all drug mentions and DDI pairs are annotated. There are four different types of DDI relationships in the set [22]; *mechanism*, *effect*, *advice* and *int*. *Mechanism* is used for DDIs that are described by their pharmacokinetic (PK) mechanism.[2] *Effect* is for DDIs describing an effect or a pharmacodynamic (PD) mechanism.[3] *Advice* is used when a recommendation or advice related to a DDI is given. *Int* is used when a DDI appears in a sentence without providing any additional information.

Non-interacting drug pairs are not explicitly provided in the DDI corpus. Hence, all drug pairs that do not overlap with positives are considered as negatives. Table 1 shows the number of positive and negative pairs before and after preprocessing. Removing pairs with the same drug mentions and the colon case described in Preprocessing filters out 29 positive and 3972 negative pairs. The removed positive pairs constitute 0.58% of the positive set. We find the fraction to be negligible compared to the advantage of not showing nearly 14% negative pairs to SVM classifiers. In a basic setting where only word features are used, this step improves $F1$ by 4% using 10-fold document-level cross validation on the training set.

[2] What the body does to the drug; absorption, distribution, metabolism, elimination.

[3] What the drug does to the body.

Table 1
Number of positive and negative pairs in the dataset. 29 positive and 3972 negative pairs were removed through preprocessing.

	Original set		Preprocessed set	
	Positive	Negative	Positive	Negative
Training	4023	23,756	3996	20,368
Test	979	4734	977	4150
Total	5002	28,490	4973	24,518

Table 2 shows the number of positive DDI pairs in the training and test sets for each DDI type. *Mechanism* and *effect* are dominant classes, while *advice* and *int* contain many fewer instances comprising about 25% of the positive set. This unbalanced size in the training data may be problematic, in particular, for a machine learning solution because it may lead to poor classification performance [47]. To address the issue, we apply the *one-against-one* approach for the DDI type classification task. Compared to the *one-against-all* strategy which takes negative examples from all non-positive classes, the *one-against-one* strategy uses only one negative class for each classifier; it alleviates the imbalance.

3.2. Performance comparison

Eight teams participated in the DDIExtraction 2013 challenge, and the official performance ranged from 21.4% to 65.1% *F*1 [8]. Table 3 compares our method with the top three ranking teams in the DDIExtraction task based on *F*1 scores. Our approach achieves 67% *F*1 for detection and classification performance ('CLA'), whereas FBK-irst, WBI and UTurku produced 65.1%, 60.9% and 59.4% *F*1, respectively. For DDI detection performance (DEC), i.e. before applying the *one-against-one* strategy, the proposed approach performs second best by achieving 77.5% *F*1.

FBK-irst [12] uses a hybrid kernel for combining linear features, shallow linguistic and path-enclosed tree kernels. WBI [13] utilizes an ensemble approach to combine outputs from other DDI prediction tools. UTurku [21] uses a linear kernel with domain knowledge from external resources as well as word and dependency graph features. For DDI type classification, FBK-irst uses binary SVMs with a *one-against-all* strategy. WBI and UTurku use a multi-class SVM, which does not require choosing either *one-against-one* or *one-against-all*. Our method, on the other hand, uses a simple binary SVM classifier with linear kernel for identifying DDIs and the *one-against-one* strategy for assigning DDI types. We choose the *one-against-one* strategy to reduce the negative effect of unbalanced classes. In Table 3, our approach performs best for *mechanism*, *effect* and *advice* types. In contrast, the same approach does not perform well for *int*. This is different from the 10-fold document-level cross validation results for the training set (Refer to Section 3.4). By definition, *int* contains DDIs which cannot be assigned to other three types. Thus, either the general description of *int* or insufficient evidence from the small number of training (188 examples) and testing (96 examples) sets may play a role.

Table 2
Positive drug pairs used for training and testing. The preprocessed set is compared with the original set.

Class	Original set		Preprocessed set	
	Training	Test	Training	Test
Mechanism	1321	302	1309	301
Effect	1688	360	1675	359
Advice	826	221	824	221
Int	188	96	188	96
Total	4023	979	3996	977

Table 3
Performance comparison between the proposed method and top-ranking approaches on the DDIExtraction 2013 test data. The performance is measured based on *F*1 scores. 'CLA' indicates detection and classification performance for all classes. 'DEC' indicates detection performance. 'MEC', 'EFF', 'ADV' and 'INT' are for *mechanism*, *effect*, *advice* and *int* types respectively. The highest scores are highlighted in bold.

Method	CLA	DEC	MEC	EFF	ADV	INT
Our method	**0.670**	0.775	**0.693**	**0.662**	**0.725**	0.483
FBK-irst	0.651	**0.800**	0.679	0.628	0.692	**0.547**
WBI	0.609	0.759	0.618	0.610	0.632	0.510
UTurku	0.594	0.696	0.582	0.600	0.630	0.507

Table 4
Performance comparison between DrugBank and MEDLINE test sets in DDIExtraction 2013. The performance is measured based on *F*1 scores. 'CLA' indicates detection and classification performance for all classes. 'DEC' indicates detection performance. 'MEC', 'EFF', 'ADV' and 'INT' are for *mechanism*, *effect*, *advice* and *int* types respectively.

Dataset	CLA	DEC	MEC	EFF	ADV	INT
DrugBank	0.698	0.804	0.714	0.706	0.736	0.497
MEDLINE	0.382	0.471	0.455	0.352	0.429	0.250

Furthermore, Table 4 shows the separate performance of our system on DrugBank and MEDLINE test documents. While the DDI detection and classification ('CLA') performance on the DrugBank set shows 69.8% *F*1, the performance on the MEDLINE set is substantially lower (38.2% *F*1). This difference is consistent with the results from the DDIExtraction 2013 challenge [8]. It may be due to the small number of training examples provided for MEDLINE. The 232 DDI pairs in the MEDLINE training set constitute only 6% of the overall training data. In addition, the DrugBank and MEDLINE documents may have different characteristics [12].

In the following subsections, we discuss the contribution of each feature type and the effect of the *one-against-one* strategy compared to the *one-against-all* strategy.

3.3. Feature analysis

Table 5 presents changes of DDI detection performance by adding each feature type to the baseline (word features). For the results, 10-fold document-level cross validation was performed ten times and the scores were averaged. Relative positions attached to word features improve the *F*1 performance by 24.1%. This significant improvement is understandable because relative position is a good indicator whether an individual word is used in describing DDIs.

Using word features with positions as a baseline, word pairs, dependency relations, parse trees and NPC are added and evaluated individually. From the table, word pairs and parse trees contribute the most by increasing *F*1 by 1.9% and 1.4%, respectively. Dependency relations and NPC have less impact on the performance, however, dependency relations help get higher precision and NPC helps the recall. While word features cover neighboring words, syntactic structure and word pair features seem to help with the overall picture of DDI sentences. It is understandable, yet remarkable that using words with relative positions alone achieves such high performance for identifying DDIs. Integrating position information into word features is important because one sentence often involves multiple drug mentions and the position information helps differentiate the context of interacting pairs from that of non-interacting ones. It would be interesting to see how the same strategy would work on other entity–entity relationship extraction tasks.

An advantage of using the linear kernel approach is that obtained results have an intuitive interpretation. Although

Table 5
Performance changes by varying feature types in DDI detection. The baseline performance was measured by using word features with position ('pos') information. 'Change' shows the *F*1 score difference between the baseline and the performance in each row. 10-fold document-level cross-validation was performed ten times for the training set and scores were averaged.

Features	Precision	Recall	*F*1	Change
Baseline (w/o pos)	0.544	0.427	0.478	−24.1%
Baseline	0.774	0.670	0.719	–
+ Word pairs	0.780	0.700	0.738	+1.9%
+ Dependency	0.791	0.669	0.725	+0.6%
+ Parse trees	0.783	0.688	0.733	+1.4%
+ NPC	0.771	0.681	0.723	+0.4%
All features	0.798	0.711	0.752	+3.3%

Table 5 provides some information, it still lacks in explaining what features contribute to identify a particular DDI in a sentence. Fig. 3 shows a simple visual aid, where significant words are highlighted based on the weights of word and word pair features from the SVM classifier. The features listed are the ones that classify the example as positive. The words with higher weights are emphasized by thicker lines and darker gray. From the highlighted words, one can understand that "*DRUG* with *DRUG*" and "not recommended" are the key elements for detecting a DDI in the sentence.

3.4. DDI type detection

Our approach to DDI extraction has two steps. First, drug pairs are classified whether they interact or not. Second, one of four DDI types (*mechanism*, *effect*, *advice* and *int*) is assigned to interacting pairs.

Two popular ways to address multi-class classification using binary classifiers are the *one-against-one* and *one-against-all* strategies [23]. The *one-against-all* method builds a classifier for each class vs. all other classes. The *one-against-one* strategy, however, builds a binary classifier for each pair of classes, and the output of the classifiers is aggregated using majority voting. DDI type classification requires 4 and 12 classifiers for *one-against-all* and *one-against-one*, respectively.

A critical issue for the DDI type classification is that the number of training examples differs significantly among the four classes (Table 2). For the *one-against-all* strategy, this imbalance may lead to poor performance on the small classes. Therefore, we use the *one-against-one* strategy for the DDI classification task. Tables 6 and 7 present the performance difference between *one-against-all* and *one-against-one*. The performance is better and more balanced with the *one-against-one* strategy. Table 8 shows *F*1 scores for *one-against-all* and *one-against-one* on the test set. Although there is a slight performance decrease on *effect*, *F*1 scores increase on other DDI types.

Table 6
Performance comparison on the *one-against-all* strategy. 10-fold document-level cross-validation was performed ten times for the training set and scores were averaged.

Class	Precision	Recall	*F*1
Mechanism	0.911	0.774	0.837
Effect	0.885	0.839	0.861
Advice	0.894	0.830	0.861
Int	0.725	0.495	0.587

Table 7
Performance comparison on the *one-against-one* strategy. 10-fold document-level cross-validation was performed ten times for the training set and scores were averaged.

Class	Precision	Recall	*F*1
Mechanism	0.941	0.964	0.952
Effect	0.943	0.979	0.960
Advice	0.921	0.960	0.940
Int	0.912	0.953	0.932

Table 8
Performance comparison (*F*1 scores) of the *one-against-all* and *one-against-one* strategies on the test set.

Class	*One-against-all*	*One-against-one*
Mechanism	0.673	0.693
Effect	0.671	0.662
Advice	0.718	0.725
Int	0.440	0.483

3.5. Annotation inconsistency

The proposed method is completely data-driven. Even though SVM classifiers are robust to noisy training examples to some degree, our approach is sensitive to the quality of the training set.

During algorithm development, we found that the DDI corpus contained irregularly formatted sentences. In one case, section titles are concatenated with the next sentence as shown in Fig. 4. A more severe problem can occur when tables are converted to text. Sentences derived from tables can cause false positive and

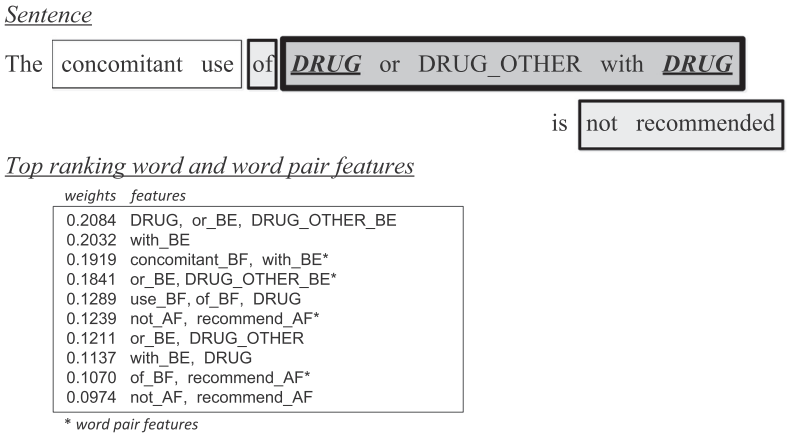

Fig. 3. A solution for presenting drug pairs with significant word and word pair features. Highly weighted words are highlighted in the sentence and emphasized according to all the weights they receive in the feature list. 'BF', 'BE' and 'AF' mean *before*, *between* and *after*, respectively. *DRUG* indicates a target drug.

Corticosteroids: Concomitant administration of ***aspirin*** and corticosteriods may decrease salicylate plasma levels.

Theophylline:
As with some other ***quinolones***, concurrent administration of ***ciprofloxacin*** with ***theophylline*** may lead to elevated serum concentrations of ***theophylline*** and prolongation of its elimination half-life.

Fig. 4. Example sentences which start with "[drug name]:". "[drug name]:" is a section title which is concatenated with the next sentence in the DDIExtraction set.

Corticosteroids: A relationship of functional antagonism exists between ***vitamin D analogues***, which promote ***calcium*** absorption, and ***corticosteroids***, which inhibit calcium absorption.

Corticosteroids: Concomitant administration of ***aspirin*** and ***corticosteroids*** may decrease salicylate plasma levels.

Fig. 5. Example sentences, where some drug names are not annotated. "calcium" in the first sentence and the second "corticosteroids" in the second sentence are not annotated as drug names.

false negative drug pairs. Positive drug pairs from a table are not useful because the proposed approach is designed for grammatically well-formed text. In our experiments, we kept the table-derived sentences because no rules could be found to remove them. We presume that DrugBank sentences were automatically extracted from HTML or XML data prior to annotation, or curators annotated the dataset in a structured form and it was later flattened by an automatic extraction process.

Another problem in the dataset is that not all drug mentions are annotated. In Fig. 5, "calcium" in the first sentence and the second "corticosteroids" in the second sentence are not annotated as drug names. It is obvious that "corticosteroids" in the second example is overlooked by curators. However, it is difficult to decide whether "calcium" is a drug name here. While "calcium" is often annotated as a drug in the dataset, there are also exceptions where "calcium" is not considered a drug. We assume that it is either overlooked or not considered a drug.

4. Conclusion

We present a two-step classification algorithm for identifying DDIs from biomedical literature. Unlike other state-of-the-art approaches, the proposed method focuses on word and syntactic features in a linear SVM. For assigning DDI types to drug pairs, positive DDI pairs are first identified by a single SVM classifier, and multiple SVM classifiers are used to decide DDI types through the *one-against-one* strategy in the second step. The features used in our approach are words with relative positions, pairs of non-adjacent words, dependency relations, syntactic structures and noun phrase-constrained coordination tags. Applied to the DDIExtraction corpus, the proposed method showed competitive performance to top-ranking teams in the DDIExtraction 2013 challenge by obtaining 67% *F*1.

The main contribution of the proposed method is the rich-feature based approach using linear SVMs. Non-linear, composite kernel approaches can directly use structural information. However, they tend to be complex and may not be readily applicable to a large-scale dataset. Our feature-based approach, on the other hand, is more flexible. Feature types can be easily evaluated, hence an intuitive interpretation is readily available. The linear kernel approach is also a practical alternative for large-scale problems. Moreover, the *one-against-one* strategy used in the DDI type classification is found to be useful for achieving higher classification performance. It addresses some difficulties of solving multi-class classification with binary classifiers. As future work, we plan to extend our system by integrating it with named entity recognition tools. We also would like to evaluate the end-to-end DDI extraction system for PubMed® abstracts in a large-scale setting.

Acknowledgments

The authors thank Isabel Segura-Bedmar for her help evaluating our preliminary results on the DDIExtraction 2013 dataset. *Funding*: This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

[1] K. Baxter, C.L. Preston (Eds.), Stockley's Drug Interactions, Pharmaceutical Press, London, UK, 2013.
[2] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, et al., DrugBank 3.0: a comprehensive resource for 'Omics' research on drugs, Nucl. Acids Res. 39 (Suppl 1) (2011) D1035–D1041.
[3] S. Duda, C. Aliferis, R. Miller, A. Statnikov, K. Johnson, Extracting drug–drug interaction articles from MEDLINE to improve the content of drug databases, in: AMIA Annual Symposium Proceedings, 2005, pp. 216–220.
[4] D.L. Rubin, C.F. Thorn, T.E. Klein, R.B. Altman, A statistical approach to scanning the biomedical literature for pharmacogenetics knowledge, J. Am. Med. Inform. Assoc. 12 (2) (2005) 121–129.
[5] I. Segura-Bedmar, P. Martínez, C. de Pablo-Sánchez, A linguistic rule-based approach to extract drug–drug interactions from pharmacological documents, BMC Bioinformatics 12 (Suppl 2) (2011) S1.
[6] I. Segura-Bedmar, P. Martínez, C. de Pablo-Sánchez, Using a shallow linguistic kernel for drugdrug interaction extraction, J. Biomed. Inform. 44 (5) (2011) 789–804. http://dx.doi.org/10.1016/j.jbi.2011.04.005.
[7] I. Segura-Bedmar, P. Martínez, D. Sánchez-Cisneros, The 1st 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 2011), 2011, pp. 1–9.
[8] I. Segura-Bedmar, P. Martinez, M. Herrero-Zazo, SemEval-2013 task 9: extraction of drug–drug interactions from biomedical texts, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 341–350.
[9] P. Thomas, M. Neves, I. Solt, D. Tikk, U. Leser, Relation extraction for drug–drug interactions using ensemble learning, in: Proceedings of the 1st Challenge Task on Drug–Drug Interaction Extraction (DDIExtraction 2011), 2011, pp. 11–18.
[10] M.F.M. Chowdhury, A. Lavelli, Drug–drug interaction extraction using composite kernels, in: Proceedings of the 1st Challenge Task on Drug–Drug Interaction Extraction (DDIExtraction 2011), 2011, pp. 27–33.
[11] 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.
[12] M.F.M. Chowdhury, A. Lavelli, 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), 2013, pp. 351–355.

[13] P. Thomas, M. Neves, T. Rocktäschel, U. Leser, WBI-DDI: drug–drug interaction extraction using majority voting, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 628–635.
[14] M. Collins, N. Duffy, Convolution kernels for natural language, in: Advances in Neural Information Processing Systems (NIPS 2001), 2001, pp. 625–632.
[15] Y. Zhang, H. Lin, Z. Yang, J. Wang, Y. Li, A single kernel-based approach to extract drug–drug interactions from biomedical literature, PLoS One 7 (11) (2012) e48901.
[16] D. Tikk, P. Thomas, P. Palaga, J. Hakenberg, U. Leser, A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature, PLoS Comput. Biol. 6 (2010) e1000837.
[17] 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.
[18] J. Björne, T. Salakoski, TEES 2.1: automated annotation scheme learning in the BioNLP 2013 shared task, in: Proceedings of the BioNLP Shared Task 2013 Workshop, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 16–25.
[19] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, T. Salakoski, Extracting complex biological events with rich graph-based feature sets, in: Proceedings of the Workshop on BioNLP (BioNLP '09), Association for Computational Linguistics, 2009, pp. 10–18.
[20] J. Björne, F. Ginter, S. Pyysalo, J. Tsujii, T. Salakoski, Scaling up biomedical event extraction to the entire PubMed, in: Proceedings of the 2010 Workshop on Biomedical Natural Language Processing (BioNLP '10), ACL, 2010, pp. 28–36.
[21] J. Björne, S. Kaewphan, T. Salakoski, UTurku: drug named entity recognition and drug–drug interaction extraction using SVM classification and domain knowledge, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 651–659.
[22] 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.
[23] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425, http://dx.doi.org/10.1109/72.991427.
[24] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: Proceedings of the 21st International Conference on Machine Learning (ICML 2004), 2004, pp. 919–926.
[25] 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.
[26] C. Giuliano, A. Lavelli, L. Romano, Exploiting shallow linguistic information for relation extraction from biomedical literature, in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006), 2006, pp. 401–408.
[27] W. Kim, W.J. Wilbur, Corpus-based statistical screening for content-bearing terms, J. Am. Soc. Inform. Sci. Technol. 52 (3) (2001) 247–259.
[28] K. Fundel, R. Küffner, R. Zimmer, Relex—relation extraction using dependency parse trees, Bioinformatics 23 (3) (2007) 365–371.
[29] D. McClosky, E. Charniak, M. Johnson, Automatic domain adaptation for parsing, in: Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2010), Association for Computational Linguistics, 2010, pp. 28–36.
[30] K. Sagae, Y. Miyao, T. Matsuzaki, J. Tsujii, Challenges in mapping of syntactic representations for framework-independent parser evaluation, in: Proceedings of the Workshop on Automated Syntactic Annotations for Interoperable Language Resources, 2008.
[31] Y. Miyao, K. Sagae, R. Saetre, T. Matsuzaki, J. Tsujii, Evaluating contributions of natural language parsers to protein–protein interaction extraction, Bioinformatics 25 (3) (2009) 394–400.
[32] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, T. Salakoski, All-paths graph kernel for protein–protein interaction extraction with evaluation of cross-corpus learning, BMC Bioinformatics 9 (Suppl 11) (2008) 1–12, http://dx.doi.org/10.1186/1471-2105-9-S11-S2.
[33] R.C. Bunescu, R.J. Mooney, A shortest path dependency kernel for relation extraction, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 2005, pp. 724–731.
[34] F. Rinaldi, G. Schneider, K. Kaljurand, S. Clematide, T. Vachon, M. Romacker, OntoGene in BioCreative II.5, IEEE/ACM Trans. Comput. Biol. Bioinf. 7 (3) (2010) 472–480.
[35] H. Liu, L. Hunter, V. Keselj, K. Verspoor, Approximate subgraph matching-based literature mining for biomedical events and relations, PLoS One 8 (4) (2013) e60954.
[36] H. Liu, K. Verspoor, D.C. Comeau, A. MacKinlay, W.J. Wilbur, Generalizing an approximate subgraph matching-based system to extract events in molecular biology and cancer genetics, in: Proceedings of the BioNLP Shared Task 2013 Workshop, 2013, pp. 76–85.
[37] T. Kuboyama, K. Hirata, H. Kashima, K.F. Aoki-Kinoshita, H. Yasuda, A spectrum tree kernel, Inform. Media Technol. 2 (1) (2007) 292–299.
[38] L. Qian, G. Zhou, Tree kernel-based protein–protein interaction extraction from biomedical literature, J. Biomed. Inform. 45 (3) (2012) 535–543.
[39] D. McClosky, E. Charniak, Self-training for biomedical parsing, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, The Association for Computer Linguistics, 2008, pp. 101–104.
[40] M.-C. de Marneffe, C.D. Manning, The Stanford typed dependencies representation, in: CrossParser '08: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, Association for Computational Linguistics, 2008, pp. 1–8.
[41] H. Liu, T. Christiansen, W.A. Baumgartner, K. Verspoor, Biolemmatizer: a lemmatization tool for morphological processing of biomedical text, J. Biomed. Semantics 3 (2012) 3.
[42] M.F.M. Chowdhury, A. Lavelli, Impact of less skewed distributions on efficiency and effectiveness of biomedical relation extraction, in: Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), 2012, pp. 205–216.
[43] L.H. Smith, W.J. Wilbur, Finding related sentence pairs in MEDLINE, Inform. Ret. 13 (6) (2010) 601–617.
[44] L. Yeganova, D.C. Comeau, W. Kim, W.J. Wilbur, Text mining techniques for leveraging positively labeled data, in: Proceedings of the 2011 Workshop on Biomedical Natural Language Processing (BioNLP '11), Association for Computational Linguistics, 2011, pp. 155–163.
[45] T. Bobić, J. Fluck, M. Hofmann-Apitius, SCAI: extracting drug–drug interactions using a rich feature vector, in: Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 675–683.
[46] A.R. Aronson, Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program, in: AMIA Annual Symposium Proceedings, 2001, pp. 17–21.
[47] N.V. Chawla, Data Mining for Imbalanced Datasets: An Overview, Springer, New York, USA, 2005.