Journal of Biomedical Informatics 44 (2011) 789–804

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

Using a shallow linguistic kernel for drug–drug interaction extraction

Isabel Segura-Bedmar *, Paloma Martínez, Cesar de Pablo-Sánchez

Computer Science Department, Carlos III University of Madrid, Leganés, Spain

ARTICLE INFO

Article history:
Received 8 October 2010
Available online 24 April 2011

Keywords:
Biomedical information extraction
Drug–drug interactions
Patient safety
Shallow linguistic kernel
Machine learning
Unified medical language system
MetaMap

ABSTRACT

A drug–drug interaction (DDI) occurs when one drug influences the level or activity of another drug. Information Extraction (IE) techniques can provide health care professionals with an interesting way to reduce time spent reviewing the literature for potential drug–drug interactions. Nevertheless, no approach has been proposed to the problem of extracting DDIs in biomedical texts. In this article, we study whether a machine learning-based method is appropriate for DDI extraction in biomedical texts and whether the results provided are superior to those obtained from our previously proposed pattern-based approach [1]. The method proposed here for DDI extraction is based on a supervised machine learning technique, more specifically, the shallow linguistic kernel proposed in Giuliano et al. (2006) [2]. Since no benchmark corpus was available to evaluate our approach to DDI extraction, we created the first such corpus, DrugDDI, annotated with 3169 DDIs. We performed several experiments varying the configuration parameters of the shallow linguistic kernel. The model that maximizes the *F*-measure was evaluated on the test data of the DrugDDI corpus, achieving a precision of 51.03%, a recall of 72.82% and an *F*-measure of 60.01%.

To the best of our knowledge, this work has proposed the first full solution for the automatic extraction of DDIs from biomedical texts. Our study confirms that the shallow linguistic kernel outperforms our previous pattern-based approach. Additionally, it is our hope that the DrugDDI corpus will allow researchers to explore new solutions to the DDI extraction problem.

1. Introduction

A drug–drug interaction (DDI) occurs when one drug influences the level or activity of another, for example, increasing plasma concentration of the drug and potentially intensifying its side effects or decreasing its plasma concentration and thereby reducing its effectiveness. Since negative DDIs can be very dangerous, DDI detection is the subject of an important field of research that is crucial for both patient safety and health care cost control. Although health care professionals are supported in DDI detection by different databases, those being used currently are rarely complete, since their update periods can be as long as three years [3]. Drug interactions are frequently reported in journals of clinical pharmacology and technical reports, making medical literature the most effective source for the detection of DDIs. Every year, 300,000 articles are published within the field of pharmacology alone [4]. The management of DDIs is a critical issue, therefore, due to the overwhelming amount of information available [5].

Information extraction (IE) can be of great benefit for both the pharmaceutical industry by facilitating the identification and extraction of relevant information on DDIs, as well as health care professionals by reducing the time spent reviewing the relevant literature. Moreover, the development of tools for automatically extracting DDIs is essential for improving and updating the drug knowledge databases.

Our focus is the detection of sentences carrying information regarding a DDI as well as the specific drugs taking part in the interaction. An additional study goal is to analyze the contribution of current IE methods to DDI extraction and evaluate their performance in select scenarios where technology aiding DDI detection exists and is available. In a previous paper [1], we proposed a hybrid method combining shallow parsing and pattern matching to extract DDIs from texts. Unfortunately, this initial approach yielded poor results (precision = 48.89%, recall = 24.81%, *F*-measure = 32.92%). In the present article, our approach is based on the use of the shallow linguistic kernel-method [2] which has successfully been applied to the extraction of protein–protein interactions (PPIs) and other relations in newspaper texts [6]. As it will be seen in the following sections, the shallow linguistic kernel is based on two configuration parameters, *n*-gram and window-size. In this article we evaluate whether kernel performance is robust across different domains and study the effect of the above-mentioned configuration parameters on the results. In order to train and evaluate our system, we have developed the first annotated

* Corresponding author. Fax: +34 91 624 91 29.

E-mail addresses: isegura@inf.uc3m.es (I. Segura-Bedmar), pmf@inf.uc3m.es (P. Martínez), cdepablo@inf.uc3m.es (C. de Pablo-Sánchez).

1532-0464/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2011.04.005

corpus with DDIs, the DrugDDI corpus. It is our belief that the DrugDDI corpus will help support and evaluate the long-term improvement of technology in drug information management.

The paper is organized as follows: Section 2 reviews the principal approaches developed for the extraction of biomedical relations as well as related work on accessing pharmacological information for specific drugs. We describe our proposal in Section 3, detailing from Subsections 3.4.1, 3.4.2, 3.4.3, 3.4.4 the experiments conducted on DDI extraction from biomedical texts and the results obtained. Finally, Section 4 discusses principal conclusions drawn from the experiments as well as proposals for future research.

2. Related work

The goal of biomedical relation extraction is to detect occurrences of a predefined type of relationship between a pair of given entity types (e.g., genes, proteins or drugs) in text. These relationships may be very specific such as protein interactions (PPIs), pharmacokinetic interactions between drugs or relationships between genes and diseases. Although relationships can generally involve three or more entities (e.g., drug-gene-disease relationships), most of the existing approaches in relation extraction have focused on the extraction of binary relationships. Typically, resulting data from this task is stored in knowledge bases, which can either be consulted directly by users or exploited by data mining algorithms to infer new knowledge. Relation detection may also help to enhance the presentation and results of specialized search engines for end users.

Different techniques have been proposed for the extraction of biomedical relations, particularly PPIs, from texts. Current methods for biomedical relation extraction (and relation extraction, in general) may be classified in three main categories: linguistic-based, pattern-based and machine learning-based approaches.

In *linguistic-based approaches*, linguistic technology is employed to capture syntactic structures or semantic meanings that could be helpful for the discovery of relations in unstructured texts. *Pattern-based approaches* design a set of domain-specific rules (also called 'patterns') that encode and capture the various forms in which a given relationship is expressed. In general terms, the linguistic-based approaches perform well for capturing relatively simple binary relationships between entities in a sentence, but fail to extract more complex relationships expressed in various coordinate and relational clauses [7]. Pattern-based approaches usually achieve high precision, but low recall. Additionally, they are incapable of handling long and complex sentences that are so common in biomedical texts.

As opposed to the previous approaches which require a laborious effort in order to define grammars or a set of rules, *machine learning-based approaches* enable the learning of meaningful models from annotated corpora. These approaches can be further classified as either *feature-based* and *kernel-based* depending on the manner in which instances are represented.

In *feature-based approaches*, relation instances are represented by a set of feature values. Two categories of features are usually selected: (1) properties of single tokens including entity type, Parts-of-Speech tag, lemma and other attributes of the tokens, and (2) relations between tokens represented as the binary presence of certain sequences, parse trees or dependency relations between tokens. In Katrenko and Adriaans [8], dependency parsing and three machine learning algorithms (i.e.,naïve Bayes, BayesNet and K nearest neighbors classifiers) were performed on the AImed [9] and LLL corpora to detect PPIs. Precision ranged between 56% and 81% and recall between 32% and 76% depending on the corpus and classifier used. The highest performance was achieved with the combination of the three classifiers on the AImed corpus (*F*-measure 72.7%). BioPPISVMExtractor [10] is a system for PPIs extraction based on Support Vector Machines (SVMs) and the link grammar parser [11]. The set of features included surface word, keyword, protein name distance and link path features. The system was trained with the IEAP corpus [12] and tested on the DIP [13] corpus. The system achieved a recall of 70.04%, a precision of 49.28% and an *F*-measure of 57.85%.

In *kernel-based approaches*, relation instances are encoded as structural representations such as bag-of-words, word-sequence, parse trees or dependency graphs to measure the similarity between them. A word-sequence kernel can be defined as the number of common word subsequences between two relation instances. Kernels do not need to represent each data instance onto a flat set of features, but rather require a similarity measure between instances [14]. Moreover, several representations may be combined by the use of composite kernels by operations like normalization, scaling, linear combination or product.

In the last decade, several kernel-based methods have been proposed to solve the problem of relation extraction in journalistic texts. In Zelenko and Aone [15], several tree kernels were adapted for the relation extraction task to calculate the similarity of shallow parse trees including head, PoS and entity tags annotations. A relation instance was represented as the smallest shallow subtree containing both entities in the relation. In Culotta and Sorensen [16], an extension of the previous approach was proposed with a richer sentence representation and through the use of composite kernels to reduce kernel sparsity. Bunescu and Mooney [17] designed a kernel-method using the shortest path between the two entities in a dependency tree.

Recent years have seen these kernel-based methods applied to the biomedical relation, as well. In Bunescu and Mooney [18], a generalization of a sequence kernel was proposed using sequences containing words and word classes. Experiments were performed for extracting PPIs from biomedical corpora (AImed and LLL) and top-level relations from newspaper corpora. For PPI extraction, experiments conducted with the AImed corpus yielded an *F*-measure = 59% (precision = 60.0%, recall = 57.2%) while LLL corpus experiments achieved higher performance with an *F*-measure of 61.7% (precision = 62.1%, recall = 61.3%). Later experiments showed that subsequence kernels performed better than the shortest path kernel [19].

Based on the work above, Giuliano et al. [2] proposed a composite sequence kernel, the shallow linguistic kernel, that uses the local context of entities and the global context of their relation to perform the classification. It was evaluated on the AImed (*F*-measure = 63.9%) and LLL (*F*-measure = 58.6%) corpora. We describe this kernel in more detail in Section 3.

Li et al. [20] compared different kernels for biomedical relation extraction: a bag-of-words kernel, a subsequence kernel and the tree kernel proposed in Zelenko and Aone [15]. A tree kernel augmented with the trace from the root node of the smallest subtree to the root of the full parse tree was also proposed. To evaluate the kernels, a corpus was built of 2000 cancer-related abstracts from Medline and a total of 8071 relation instances, 2156 of which being identified as true relations. Best results were yielded by the composite kernel combining the sequence kernel and the trace-tree kernel, achieving an *F*-measure of 67.23% (recall = 64.68%, precision = 70.11%, accuracy = 83.14%).

In Airola et al. [21], a dependency-path kernel was proposed to extract PPIs. Each relation instance was represented with a weighted graph consisting of two unconnected subgraphs, one representing the dependency structure of the sentence and the other the linear order of the words. Experiments were performed across five corpora annotated for PPIs (AImed, LLL, IEAP, BioInfer [22], HPRD50 [23], DIP and BioCreAtIvE-PPI [24]) and demonstrated that *F*-measures varied remarkably across the different corpora. An important variable is the proportion of positive and negative examples in each corpora (i.e., where positive and negative examples involve the respective existence and non-existence of a PPI between proteins in a sentence). The highest *F*-measure (56.4%) was achieved on the AImed corpus. The above five corpora were unified in a common format in Pyysalo et al. [25].

Recently, Tikk et al. [26] compared some of the above-mentioned convolution kernels (i.e., kernels based on the use of parse trees or dependency graphs of sentences) for PPI extraction. Experiments were performed on the above five gold standard corpora, using different parameters and different evaluation metrics. They showed that even the best kernel is not significantly better than the RelEx system [26], a rule-based method not requiring any training or parameter tuning.

2.1. Extracting drug information

The recognition of drug names is an essential prerequisite step for the automatic discovery of DDIs from biomedical texts. While many studies in biomedical named entity recognition have focused on genes and proteins [27–33], only a few have addressed drug names [34–36]. The DrugNer system [37] is a hybrid method that combines semantic information provided by the Unified Medical Language System (UMLS) MetaMap Transfer (MMTx) tool [38] and a set of affixes recommended by the WHOINN program to identify and classify drug names. The affixes enable the recognition of drugs not detected by MMTx, and establish important information such as drug families. Although experiments showed that affixes alone are not sufficient enough for the detection of drugs, they do help slightly improve coverage.

In our previous study [1], we proposed a set of syntactic patterns to split long sentences into clauses from which DDIs were extracted by a pattern matching algorithm. Recently, Garcia-Blasco et al. [41] proposed a method to detect DDI sentences based on maximal frequent sequences.

In Kolarik et al. [39], the use of lexico-syntactic patterns was proposed for the identification and extraction of relevant information on pharmacological properties. The goal of the system was to support database content update by providing additional descriptions of pharmacological effects not found in databases like DrugBank [40].

In Duda et al. [4], a corpus of 2000 abstracts of positive and negative drug interaction citations was manually created in order to evaluate the use of an SVM for locating articles about DDIs. Nevertheless, the goal in this particular case was not the extraction of relations, but rather the classification of articles reporting some kind of interaction. A similar purpose was pursued by Rubin et al. [42] in which an automated method was developed for the identification of articles in Medline citations containing gene-drug relationships. In that study, three types of statistical models (i.e., naïve Bayes, logistic regression, and a log-likelihood) and a heuristic method (i.e., a ‘gene-drug filter’) were implemented to detect pharmacogenetics articles. A sampling of the articles identified from the Medline scans was then reviewed by a pharmacologist to assess the performance of the method. The system achieved an *F*-measure of 88% with a precision of 80% and a recall of 97%.

More recently, several machine learning algorithms were evaluated in Danger et al. [43] in order to obtain a satisfactory classifier for the identification of drug target articles. Best results were achieved by a fuzzy lattice reasoning classifier, reaching 98% of ROC area under curve measurement.

2.2. Discussion

Although several works have applied text mining to related problems in the pharmacological domain, none have carried out research specifically on DDI extraction.

With regard to the different approaches to relation extraction presented above, while hand-built patterns and linguistic-based approaches achieve strong performance, it is also essential that domain experts get involved in the definition of these patterns and the development of these linguistic tools. Such tasks require labor-intensive manual processing with resulting patterns and tools that are unable to easily adapt to other subdomains. Machine-learning approaches, on the other hand, can be easily extended to new domains or relations when annotated corpora to support their training are created.

Certain studies [15,20] have declared that tree kernels not only outperform feature-based methods, but also achieve better results than sequence kernels. It is difficult, however, to reach a conclusion in this case since experiments have been performed on different corpora with different distributions and different experimental conditions. Conversely, in Tikk et al. [26] convolution kernels were shown to provide no significant improvement when compared to rule-based methods. One final issue to consider is the computational complexity of tree kernels [17,20] which may render them inappropriate for practical purposes.

Another important issue is that some research on relation extraction has evaluated only the relation detection step (i.e., assuming perfect linguistic analysis and entity identification), while others have presented results of a complete system that may have potentially included cascading errors produce by NERC, PoS taggers and parsers. For example, one of the advantages of the shallow linguistic kernel [6] is that it has been shown to be robust to noise generated through the use of NERC output.

In this paper, our ultimate goal is to compare the performance of a machine learning method with that of our previous approach [1] based on the use of patterns and yielding an *F*-measure in experiments of only 33.64%. Following consideration of the issues described in this section, we have selected sequence kernels and, in particular, the shallow linguistic kernel proposed in Giuliano et al. [2] as the DDI extraction method to be studied. Motivating the selection was our use of the UMLS MetaMap Transfer (MMTx) tool [38] to analyze the DrugDDI corpus (MMTx only provides shallow syntactic and semantic information) and the findings in Tikk et al. [26] that the kernel was nearly as good as the best dependency-based kernels.

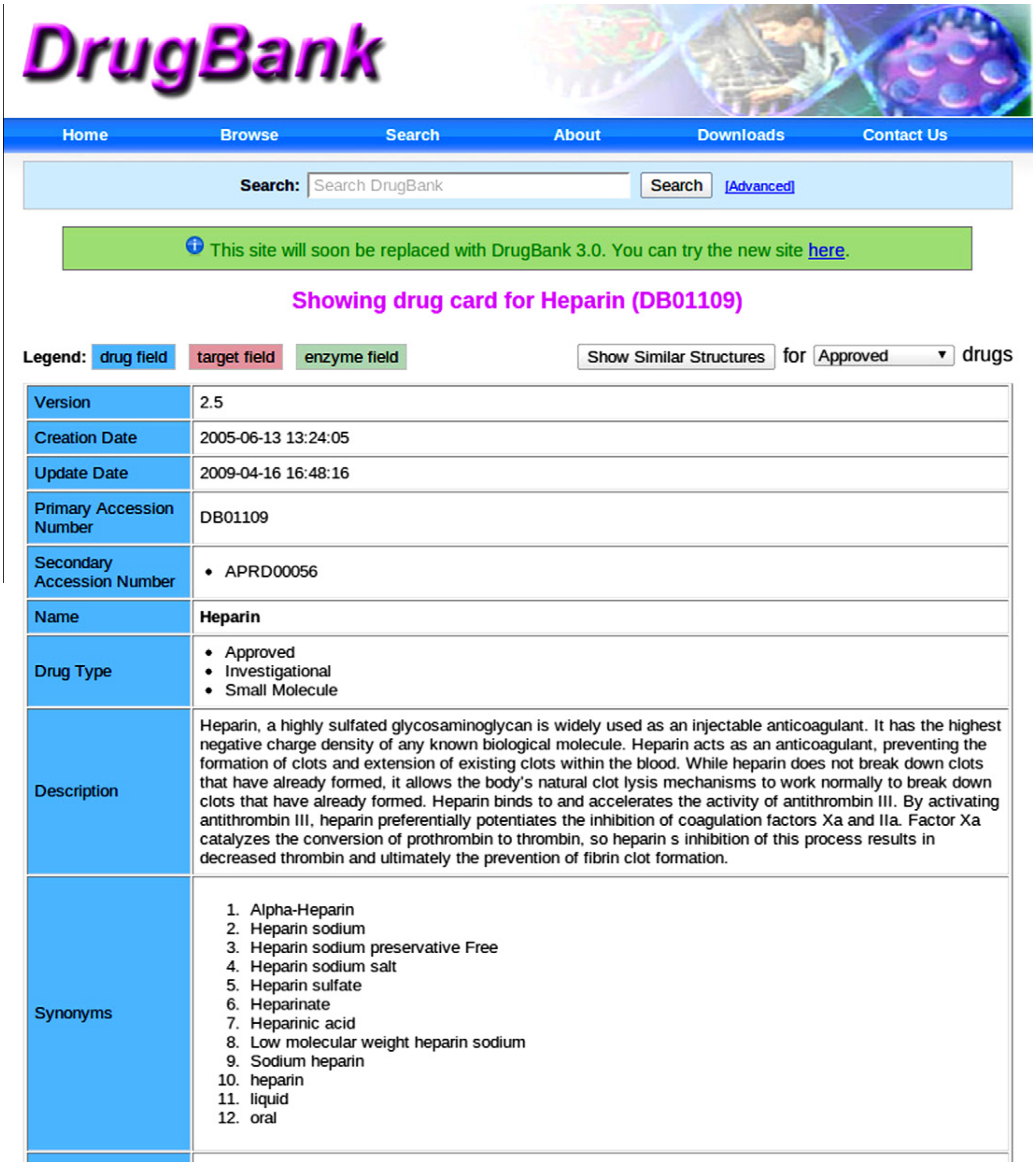
3. Method

The main goal of this work is the automatic extraction of DDIs from texts. We address the problem using an IE method based on supervised machine learning and kernel-methods. To train and evaluate our approach, we developed and used a corpus of text containing potential DDIs. This corpus is described in Subsection 3.1. Subsection 3.3 briefly describes the details of the kernel that we selected, the shallow linguistic kernel initially proposed in Giuliano et al. [2] and used here. Results and experiments regarding the different models have been outlined in Subsection 3.4. This subsection also includes additional experiments performed in response to the imbalanced nature of the data.

3.1. Dataset

In certain studies [44,22,9,24], the biomedical corpora presented focused on the description of several relationships between biological entities. None, however, contained DDIs.

While Natural Language Processing(NLP) techniques are relatively domain-portable, corpora are not. For this reason, we created the first annotated corpus, the DrugDDI corpus, studying the phenomenon of interactions among drugs. This corpus allows us to automatically evaluate our approach for extracting DDIs from



This site will soon be replaced with DrugBank 3.0. You can try the new site here.

Showing drug card for Heparin (DB01109)

Version	2.5
Creation Date	2005-06-13 13:24:05
Update Date	2009-04-16 16:48:16
Primary Accession Number	DB01109
Secondary Accession Number	• APRD00056
Name	**Heparin**
Drug Type	• Approved • Investigational • Small Molecule
Description	Heparin, a highly sulfated glycosaminoglycan is widely used as an injectable anticoagulant. It has the highest negative charge density of any known biological molecule. Heparin acts as an anticoagulant, preventing the formation of clots and extension of existing clots within the blood. While heparin does not break down clots that have already formed, it allows the body's natural clot lysis mechanisms to work normally to break down clots that have already formed. Heparin binds to and accelerates the activity of antithrombin III. By activating antithrombin III, heparin preferentially potentiates the inhibition of coagulation factors Xa and IIa. Factor Xa catalyzes the conversion of prothrombin to thrombin, so heparin s inhibition of this process results in decreased thrombin and ultimately the prevention of fibrin clot formation.
Synonyms	1. Alpha-Heparin 2. Heparin sodium 3. Heparin sodium preservative Free 4. Heparin sodium salt 5. Heparin sulfate 6. Heparinate 7. Heparinic acid 8. Low molecular weight heparin sodium 9. Sodium heparin 10. heparin 11. liquid 12. oral

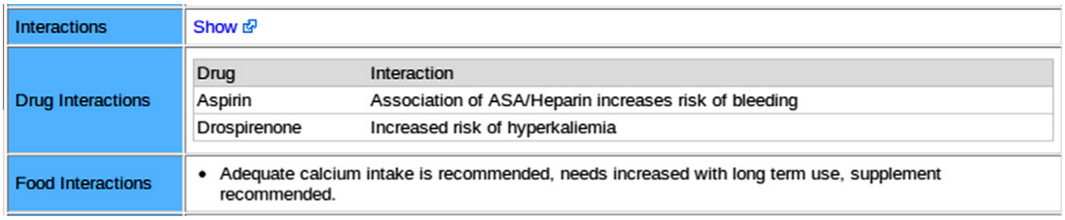
Fig. 1. DrugBank card for Heparin.

biomedical texts. Moreover, we believe that the corpus may also serve to encourage the NLP community to conduct further research in the field of pharmacology.

We used the DrugBank database [40] as the source of unstructured textual information on drugs and their interactions. DrugBank is a chemical and pharmaceutical database containing information of approximately 4900 pharmacological substances. This database provides information oriented to biochemists and biologists regarding the nomenclature, structure and physical properties of drugs and their drug targets. DrugBank also offers detailed clinical information often used by healthcare professionals about drugs including pharmacology, metabolism and indications. The database has enjoyed wide use in several contexts including drug design, drug target discovery or drug interaction prediction, among many other applications. Furthermore, it is a free, online resource.

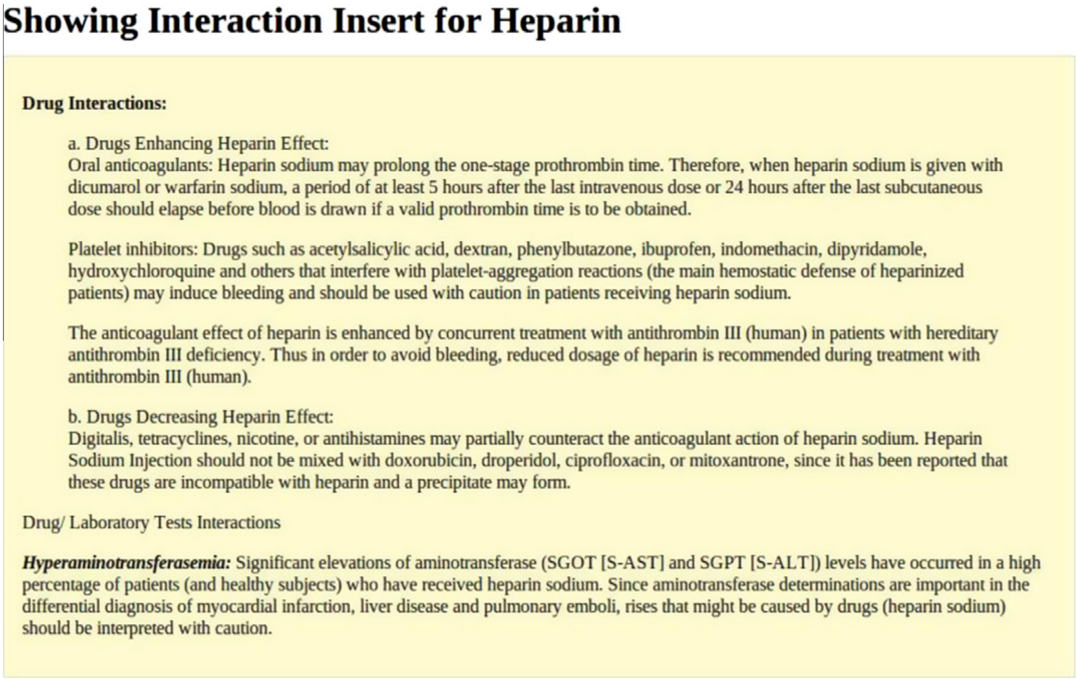
For each drug, DrugBank contains more than 100 data fields including drug synonyms, brand names, chemical formula and structure, drug categories, ATC and AHFS codes (i.e., codes of standard drug families), mechanism of action, indication, dosage forms and toxicity (see Fig. 1). Of particular interest to this study, DrugBank offers a complete collection of DDIs, which was compiled from several resources, checked by an accredited pharmacist and entered manually into the database. This collection consists of 714 food interactions and 13,242 drug–drug interactions contained respectively in the structured information fields, food interactions and drug interactions (see Fig. 2).

Additional information can be found in the field ‘interactions’ (see Fig. 3), containing a link to a document describing DDIs in unstructured texts. This document not only contains a detailed description of the interactions contained in the above-mentioned fields (i.e., ‘food interactions’ and ‘drug interactions’), but also offers information on other interactions not included therein. For the present study, we used the ‘interactions’ field as a source of unstructured textual information on DDIs. We believe that these texts are a reliable and representative source of data for expressing DDIs since the language used is mostly devoted to descriptions of DDIs. Additionally, the highly specialized pharmacological



Interactions	Show
Drug Interactions	Drug – Interaction Aspirin – Association of ASA/Heparin increases risk of bleeding Drospirenone – Increased risk of hyperkaliemia
Food Interactions	• Adequate calcium intake is recommended, needs increased with long term use, supplement recommended.

Fig. 2. Food and drug interaction fields in Heparin drug card.



Showing Interaction Insert for Heparin

Drug Interactions:

a. Drugs Enhancing Heparin Effect:
Oral anticoagulants: Heparin sodium may prolong the one-stage prothrombin time. Therefore, when heparin sodium is given with dicumarol or warfarin sodium, a period of at least 5 hours after the last intravenous dose or 24 hours after the last subcutaneous dose should elapse before blood is drawn if a valid prothrombin time is to be obtained.

Platelet inhibitors: Drugs such as acetylsalicylic acid, dextran, phenylbutazone, ibuprofen, indomethacin, dipyridamole, hydroxychloroquine and others that interfere with platelet-aggregation reactions (the main hemostatic defense of heparinized patients) may induce bleeding and should be used with caution in patients receiving heparin sodium.

The anticoagulant effect of heparin is enhanced by concurrent treatment with antithrombin III (human) in patients with hereditary antithrombin III deficiency. Thus in order to avoid bleeding, reduced dosage of heparin is recommended during treatment with antithrombin III (human).

b. Drugs Decreasing Heparin Effect:
Digitalis, tetracyclines, nicotine, or antihistamines may partially counteract the anticoagulant action of heparin sodium. Heparin Sodium Injection should not be mixed with doxorubicin, droperidol, ciprofloxacin, or mitoxantrone, since it has been reported that these drugs are incompatible with heparin and a precipitate may form.

Drug/ Laboratory Tests Interactions

Hyperaminotransferasemia: Significant elevations of aminotransferase (SGOT [S-AST] and SGPT [S-ALT]) levels have occurred in a high percentage of patients (and healthy subjects) who have received heparin sodium. Since aminotransferase determinations are important in the differential diagnosis of myocardial infarction, liver disease and pulmonary emboli, rises that might be caused by drugs (heparin sodium) should be interpreted with caution.

Fig. 3. Interactions field linked document from Heparin drug card describing DDIs.

language is very similar to that found in the Medline pharmacology abstracts.

Due to the cost-intensive and time consuming nature of the annotation process, we randomly selected a subset of 579 documents to be annotated for the present study. We used the Kapow's free RoboMaker screen-scraper[1] to download the interaction documents. These documents were then analyzed by the UMLS MetaMap Transfer (MMTx) tool performing sentence splitting, tokenization, POS-tagging, shallow syntactic parsing (see Fig. 6) and linking of phrases with UMLS Metathesaurus concepts.

Fig. 4 shows part of the output produced by MMTx for a given document. The output of MMTx is transformed into XML format providing maximum flexibility for the use of the DrugDDI corpus. In this transformation process, MMTx first splits the text into sentences. The SPECIALIST minimal commitment parser [45] is then used to produce a shallow syntactic parsing of the texts where phrases in each sentence are identified and classified. The resulting XML document gives the type, number of tokens, text and an identifier for each phrase.

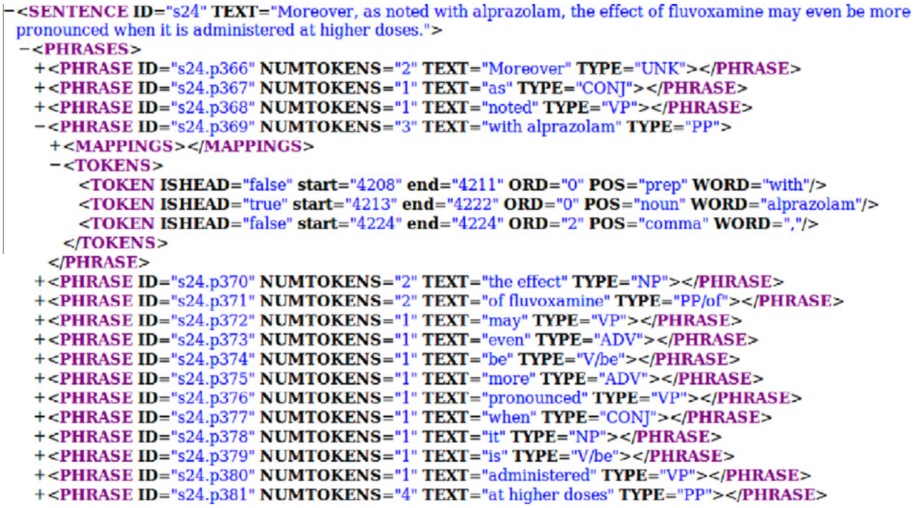
The parser then uses the SPECIALIST lexicon to assign POS tags to the tokens, relying on the Xerox part-of-speech tagger [46] in order to determine on the correct tag when a token has several tags in the lexicon. Each token is annotated with its POS tag, word, and a boolean value indicating if it is the head of the phrase (i.e., the attribute 'ISHEAD'). In addition, the start and end offsets of each token within the text are stored in the attributes, 'start' and 'end', respectively. These character offsets enable the mapping from the annotation to the raw text. Fig. 4 shows the tokens and their offsets contained in the phrase 'with alprazolam,'.

For each phrase, a set of variants is generated using the SPECIALIST lexicon and linguistic techniques. These variants are the text of the phrase plus its acronyms, abbreviations and synonyms, as well its derivational, inflectional and spelling variants. These variants are then searched for in the UMLS Metathesaurus, retrieving those concepts containing at least one of them. Each concept is evaluated against the text of the phrase using several linguistic metrics to determine its similarity. Finally, those concepts with the highest similarity are selected as the final mapping. A more detailed description of this process can be found in Aronson [47]. For each concept in the final mapping set, MMTx provides its concept unique identifier (CUI), concept name and semantic types.

In this way, drugs are automatically identified by MMTx since the tool allows for the recognition and annotation of biomedical entities occurring in texts according to the UMLS semantic types (e.g., Clinical Drug [clnd], Pharmacological Substance [phsu], Antibiotic [antb]). As an example, Fig. 5 shows that the phrase 'Aspirin' is classified with the semantic type 'pharmacological substances (phsu)'.

The principal value of the DrugDDI corpus undoubtedly comes from its DDIs annotations. To obtain these annotations, all docu-

[1] http://openkapow.com/.



```
-<SENTENCE ID="s24" TEXT="Moreover, as noted with alprazolam, the effect of fluvoxamine may even be more
 pronounced when it is administered at higher doses.">
 -<PHRASES>
   +<PHRASE ID="s24.p366" NUMTOKENS="2" TEXT="Moreover" TYPE="UNK"></PHRASE>
   +<PHRASE ID="s24.p367" NUMTOKENS="1" TEXT="as" TYPE="CONJ"></PHRASE>
   +<PHRASE ID="s24.p368" NUMTOKENS="1" TEXT="noted" TYPE="VP"></PHRASE>
   -<PHRASE ID="s24.p369" NUMTOKENS="3" TEXT="with alprazolam" TYPE="PP">
     +<MAPPINGS></MAPPINGS>
     -<TOKENS>
         <TOKEN ISHEAD="false" start="4208" end="4211" ORD="0" POS="prep" WORD="with"/>
         <TOKEN ISHEAD="true" start="4213" end="4222" ORD="0" POS="noun" WORD="alprazolam"/>
         <TOKEN ISHEAD="false" start="4224" end="4224" ORD="2" POS="comma" WORD=","/>
       </TOKENS>
     </PHRASE>
   +<PHRASE ID="s24.p370" NUMTOKENS="2" TEXT="the effect" TYPE="NP"></PHRASE>
   +<PHRASE ID="s24.p371" NUMTOKENS="2" TEXT="of fluvoxamine" TYPE="PP/of"></PHRASE>
   +<PHRASE ID="s24.p372" NUMTOKENS="1" TEXT="may" TYPE="VP"></PHRASE>
   +<PHRASE ID="s24.p373" NUMTOKENS="1" TEXT="even" TYPE="ADV"></PHRASE>
   +<PHRASE ID="s24.p374" NUMTOKENS="1" TEXT="be" TYPE="V/be"></PHRASE>
   +<PHRASE ID="s24.p375" NUMTOKENS="1" TEXT="more" TYPE="ADV"></PHRASE>
   +<PHRASE ID="s24.p376" NUMTOKENS="1" TEXT="pronounced" TYPE="VP"></PHRASE>
   +<PHRASE ID="s24.p377" NUMTOKENS="1" TEXT="when" TYPE="CONJ"></PHRASE>
   +<PHRASE ID="s24.p378" NUMTOKENS="1" TEXT="it" TYPE="NP"></PHRASE>
   +<PHRASE ID="s24.p379" NUMTOKENS="1" TEXT="is" TYPE="V/be"></PHRASE>
   +<PHRASE ID="s24.p380" NUMTOKENS="1" TEXT="administered" TYPE="VP"></PHRASE>
   +<PHRASE ID="s24.p381" NUMTOKENS="4" TEXT="at higher doses" TYPE="PP"></PHRASE>
```

Fig. 4. Example of document processed by MMTx.

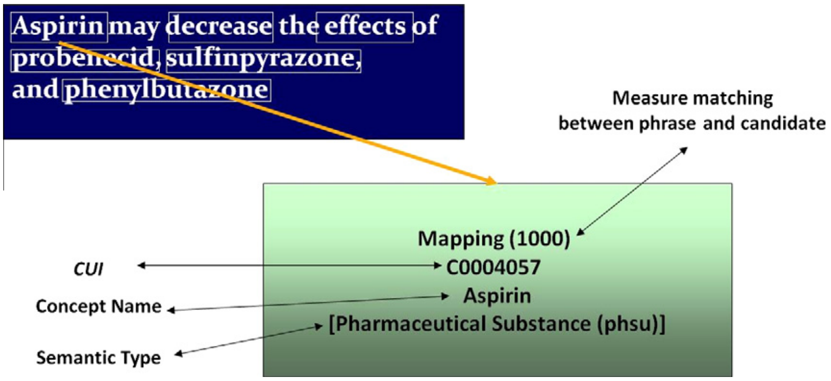
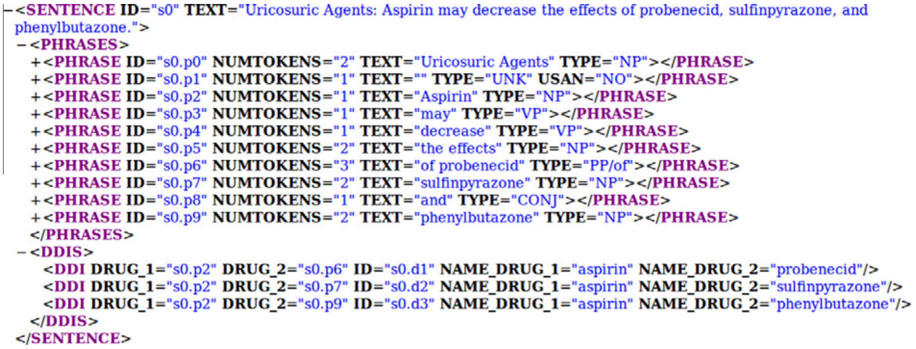

Aspirin may decrease the effects of probenecid, sulfinpyrazone, and phenylbutazone
Measure matching between phrase and candidate
Mapping (1000)
CUI
C0004057
Concept Name
Aspirin
Semantic Type
[Pharmaceutical Substance (phsu)]

Fig. 5. Mapping for the phrase, ‘Aspirin’.



```
-<SENTENCE ID="s0" TEXT="Uricosuric Agents: Aspirin may decrease the effects of probenecid, sulfinpyrazone, and
 phenylbutazone.">
 -<PHRASES>
   +<PHRASE ID="s0.p0" NUMTOKENS="2" TEXT="Uricosuric Agents" TYPE="NP"></PHRASE>
   +<PHRASE ID="s0.p1" NUMTOKENS="1" TEXT="" TYPE="UNK" USAN="NO"></PHRASE>
   +<PHRASE ID="s0.p2" NUMTOKENS="1" TEXT="Aspirin" TYPE="NP"></PHRASE>
   +<PHRASE ID="s0.p3" NUMTOKENS="1" TEXT="may" TYPE="VP"></PHRASE>
   +<PHRASE ID="s0.p4" NUMTOKENS="1" TEXT="decrease" TYPE="VP"></PHRASE>
   +<PHRASE ID="s0.p5" NUMTOKENS="2" TEXT="the effects" TYPE="NP"></PHRASE>
   +<PHRASE ID="s0.p6" NUMTOKENS="3" TEXT="of probenecid" TYPE="PP/of"></PHRASE>
   +<PHRASE ID="s0.p7" NUMTOKENS="2" TEXT="sulfinpyrazone" TYPE="NP"></PHRASE>
   +<PHRASE ID="s0.p8" NUMTOKENS="1" TEXT="and" TYPE="CONJ"></PHRASE>
   +<PHRASE ID="s0.p9" NUMTOKENS="2" TEXT="phenylbutazone" TYPE="NP"></PHRASE>
   </PHRASES>
 -<DDIS>
     <DDI DRUG_1="s0.p2" DRUG_2="s0.p6" ID="s0.d1" NAME_DRUG_1="aspirin" NAME_DRUG_2="probenecid"/>
     <DDI DRUG_1="s0.p2" DRUG_2="s0.p7" ID="s0.d2" NAME_DRUG_1="aspirin" NAME_DRUG_2="sulfinpyrazone"/>
     <DDI DRUG_1="s0.p2" DRUG_2="s0.p9" ID="s0.d3" NAME_DRUG_1="aspirin" NAME_DRUG_2="phenylbutazone"/>
   </DDIS>
 </SENTENCE>
```

Fig. 6. Example of DDI annotations.

ments were marked-up by a researcher with pharmaceutical background. DDIs were annotated at the sentence level and, thus, any interactions spanning over several sentences were not annotated here. For the annotation of interactions, then, the annotator needs only select a sentence and indicate the interacting drugs. The annotator should then annotate an interaction for each pair of interacting drugs.

Fig. 6 shows an example of an annotated sentence in our XML format containing three interactions. Each interaction is represented as a DDI node in which the names of the interacting drugs are registered in its NAME_DRUG_1 and NAME_DRUG_2 attributes. The identifiers of the phrases (i.e., ‘DRUG_1’ and ‘DRUG_2’) containing these interacting drugs were also provided to enable access to the related concepts provided by MMTx.

Table 1 shows basic statistics of the DrugDDI corpus. In general, the size of biomedical corpora is quite small and usually does not exceed 1000 sentences. The average number of sentences per MedLine abstract was estimated at 7.2 ± 1.9 [48]. Our corpus contains 5806 sentences with 10.3 sentences per document on average. MMTx identified a total of 66,021 phrases of which 12.5% (8260) are drugs. The average number of drug mentions per document was 24.9, and the average number of drug mentions per sentence

Table 1
Basic statistics on the DrugDDI corpus.

	Number	Avg. per document
Documents	579	
Sentences	5806	10.03
Phrases	66,021	114.02
Tokens	127,653	220.47
Sentences with at least one DDI	2044	3.53
Sentences with no DDI	3762	6.50
DDIs	3160	5.46 (0.54 per sentence)

was 2.4. The corpus contains a total of 3775 sentences with two or more drug mentions, although only 2044 sentences contain at least one interaction. With the assistance of a pharmacist, a total of 3160 DDIs were with an average of 5.46 DDIs per document and 0.54 per sentence. The DrugDDI corpus is available for research purposes at http://labda.inf.uc3m.es/DrugDDI/.

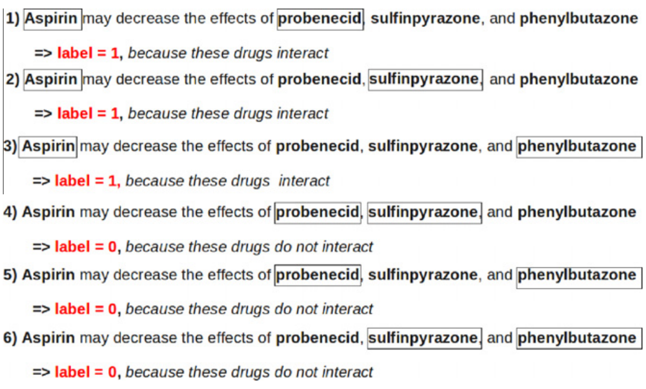
3.2. DDI relation extraction as a classification task

In our approach, DDI extraction is formulated as a supervised learning problem, more particularly, as a drug pair classification task. Therefore, a crucial step is to generate suitable datasets to train and test a classifier from the DrugDDI corpus. The simplest way to generate examples to train a classifier for a specific relation *R* is to enumerate all possible ordered pairs of sentence entities. In our study, we proceeded in a similar way. Given a sentence *S* with at least two drugs, we defined *D* as the set of drugs in *S* and *N* as the number of drugs. The set of examples generated for *S*, therefore, was defined as follows: $\{(D_i,D_j)\colon D_i,D_j \epsilon D, 1 \leqslant i,j \leqslant N, i \neq j, i < j\}$.

If the interaction existed between the two DDI candidate drugs, then the example was labeled 1. Otherwise, it was labeled 0. Although some DDIs may be asymmetrical, the roles of the interacting drugs were not included in the corpus annotation and are not specifically addressed in this article. As a result, we enumerate candidate pairs here without taking their order into account, such that (D_i,D_j) and (D_j,D_i) are considered as a single candidate pair.

Since the order of the drugs in the sentence was not taken into account, each example is the copy of the original sentence *S* where the candidates were assigned the tag, 'DRUG', and remaining drugs were assigned the tag, 'OTHER'. The set of possible candidate pairs was the set of 2-combinations from the whole set of drugs appearing in *S*. Thus, the number of examples was $C_{N,2} = \binom{N}{2}$.

The sentence shown in Fig. 7 contains four drugs: 'aspirin', 'probenecid', 'sulfinpyrazone' and 'phenylbutazone'. Therefore, the total number of examples generated is $C_{4,2} = \binom{4}{2} = 6$.



1) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 1, *because these drugs interact*

2) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 1, *because these drugs interact*

3) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 1, *because these drugs interact*

4) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 0, *because these drugs do not interact*

5) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 0, *because these drugs do not interact*

6) **Aspirin** may decrease the effects of **probenecid**, **sulfinpyrazone**, and **phenylbutazone**

=> label = 0, *because these drugs do not interact*

Fig. 7. Labeling candidate drugs.

Table 2
Distribution of positive and negative examples in training and testing datasets.

Set	Documents	Examples	Positives	Negatives
Train	437 (75.5%)	25,209	2421 (9.6%)	22,788 (90.4%)
Final test	142 (24.5%)	5548	739 (13.3%)	4809 (86.7%)
Total	579	30,757	3160 (10.27%)	27,597 (89.73%)

Table 3
Training and testing datasets.

Set	Documents	Sentences	Drugs	DDIs
Training	437	4578	2560	2421
Final test	142	1228	753	739
Total	579	5806	3313	3160

Table 2 shows the total number of relation examples or instances generated from the DrugDDI corpus. In our corpus consisting of a total of 5806 sentences (see Table 1), we considered only those sentences with at least two drugs, obtaining 3775 sentences with 3313 different drug types. Among the 30,757 candidate drug pairs, only 3160 (10.27%) were marked as positive interactions (i.e., DDIs) while 27,597 (89.73%) were marked as negative interactions (i.e., non-DDIs).

Once we generated the set of relation instances from the DrugDDI corpus, the set was then split in order to build the datasets for the training and evaluation of the different DDI extraction models. In order to build the training dataset used for development tests, 75.5% of the DrugDDI corpus files (437 files) were randomly selected. The remaining 24.5% (142 files) was used in the final evaluation to determine which model was superior. Table 3 shows the distribution of the documents, sentences, drugs and DDIs in each set. Approximately 90% of the instances in the training dataset were negative examples (i.e., non-DDIs). The distribution between positive and negative examples in the final test dataset was also quite similar (see Table 2).

With our modeling, we treated DDI extraction as a classification problem between pairs of entities having been annotated as drugs. Relation extraction was performed using the MMTx annotation including tokenization, POS-tagging, lemmatization and chunking. In addition, it also provides semantic annotation by linking phrase concepts to UMLS concepts. For the present study, we chose the shallow linguistic kernel originally proposed in Giuliano et al. [2] due to its strong performance using only shallow linguistic information and its robustness in the face of annotation errors, such as the incorrect identification of named entities and their boundaries.

3.3. Shallow linguistic kernel

Machine-learning classifiers try to find optimal frontiers between classes. When the instances of classes are not linearly separable, kernel methods can transform the problem space to a higher dimensional space, in which the instances might be separable. Formally, a kernel function is a binary function $K{:}X \times X \rightarrow [0,\infty)$ that maps a pair of instances x, $y \in X$ to their similarity score $K(x,y)$. The kernel function must satisfy the following:

$$\forall x,y \in X : K(x,y) = \langle \phi(x), \phi(y) \rangle, \tag{1}$$

where $\phi : X \rightarrow F \subseteq \mathbb{R}^n$ is a mapping from the input space *X* to a vector space *F*. The mapping function ϕ transforms each instance $x \in X$ into a feature vector $\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_m(x))$, where $\phi_i : X \rightarrow \mathbb{R}$, with no need to know the explicit representation of *x*. Thus, the mapping function ϕ allows $K(x,y)$ to be expressed as the dot-product of the features vectors of the input objects *x* and *y*. The kernel function allows for the computation of the product of the two embedded vectors without requiring any prior knowledge regarding the features of each vector. Global and local context kernels are normalized using Eq. (3) to integrate information from heterogeneous feature spaces (e.g., combining tokenization, PoS tags or entity tags).

$$\forall x, y \in X : K(x,y) = \langle \phi(x), \phi(y) \rangle = \sum_{i=1}^{m} \phi_i(x) \cdot \phi_i(y). \tag{2}$$

$$K(x_i, x_j) = \frac{\langle \phi(x_i), \phi(x_j) \rangle}{\|\phi(x_i)\| \|\phi(x_j)\|} \tag{3}$$

The shallow linguistic kernel (K_{SL}) is a composite kernel defined in Giuliano et al. [2] as the linear combination of two different sequence kernels, a global context kernel (K_{GC}) and local context kernel (K_{LC}). In Eq. (4), R_i and R_j represent examples of two different candidate DDI between drugs.

$$K_{SL}(R_i, R_j) = K_{GC}(R_i, R_j) + K_{LC}(R_i, R_j) \tag{4}$$

Global context kernel. The global context kernel is designed to discover the presence of a relation between two entities by using information from the whole sentence. Its basis can be explained by the following observation from Bunescu and Mooney [18]: "When a sentence asserts a relationship between two entity mentions, it generally does this using one of the following three contexts: fore-between, between, and between-after". In other words, a relationship between two entities is usually expressed using the words appearing before and between the entities (i.e., fore-between pattern [FB]), only between them (i.e., between pattern [B]) or between and after them (i.e., between-after pattern [BA]).

The global context kernel, therefore, is a composite kernel formed by the linear combination of kernels defined for the three contexts relevant for the detection of relations, as shown in following equation.

$$K_{GC}(R_1, R_2) = K_{FB}(R_1, R_2) + K_B(R_1, R_2) + K_{BA}(R_1, R_2) \tag{5}$$

As stated in Giuliano et al. [2], while this kernel uses only lexical tokens or words, it is important to note that its representation nevertheless preserves stop words and punctuation marks since they found to be useful tokens for the relation extraction task. For each of the three patterns described above, the representation uses the term frequency of tokens in the context C, $tf(t_i, C)$. To calculate the similarity between two patterns, the authors proposed the use of the *n*-gram kernel, also known in the literature as the *n*-spectrum kernel [14]. The *n*-gram kernel compares two relation instances by counting the common *n*-grams between the two in each of the three contexts discussed above (i.e., FB, B and BA). Figs. 8 and 9 present examples of a global context kernel calculated with *n*-gram = 1 and *n*-gram = 2, respectively, in order to estimate the similarity between two relation instances. While the 1-g kernel only counts the unigrams in common, the 2-gram kernel scores both unigrams and bigrams.

Local context kernel. The local context kernel is based on the hypothesis that the contextual information of candidate entities is particularly useful for the verification of a relationship existing between them. In particular, windows of limited size around entities provide useful clues for the identification of the entities' roles within a relation.

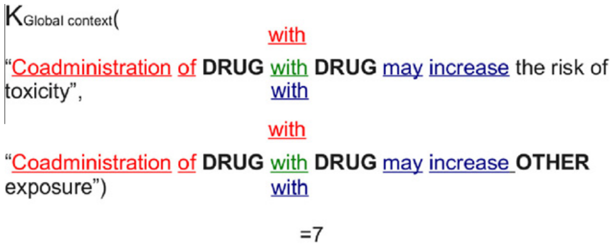

KGlobal context(
with
"Coadministration of DRUG with DRUG may increase the risk of toxicity",
with
with
"Coadministration of DRUG with DRUG may increase OTHER exposure")
with
=7

Fig. 8. Example of global context kernel for *n*-gram = 1.

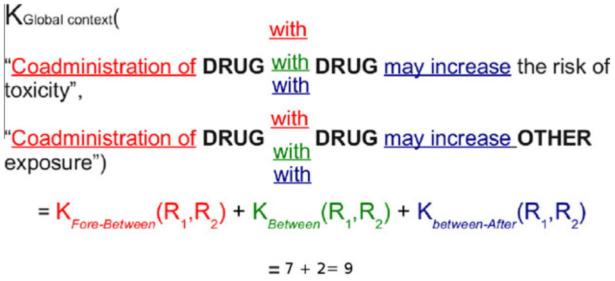

KGlobal context(
with
"Coadministration of DRUG with DRUG may increase the risk of toxicity",
with
with
"Coadministration of DRUG with DRUG may increase OTHER exposure")
with
with
= K_Fore-Between(R1,R2) + K_Between(R1,R2) + K_between-After(R1,R2)
= 7 + 2= 9

Fig. 9. Example of global context kernel for *n*-gram = 2.

Therefore, Giuliano et al. [2] used the information provided by the two local contexts of the candidate interacting entities, called left and right local context, respectively. Each local context was represented using lexical and morphological features such as tokens, lemmas, PoS tags and stems. Each example was basically represented as an instance of the original sentence with the two candidate entities properly annotated (i.e., with the tag 'DRUG' in the case of the present study). The roles of the candidates are labeled with the tags 'A' (agent) and 'T' (target) which, in the case of the present study, were always the first and second arguments, respectively. Any other entity or tokens that were not candidates were labeled 'O'. Since MMTx does not provide lemmatization, our approach here used a 'stem' feature rather than 'lemma'. We obtained stems using the Porter Stemming Algorithm [49]. Then, the local context kernel can be defined as the sum of the left and right context kernels, as shown in the following equation:

$$K_{LC}(R_1, R_2) = K_{left}(R_1, R_2) + K_{right}(R_1, R_2) \tag{6}$$

K_{LC} differs substantially from K_{GC} in that it considers the ordering of tokens and enriches the feature space with PoS tags, lemmas, stems and orthographic features. A more detailed description of both kernels can be found in Giuliano et al. A more detailed description of both kernels can be found in Giuliano et al. [2].

In the same study [*ibid.*], the authors also developed a Java tool for Relation Extraction (jSRE) in order to implement their shallow linguistic kernel. This jSRE implementation used the SVM package LIBSVM [50]. The kernel was represented by a matrix containing the pairwise similarity of all instances. This matrix was then passed over to the LIBSVM, where it was used to learn a classification function (Eq. 7). The SVM classification function takes the form $f(x) = wx - b$ where w is the weight vector and b is the bias computed by the SVM in the training process. When the training data is not linearly separable, linear SVMs must be extended to nonlinear SVMs through the use of a mapping function ϕ transforming the input vectors into high-dimensional feature vectors. Thus, the weight can be reformulated as: $w = \sum \alpha_i y_i \phi(x_i)$ where $\phi(x_i)$ are the support vectors. As the SVM computed the dot-product between instances, it can be generalized to kernels since these functions are defined as the dot-product in some expanded feature space. Hence, applying Eq. (2), the classification function becomes:

$$\begin{aligned} f(x) = wx - b = w\phi(x) - b = \sum \alpha_i y_i \phi(x_i)\phi(x) - b \\ = \sum \alpha_i y_i K_{SL}(x_i, x) - b \end{aligned} \tag{7}$$

In both general and biological domains, jSRE has demonstrated strong performance [2,6]. We conducted a set of experiments to study the new problem of DDI extraction from biomedical texts.

Table 4
Average results shallow linguistic kernels according to parameters. The highest scores are marked with asterisks (∗).

Experiment	Avg. P	Avg. R	Avg. F_1
allDDI	0.11	1	0.19
n-gram = 1, window-size = 1	0.4238	0.7841	0.5490
n-gram = 1, window-size = 2	0.4397	0.7863*	0.5630
n-gram = 1, window-size = 3	0.4551	0.7748	0.5727
n-gram = 2, window-size = 1	0.4693	0.7079	0.5632
n-gram = 2, window-size = 2	0.4914	0.7115	0.5797
n-gram = 2, window-size = 3	0.4887	0.7079	0.5779
n-gram = 3, window-size = 1	0.5040	0.7321	0.5964*
n-gram = 3, window-size = 2	0.5079	0.6963	0.5861
n-gram = 3, window-size = 3	0.5207*	0.6996	0.5964*

3.4. Experiments

This subsection describes the experiments run in the present study to evaluate the effectiveness of the shallow linguistic kernel. Since one of our main objectives was to investigate the influence of the configuration parameters of the jSRE tool – namely, window-size of the local context and n-gram of the global context – on final performance, we designed a set of experiments in which these parameters were varied. Additionally, due to the greately imbalanced nature of the training and test data (i.e., with regard to negative and positive examples), different experiments were also run in an attempt to compensate for this fact.

Starting out, we considered as baseline system, referred to here as *allDDI*, the case in which every relation instance was classified as a DDI (i.e., a positive example). This baseline yielded the maximum recall, but low precision. Evaluated on the test dataset, the baseline system achieved a baseline precision of 11% and F-measure of 19% (see Table 4, row 1).

3.4.1. Kernel selection experiments

In our experiments, we used 10-fold cross-validation on the training dataset. For each run, nine folds were used to train a model that was evaluated with the remaining fold. The folds were built considering that examples from the same sentence must belong to the same fold. We followed the *OAOD* (*One Answer per Ocurrence in the Document*) [51] evaluation methodology, such that, each individual occurrence of a DDI had to be extracted from the document regardless of the number of times it was stated.

For the present study, we ran a number of experiments varying configuration parameter values for the local and global kernels in order to contrast performance trade-offs. For the global kernel, the principal parameter is the n-gram size which we varied here between 1 and 3. For the local kernel, the primary parameter is the size of the window delineating the context around the candidate entities. We varied window-size in equal length from ±1 to ±3. One reason for this selection is that for the two parameters, jSRE implementation does not allow for values superior to 3. The average 10-fold cross-validation results are presented in Table 4 above.

Table 4 and Fig. 10 show performance to differ significantly from one configuration to another with average precision ranging from 42.38% (n-gram = 1, window-size = 1) to 52.07% (n-gram = 3, window-size = 3) and average recall from 69.63% (n-gram = 3, window-size = 2) to 78.63% (n-gram = 1, window-size = 2). Thus, the highest average precision (52.07%) was achieved with an n-gram and window-size of 3 and the highest average recall (78.63%) was achieved with an n-gram of size 1 and window-size of 2. On the contrary, these latter two parameter values (n-gram = 1, window-size = 2) yielded the second lowest average precision (43.97%). The highest average F-measure (59.64%) was achieved with an n-gram of size 3 and a window-size of either 1 or 3.

As parameter values increase, the average precision improved and the average recall declined. Across the experiments generally, a small n-gram size favored the obtainment of a higher recall value while a larger n-gram favored the obtainment of greater precision value. The choice of the parameter window-size, however, does not seem to have significantly affected performance (save for an n-gram of size 3). Such results are coherent with the fact that the window-size parameter is designed to identify the roles of entities within a relation, a consideration not addressed in the context of our current DDI annotation. Among the trained models, we selected the model maximizing both F-measure and precision (n-gram = 3, window-size = 3) in order to avoid overloading database curators with too many false positives during DDI extraction. Nevertheless, it is important to note that a different choice may

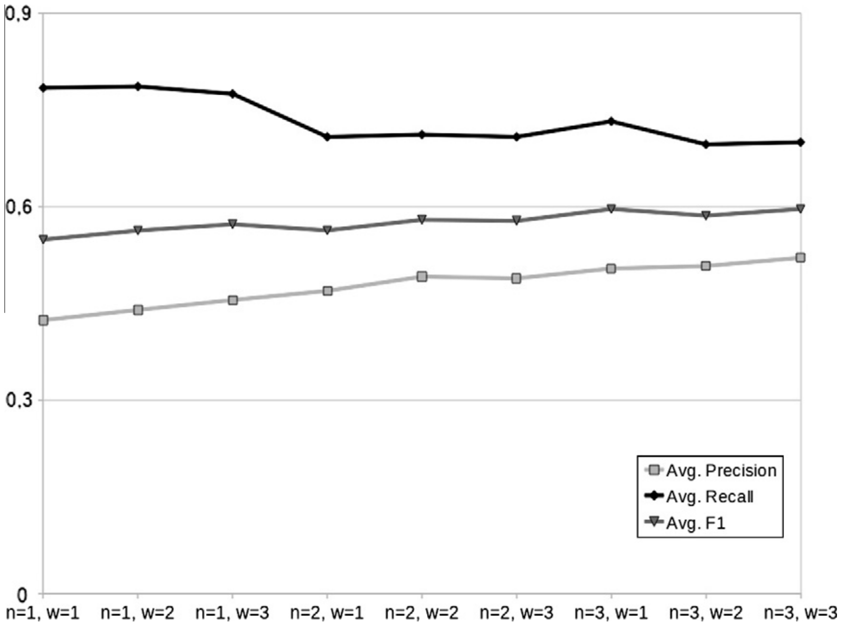


Fig. 10. Average results for shallow linguistic kernel.

Table 5
Comparative analysis of global, local and shallow kernels. The highest scores are marked with asterisks (∗).

Kernel	*P*	*R*	*F*
Global context (*n*-gram = 3)	0.5158	0.7114	0.5974*
Local context (window-size = 2)	0.4387	0.7843*	0.5618
Shallow (*n*-gram = 3, window-size = 3)	0.5207*	0.6996	0.5964

Table 6
Final results obtained by the shallow kernels. The highest scores are marked with asterisks (∗).

	TP	FP	FN	*P*	*R*	*F*
allDDI	739	6270	0	0.11	1	0.19
n-gram = 1, window-size = 1	569	724	178	0.4401	0.7617*	0.5578
n-gram = 1, window-size = 2	555	683	192	0.4483	0.7430	0.5592
n-gram = 1, window-size = 3	552	641	195	0.4627	0.7390	0.5691
n-gram = 2, window-size = 1	562	579	185	0.4926	0.7523*	0.5953
n-gram = 2, window-size = 2	557	580	190	0.4899	0.7456	0.5913
n-gram = 2, window-size = 3	553	557	194	0.4982	0.7403	0.5956
n-gram = 3, window-size = 1	539	568	208	0.4869	0.7216	0.5814
n-gram = 3, window-size = 2	542	549	205	0.4968	0.7256	0.5898
n-gram = 3, window-size = 3	544	522	203	0.5103*	0.7282	0.6001*

have been justified following an exhaustive search in other types of information access applications.

In our kernel evaluations, we evaluated each kernel separately in order to analyze the contributions of the global and local kernels to the overall shallow linguistic kernel. Table 5 presents the results yielded with optimal configurations for each kernel type. Results show that global context is more useful than local context for DDI detection since highest *F*-measure (59.74%) was achieved with the former rather than the latter. Although the local kernel was designed to identify the roles of candidate entities within a relation [2], our results show that the local kernel also positively influences DDI detection since the combination of both kernels improved the precision (52.07%), though also causing a slight decrease in the *F*-measure (59.64%). The model using a global context kernel with *n*-gram = 1 and no local context kernel is very similar to traditional bag-of-words instance representation with an SVM classifier. This configuration showed a precision of 40.18%, a recall of 71.28% and an *F*-measure of 50.78%. These results confirm the usefulness of the composite kernel and, particularly, the advantage obtained by using larger *n*-grams.

Finally, the shallow kernel (trained with *n*-gram = 3 and window-size = 3) was evaluated on the final testing dataset, achieving a precision of 51.03%, a recall of 72.82% and an *F*-measure of 60.01% (see Table 6). With regard to the baseline *F*-measure recorded, these results represent an improvement of 41%. In addition, we also evaluated the other models from Table 10 using the final test dataset. In general, results for each model were similar to those obtained from the 10-fold cross-validation experiments.

Learning curves are useful to show the results achieved in the learning process for different percentages of training documents used. We used the jSRE configuration (*n*-gram = 3 and window-size = 3) having yielded the best results in the previous experiments discussed. In Fig. 11, we calculated the *F*-measure, precision and recall for different percentages of training documents used. As demonstrated in Fig. 11, performance barely improved when the training size was increased to beyond 60% of the training corpus.

Several works have reported that metrics derived from the confusion matrix provide a poor estimate of the performance of a model [52]. Additionally, these metrics are highly sensitive to data anomalies such as class skew. Receiver Operator Characteristic (ROC) curves offer an alternative to traditional metrics since ROC curves describe classifier behavior regardless of class distributions or error costs.

Before offering a definition of ROC curves, it is important to briefly review two performance metrics here. The false positive rate (FPR) is the percentage of negative examples misclassified as positive, whereas the true positive rate (TPR), or recall, measures the fraction of positive examples correctly labeled as such. Typically, ROC graphs are constructed by plotting the TPR along *y*-axis and the FPR along the *x*-axis. ROC curves are able to depict results information in a more robust and intuitive manner than traditional metrics. At the same time, however, ROC curves may also provide a too optimistic view of classifier performance when dealing with highly skewed datasets.

Precision-recall (PR) curves are an alternative to ROC curves when there is a large skew in the class distribution. In PR space, recall (TPR) is plotted along the *x*-axis and Precision along the *y*-axis. Precision-recall curves are more suitable for our data since the amount of negative examples (90%) greatly exceeds the amount of positives examples (10%).

When a corpus is unbalanced, for example, when the number of negative examples greatly exceeds the number of positives examples, a small change in the number of false positives may be disguised by the large number of true negative in FPR (FP/(FP + TN)). While, in turn, precision (TP/(TP + FP)) is able to reflect the effect of the large number of negative examples on classifier performance because precision compares quantities in a closer order of magnitude (FP to TP rather than TN). Therefore, PR curves are more suitable than ROC curves for comparison in unbalanced datasets. In addition, Davis and Goadrich [53] showed that there is a strong connection between ROC space and PR space, such that a curve dominates in ROC space if and only if it dominates in PR space.

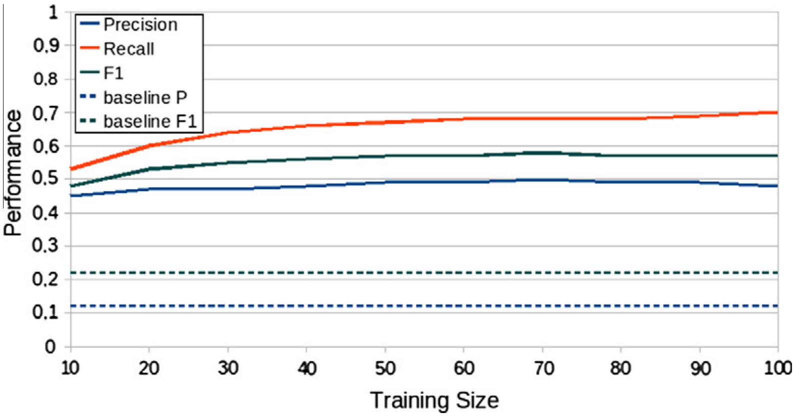


Fig. 11. Learning curves.

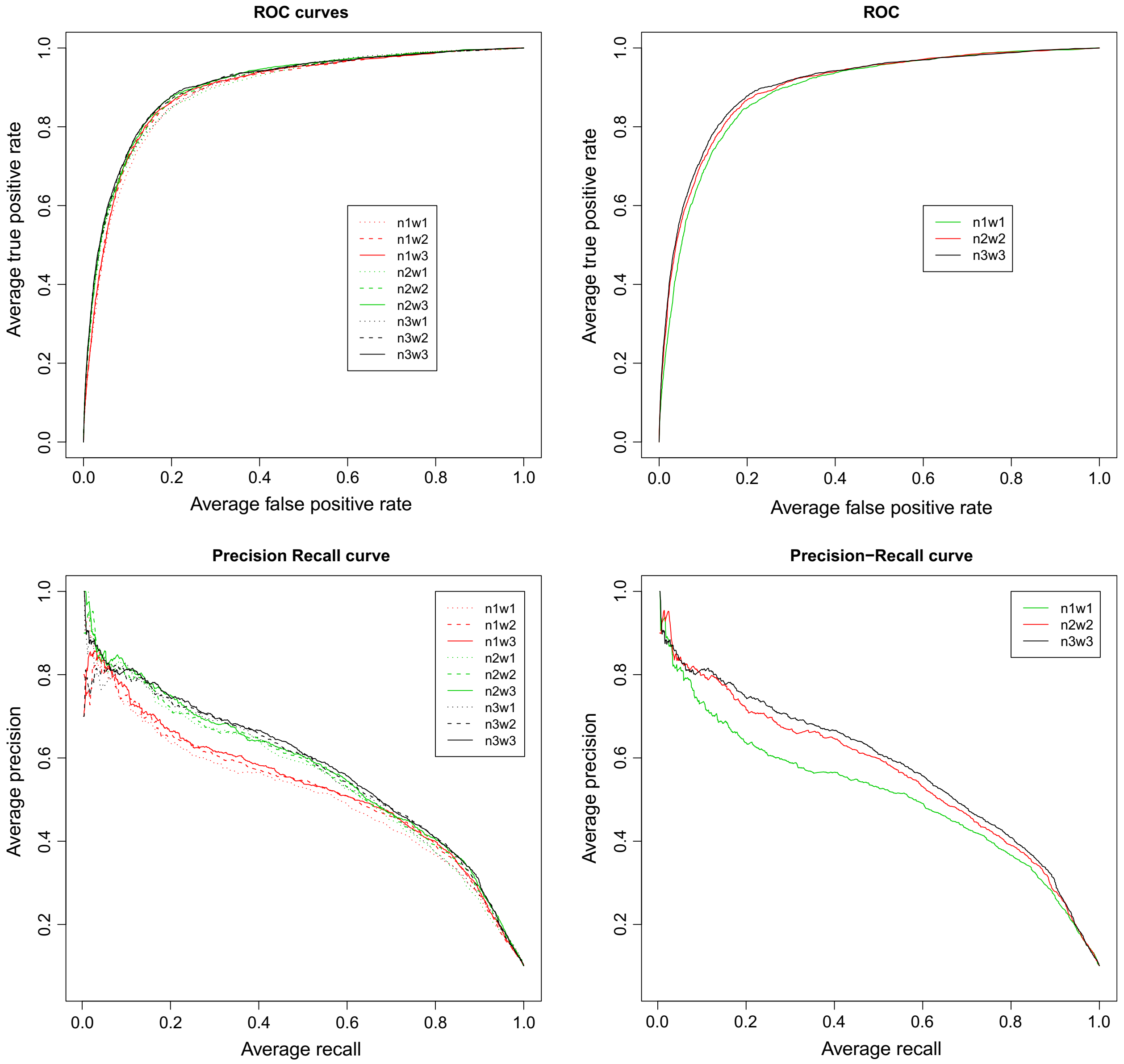


Fig. 12. ROC and precision-recall curves.

We generated both kinds of curves using the ROCR [54] package for visualizing classifier performance (Fig. 12). Whereas it is very difficult to distinguish between the different ROC curves produced, PR curves provide for more adequate distinctions between the different models. Moreover, PR curves were recommended by Jiang et al. [55] in cases when ROC curves are not capable of revealing differences in the performance of different classifiers. Scrutiny of the PR curves presented in Fig. 13 shows that an increase of both parameters led to improved performance, though the parameter *n*-gram exerted a greater influence on performance than window-size. Furthermore, improvement was greater when the *n*-gram parameter was increased from 1 to 2 than when it was increased from 2 to 3. Fig. 12 shows the ROC and PR curves predicted from all models. It can be observed that the model with *n*-gram = 3 and window-size = 3 dominates both ROC and PR spaces. Therefore, this model is at least as good as all other models for all possible error costs and class distributions. This finding is consistent with the results shown in Tables 4 and 6.

3.4.2. Statistical significance tests

McNemar's significance test [56] is a χ^2-based significance test used to compare two groups, such as two classifiers or two population samples. We applied the McNemar's significance test to compare the performance of the different configurations and determine whether or not they differ significantly. Thus, for each pair of possible configurations C_a and C_b, their corresponding models were performed on the final test document set. The classification of each example in the test set by each model was recorded, counting the number of examples correctly classified by C_a and C_b (n_{11}), the number of examples correctly classified by C_a but not by C_b (n_{10}), the number of examples misclassified by C_a but not by C_b (n_{01}), and the number of examples misclassified by both C_a and C_b (n_{00}). The contingency matrix shown in Table 7 was then built for any pair of configurations.

McNemar's test is based on a χ^2 goodness-of-fit test comparing the distribution of counts expected under the null hypothesis to the counts observed. The null hypothesis H_0 states that the two configurations should have the same error rate (i.e., $n_{10} = n_{01}$).

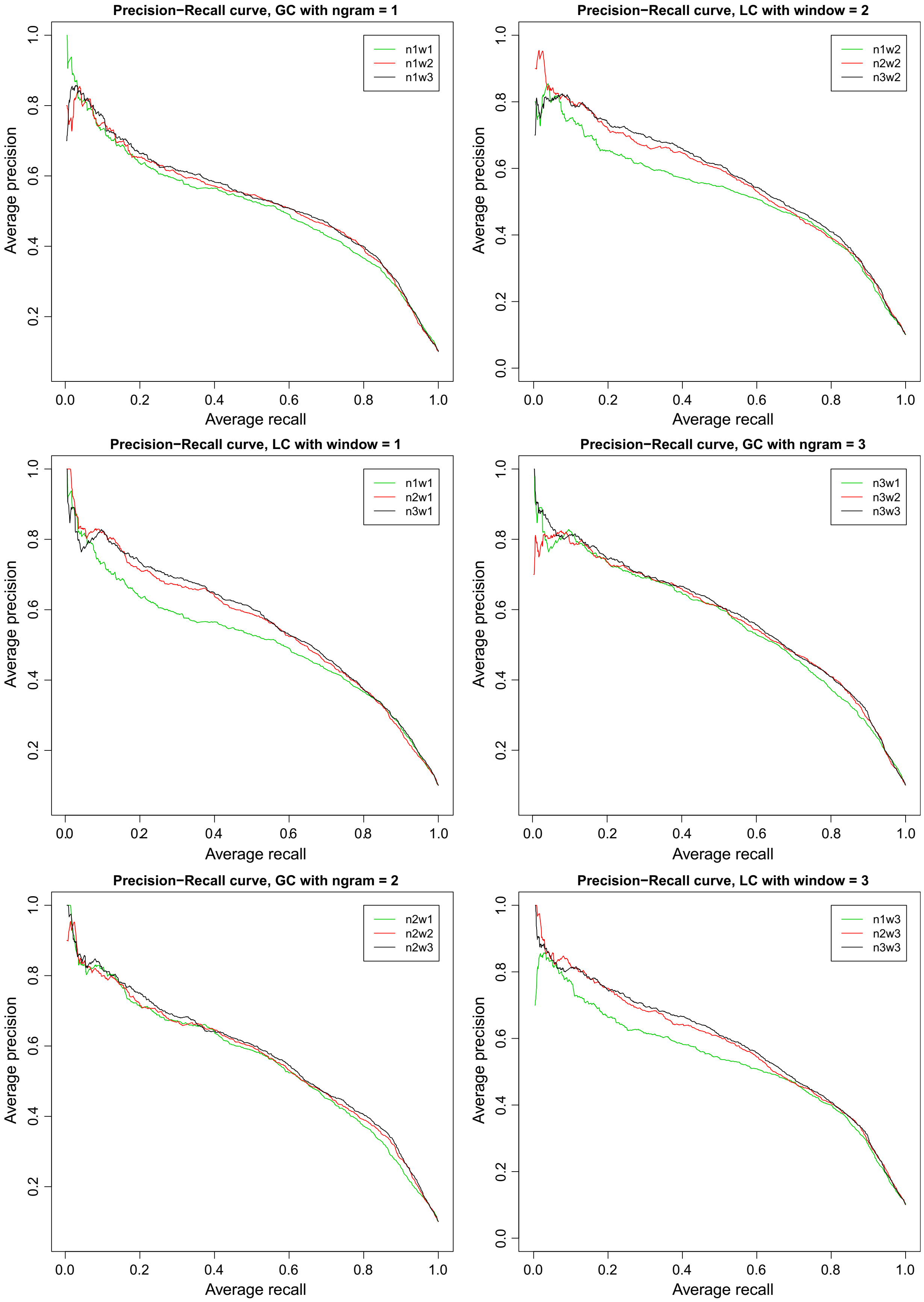


Fig. 13. Precision-recall curves.

Table 7
McNemar's test contingency table.

n_{11}	n_{10}
n_{01}	n_{00}

Table 8
χ statistic values using McNemar's test. The highest scores are marked with asterisks (∗).

	n1w2	n1w3	n2w1	n2w2	n2w3	n3w1	n3w2	n3w3
n1w1	2.53	12.80*	49.65*	38.13*	46. 20*	35.35*	44.10*	59.68*
n1w2	–	7.04*	30.95*	32.68*	38. 20*	22.08*	36.45*	49.33*
n1w3	–	–	12.66*	12.14*	22.23*	7.95*	16.40*	31.78*
n2w1	–	–	–	0.11	0.57	0.75	0.31	5.49*
n2w2	–	–	–	–	1.99	0.09	1.41	8.31*
n2w3	–	–	–	–	–	2.02	0.02	4.00*
n3w1	–	–	–	–	–	–	2.20	10.72*
n3w2	–	–	–	–	–	–	–	4.48*

According to Dietterich [57], under the null hypothesis the following statistic (see Eq. 8) is distributed as an χ^2 distribution with one degree of freedom.

$$\chi = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{8}$$

To test for significance, χ^2 was compared to the appropriate χ^2 table. Results with a probability greater than or equal to 0.05 are generally considered to be significant. Thus, the null hypothesis was correct if χ^2 was lower than $\chi^2_{1,0.05} = 3.841459$. In other cases, the null hypothesis could be rejected in favor of the other hypothesis that the two configurations produce different levels of performance.

Table 8 summarizes the χ statistic values for the pairwise comparison of the nine possible configurations using the McNemar's significance test (i.e., a total of 36 [9∗8/2] comparisons). Each cell of this pairwise comparison matrix represents the χ statistic value for a given pair of configurations.

Given the χ statistic values for the pairwise comparisons, differences in performance for pairs of configurations with *n*-gram parameters set at 2 are not significant. Similarly, any configuration with an *n*-gram of 2 does not significantly differ from those configurations with an *n*-gram equal of 3 and a window-size of less than 3 (see the gray cells in Table 8). Therefore, it may be concluded that those configurations with an *n*-gram parameter ranging in size from 2 to 3 and window-size parameter of less than 3 have the same rate error. As can be observed in Tables 4 and 6, these configuration pairs demonstrated very similar performance in the experiments run here.

On the other hand, the last column in Table 8 shows that values for the configuration with an *n*-gram and window-size of 3 significantly differs from all others. As these values support the findings from our experiments, we may assert that this configuration (*n*-gram = 3 and window-size = 3) achieves the highest performance.

3.4.3. Error analysis

Despite the findings above, it is important here to discuss certain limitations in our approach. Evaluated with the final test set (containing a total 739 DDIs), even the model with the highest *F*-measure (*n*-gram = 3, window-size = 3) was observed making a total of 597 errors, 276 of which being false negatives (i.e., interacting pairs that the system failed to detect) and 321 of which being false positives (i.e., pairs wrongly extracted by the system). To better understand these limitations of the system, we selected a random sample of 75% of these false positives and negatives for error analysis. Tables 9 and 10 present the principal causes for the false positives and false negatives generated, respectively.

Table 9
Principal causes of false positives generated.

Cause	%
Requiring resolution of coordinate structures and appositions	46
Requiring negation treatment	34
MMTx appositions	14
Corpus error	6

Table 10
Causes of error of false negatives generated.

Error in prediction model	41%
Detection of appositions and coordinate structures	24%
Long DDI descriptions	16%
Resolution of complex and compound sentences required	7%
Treatment of negation required	7%
Resolution of anaphora and cataphora required	5%

The most frequent cause of false positives was the system's incapability to distinguish between drugs constituting an apposition or a coordinate structure, and therefore, to recognize that they cannot interact. The following sentences are some examples of these false positives:

> *Bentiromid may interact with* ***acetaminophen*** *(e.g., Tylenol), chloramphenicol (e.g.,* ***Chloromycetin****), local anesthetics (e.g., benzocaine and lidocaine), para-aminobenzoic acid (PABA) – containing preparations (e.g., sunscreens and some multivitamins), procainamide (e.g., Pronestyl), sulfonamides (sulfa medicines), thiazide diuretics (use of these medicines during the test period will affect the test results).*

A possible approach to improve our system, therefore, could be to introduce a pre-processing step to detect appositions and coordinate structures.. Thus, the pairs of drugs contained within these structures could be removed from the set of possible relation instances. The second most important cause identified was the system's inability to properly deal with negation. For example, an interaction between azithromycin and warfarin was wrongly detected by the system in the following sentence: '**Azithromycin** did not affect the prothrombin time response to a single dose of **warfarin**'.

Another frequent cause of false positives were parsing errors made by MMTx. Furthermore, certain erros were caused by the incorrect classification of drug names by MMTx. Approximately 6% of the false positives analyzed were due to corpus annotation errors. In other words, the candidate pair actually represented a DDI and was identified as such by the system; however, the absence of an annotation as such in the corpus led to its classification as a false positive.

Regarding false negatives (see Table 10), the most frequent was the need for patterns that could not be extracted from the training corpus. For example, the system was not able to detect the DDI in the following sentence: '**Quinolon** has also been shown to interfere with the metabolism of **caffeine**'. Additional training data from different sources such as MedLine may improve these results.

In such a case, the resolution of coordinate and appositive structures through a pre-processing step could help improve performance: '**Quinolon**, including cinoxacin, may enhance the effects of oral anticoagulants, such as **warfarin** or its derivatives'.

Furthermore, some interactions were described with extremely long text at times including additional information about dosages or adverse reactions (see below). Global and local context kernels are not capable of dealing with these types of sentences. For example, the following sentence:

The incidence of akathisia in clinical trials of the weekly dosage schedule was greater(8.5%, 4/47 patients) when ***prochlorperazine*** *was administered on the same day as* ***CAMPTOSAR*** *than when these drugs were given on separate days (1.3%, 1/80 patients).*

Other interactions were not detected due to the inability of the system to deal with complex and compound sentences: 'Urinari alkalinizing agents increase blood levels and decrease excretion of amphetamines'. Another way in which system performance could be improved is through a greater attentiveness to negations. For example: 'Therefore, chloroprocaine should not be used in any condition in which a sulfonamide drug is being employed'.

Finally, it should be noted that the resolution of cataphora may also improve results. For example, in the following sentence, the term 'drugs' references to the following drugs:

Other ***drugs*** *which may enhance the neuromuscular blocking action of nondepolarizing agents such as* ***NUROMAX*** *include certain antibiotics (e.g., aminoglycosides, tetracyclines, bacitracin, polymyxins, lincomycin, clindamycin, colistin, and sodium colistimethate), magnesium salts, lithium, local anesthetics, procainamide, and quinidine'.*

3.4.4. Balancing experiments

The textual corpus used for this study was collected from a text field describing DDIs for a given drug in the pharmacological database, DrugBank. Had we instead chosen to build a corpus from Medline abstracts, we imagined that the number of sentences containing DDIs would have been much lower. As discussed earlier, of all pair of drugs occurring in our corpus (30,757), only 10% of them (3160) are drugs that interact. In other words, only a 10% of all relation instances are DDI (positive examples). For this reason, we wanted to study the impact of an imbalanced dataset on the performance of the kernel-based method. A common problem in most of the machine learning algorithms is their inability to accurately learn from imbalanced data. Minority classes are usually underrepresented and rules are fewer and weaker than those of the majority classes [58,59]. Solutions for imbalanced learning include sampling, as well as cost-sensitive and active learning methods. While a detailed description of these solutions can be found in [58], in the present study we focused on undersampling, a simple technique removing examples from the majority class in order to provide a balanced distribution of examples. Undersampling involves a considerable information loss, in which discriminative features to differentiate among classes may be discarded.

We therefore performed two experiments with different data distributions:

- *Imbalanced*: In this experiment, both the training and testing dataset are imbalanced as in our previous experiments. The experiment, therefore, is the closest to the real situation of previous experiments.
- *TrainingBalanced*: In this experiment, we used *undersampling* to randomly remove negative examples from the training dataset, while maintaining the test dataset imbalanced. The model trained on the balanced dataset was then applied to the imbalanced test dataset. Our hypothesis was that if the amount of positive and negative examples were the same in the training set, the model can distinguish the minority class (i.e. DDI) better. One drawback of this experiment is that the size of training dataset is reduced notably.

In each experiment, results were compared to a baseline *allDDI*, in which all examples were labeled as DDI (i.e positive examples). This baseline allowed us to estimate the improvement achieved in each experiment. The increment can be defined as follows:

$$Inc(F_{baseline}, F_{SL}) = \frac{F_{SL} - F_{baseline}}{F_{baseline}} \quad (9)$$

Table 11
Experimental results for imbalanced and balanced datasets.

Experiment	P	R	F_1	Inc.
allDDI	0.11	1	0.19	–
Imbalanced	0.5103	0.7282	0.6001	2.1584
TrainingBalanced	0.3469	0.8782	0.4973	1.6173

Table 12
Experimental results for imbalanced and balanced datasets grouped by class.

Experiment	Class	P	R	F_1
Imbalanced	0	0.97	0.92	0.94
	1	0.51	0.72	0.60
TrainingBalanced	0	0.98	0.80	0.88
	1	0.35	0.88	0.50

Table 11 shows the results obtained in each experiment. In the first experiment (i.e. *Imbalanced*), the baseline only achieves a precision of 11%, a perfectly predictable result given that the percentage of positive examples was 11%. The learned model, on the other hand, achieved good performance with an *F*-measure of 60.01%, an improvement of 41% and an increment of 2.1584 with respect to the baseline. In the second experiment (i.e. *TrainingBalanced*), a high recall (87.82%) was obtained; however, precision was also quite low (34.69%). Thus, while the balancing of training data helps to improve the recall, precision values are nevertheless adversely affected. The *F*-measure increment with respect to the baseline was lower in the *TrainingBalanced* experiment (1.6173) than in the *Imbalanced* experiment (2.1584). As a result, it can be concluded that balancing positive and negative examples by undersampling mechanism does not lead to results (i.e. a superior *F*-measure) better than those obtained from the imbalanced data.

Regarding the classification task, Table 12 presents experiment performance separated by class (i.e where DDI = 1 and non-interaction = 0). As evidenced by the table, the experiments demonstrate strong performance with negative examples. In the *Imbalanced* experiment, these results are due to the fact that theres is a significantly greater amount of negative examples than the number of positive examples, providing strong clues for the description of the majority class (i.e. non-interactions). It must be noted, however, that in the *TrainingBalanced* experiment in which the number of negatives examples was reduced to equal the number of positive examples, the results for obtained for the negative examples (i.e. the non-interaction class) were nevertheless considerably high. Thus, we are led to believe that the determination of a non-interaction is easier the determination of a DDI.

4. Conclusion and discussion

In the present study, our major objective was to evaluate the performance of the shallow linguistic kernel-method introduced in Giuliano et al. [2] in the extraction of DDI from biomedical texts. Several experiments have been conducted on the DrugDDI corpus. In our experiments, we varied *n*-gram (global context kernel) and window-size (local context kernel) configuration parameters. Greatest precision (52.07%) was achieved when both *n*-gram size and window-size were equal to 3. The highest recall value (78.63%) was produced with an *n*-gram size of 1 and window-size of 2. Nevertheless, it is important to note that this configuration also led to the second lowest recorded precision value (43.97%). Among all trained models, we choose that which maximized the

F-measure and precision values (i.e. *n*-gram = 3, window-size = 3). With the final testing dataset, the model achieved a precision of 51.03%, a recall of 72.82% and an *F*-measure of 60.01%. In the experiments, a small *n*-gram size appeared to favor the obtainment of larger recall values while a larger *n*-gram size favored greater precision values. While the local context kernel was originally designed to identify the roles of the candidate entities within a relation [2] (and no distinguishing between roles of interacting drugs was done here), our results nevertheless show that the local kernel also assists with DDI extraction since the combination of both global and local kernels improved the precision of the shallow linguistic kernel. The DrugDDI corpus presented a large, imbalanced distribution between positive and negative examples,we followed up our principal experiments with others to study the influence of this imbalance on study results. In these latter experiments, we found that efforts to balance the positive and negative examples did not lead to higher performance.

From previous studies, the shallow kernel had already shown strong performance in both general and biological domains [2,6]. In particular, [2] performed several experiments on two different biomedical corpora for protein–protein interactions, AImed and LLL. Their experiments were performed using the correct named entities, that is, entities manually annotated in the corpora. Results obtained on the AImed corpus showed a precision of 60.0%, a recall of 57.2%, and *F*-measure of 59%. Superior performance was nevertheless achieved on the LLL corpus, with a precision of 62.1%, a recall of 61.3%, and an *F*-measure of 61.7%. Although direct comparisons between these experiments and our own are not possible due to the fact that a different type of relation (i.e. DDIs) was studied here and for which a new corpus was built, the same shallow linguistic kernel applied to the new task of DDI extraction appears to have achieved a similar *F*-measure (60.01%) and a higher recall (72.82%). Nevertheless a lower precision (51.03%) also resulted. One possible explanation for this lower precision values could be that our performance demonstrated the remarkable impact of automatic entity recognition on the relation extraction task. Had drug names been manually labeled in our corpus, it is highly likely that our results would have been significantly improved. Furthermore, while the LLL corpus is smaller than the DrugDDI corpus, the average number of interactions per sentence is higher in the former corpus (i.e., 2.0 in the LLL corpus) than in the latter (i.e., 0.6 in the DrugDDI corpus). We believe that a higher density of interactions would positively affect performance since sentences in the LLL corpus are focused on interaction description, whereas DrugDDI corpus sentences may be less discriminating.

Our pattern-based approach from a previous study [1] was evaluated on the DrugDDI corpus, achieving a precision of 48.89%, a recall of 24.81% and an *F*-measure of 32.92%. In order to compare the pattern-based approach and shallow linguistic kernel, the latter was tested on the whole DrugDDI corpus using 10-fold cross-validation. It is clear from the study results that the kernel-based method is far superior to our earlier pattern-based approach. The most significant improvement observed in the kernel-based approach was achieved for recall and *F*-measure values, increasing to 71.19% and 59.52%, respectively. Thus, relative to values obtained from the earlier, pattern-based approach, recall increased by nearly 47% and the *F*-measure by nearly 27%. A minor improvement was also achieved for precision which increased by 2.36%. As a result, we can conclude that the machine learning-approach is far more efficient than the pattern-based approach for tackling DDI extraction from texts.

To conclude, we believe that the solid performance achieved using the shallow linguistic kernel may provide a higher baseline, permitting the measurement of improvements with other methods that use full syntactic or semantic information. We propose several specific ideas for future work:

- Evaluate the performance of the kernel-method when drug names are manually annotated.
- Label the roles of drugs in the DrugDDI corpus in order to evaluate the contribution of the local kernel in their detection.
- Define a semantic kernel using semantic information such as UMLS semantic type or drug families obtained by our DrugNer system [37].
- Design parse tree or dependency graph kernels for DDI extraction.
- Evaluate other solutions for imbalanced learning such as hybrid sampling or cost-sensitive methods.

Finally, in addition to a list of potential DDIs, an ideal description for a particular drug should also include more specific information about each interaction including the interaction mechanism, its relation to the doses of both drugs, its time course, the factors altering an individual's susceptibility to the DDI, its seriousness and severity, as well as the probability of its occurrence [60,61]. In practice, however, this information is rarely available in DDI knowledge bases [62]. Nevertheless, it may be included using similar techniques. As the detection of this additional information could help healthcare professionals assign real clinical significance to each DDI, it represents an additional, important issue for future study.

Acknowledgments

This study was funded by the Projects MA2VICMR (S2009/TIC-1542) and MULTIMEDICA (TIN2010-20644-C03-01). The authors are particularly grateful to María Segura-Bedmar, manager of the Drug Information Center of the Mostoles University Hospital (Spain) for her invaluable support in the creation and evaluation of the DDI corpus.

References

[1] Segura-Bedmar I, Martínez P, De Pablo-Sánchez C. Combining syntactic information and domain-specific lexical patterns to extract drug–drug interactions from biomedical texts. In: Proceedings of the ACM fourth international workshop on data and text mining in biomedical informatics (DTMBIO'10); 2010. p. 49–56.

[2] Giuliano C, Lavelli A, Romano L. Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Proceedings of the eleventh conference of the European chapter of the association for computational linguistics (EACL-2006); 2006. p. 5–7.

[3] Rodríguez-Terol A, Camacho C. Others, calidad estructural de las bases de datos de interacciones. Farmacia Hospitalaria 2009;33(03):134.

[4] Duda S, Aliferis C, Miller R, Statnikov A, Johnson K. Extracting drug–drug interaction articles from MEDLINE to improve the content of drug databases. In: AMIA annual symposium proceedings, vol. 2005; 2005. p. 216.

[5] Hansten PD. Drug interaction management. Pharm World Sci 2003;25(3): 94–7.

[6] Giuliano C, Lavelli A, Romano L. Relation extraction and the influence of automatic named-entity recognition. ACM Trans Speech Lang Process (TSLP) 2007;5(1):2. doi:10.1145/1322391.1322393.

[7] Zhou D, He Y. Extracting interactions between proteins from the literature. J Biomed Inform 2008;41(2):393–407.

[8] Katrenko S, Adriaans P. Learning relations from biomedical corpora using dependency trees. Lect Notes Comput Sci 2007;4366:61–80.

[9] Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, et al. Comparative experiments on learning information extractors for proteins and their interactions. Artif Intell Med 2005;33(2):139–55.

[10] Yang Z, Lin H, Li Y. BioPPISVMExtractor: a protein–protein interaction extractor for biomedical literature using SVM and rich feature sets. J Biomed Inform 2010;43(1):88–96.

[11] Grinberg D, Lafferty J, Sleator D. A robust parsing algorithm for link grammars. Arxiv preprint cmp-lg/9508003.

[12] Ding J, Berleant D, Nettleton D, Wurtele E. Mining MEDLINE: abstracts, sentences, or phrases? In: Pacific symposium on biocomputing, January 3–7, 2002, Kauai, Hawaii; 2002. p. 326.

[13] Corney DPA, Buxton BF, Langdon WB, Jones DT. BioRAT: extracting biological information from full-length papers. Bioinformatics 2004;20(17):3206–13.

[14] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge: Cambridge University Press; 2004.

[15] Zelenko D, Aone C, Richardella A. Kernel methods for relation extraction. J Mach Learn Res 2003;3:1083–106.

[16] Culotta A, Sorensen JS. Dependency tree kernels for relation extraction. In: Proceedings of ACL, vol. 4; 2004.

[17] Bunescu R, Mooney RJ. A shortest path dependency kernel for relation extraction. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, October, Association for Computational Linguistics Morristown, NJ, USA; 2005. p. 724–31.

[18] Bunescu R, Mooney R. Subsequence kernels for relation extraction. Adv Neural Inform Process Syst 2006;18:171.

[19] Bunescu R, Mooney RJ. Extracting relations from text: from word sequences to dependency paths. In: Natural language processing and text mining; 2007. p. 29–44.

[20] Li J, Zhang Z, Li X, Chen H. Kernel-based learning for biomedical relation extraction. J Am Soc Inform Sci Technol 2008;59(5):756–69.

[21] Airola A, Pyysalo S, Bjorne J, Pahikkala T, Ginter F, Salakoski T. A graph kernel for protein-protein interaction extraction. In: Proceedings of BioNLP; 2008. p. 1–9.

[22] Pyysalo S, Ginter F, Heimonen J, Björne J, Boberg J, Järvinen J, et al. BioInfer: a corpus for information extraction in the biomedical domain. BMC Bioinform 2007;8(1):50.

[23] Fundel K, Kuffner R, Zimmer R. RelEx-relation extraction using dependency parse trees. Bioinformatics 2007;23(3):365.

[24] Krallinger M, Leitner F, Rodriguez-Penagos C, Valencia A. Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biol 2008;9(Suppl. 2):S4.

[25] Pyysalo S, Airola A, Heimonen J, Bjorne J, Ginter F, Salakoski T. Comparative analysis of five protein-protein interaction corpora. BMC Bioinform 2008;9(Suppl. 3):S6.

[26] Tikk D, Thomas P, Palaga P, Hakenberg J, Leser U. A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature. PLoS Comput Biol 2010;6(7):e1000837.

[27] Ando R. BioCreative II gene mention tagging system at IBM Watson. In: Proceedings of the second biocreative challenge evaluation workshop, Citeseer; 2007. p. 101–3.

[28] Kuo C, Chang Y, Huang H, Lin K, Yang B, Lin Y, Hsu C, Chung I. Rich feature set, unification of bidirectional parsing and dictionary filtering for high *F*-score gene mention tagging. In: Proceedings of the second biocreative challenge evaluation workshop (BioCreative II), Madrid, Spain, Citeseer; 2007.

[29] Huang H, Lin Y, Lin K, Kuo C, Chang Y, Yang B, et al. High-recall gene mention recognition by unification of multiple backward parsing models. In: Proceedings of the second biocreative challenge evaluation workshop, Citeseer; 2007. p. 109–11.

[30] Klinger R, Friedrich C, Fluck J, Hofmann-Apitius M. Named entity recognition with combinations of conditional random fields. In: Proceedings of the second biocreative challenge evaluation workshop, Citeseer; 2007. p. 105–7.

[31] Yang Z, Lin H, Li Y. Exploiting the performance of dictionary-based bio-entity name recognition in biomedical literature. Comput Biol Chem. 2008;32(4):287–91.

[32] Leaman R, Gonzalez G. Banner: an executable survey of advances in biomedical named entity recognition. In: Pacific symposium on biocomputing, vol. 13; 2008. p. 652–63.

[33] Pafilis E, O'Donoghue S, Jensen L, Horn H, Kuhn M, Brown N, et al. Reflect: augmented browsing for the life scientist. Nat Biotechnol 2009;27(6):508–10.

[34] Sirohi E, Peissig P. Study of effect of drug lexicons on medication extraction from electronic medical records. In: Biocomputing 2005: proceedings of the pacific symposium, Hawaii, USA, 4–8 January 2005; 2005.

[35] Hettne K, Stierum R, Schuemie M, Hendriksen P, Schijvenaars B, van Mulligen E, et al. A dictionary to identify small molecules and drugs in free text. Bioinformatics 2009;25(22):2983–91.

[36] Gurulingappa H, Kolarik C, Hofmann-Apitius M, Fluck J. Concept-based semi-automatic classification of drugs. J Chem Inform Model 2009;49(8):1986–92.

[37] Segura-Bedmar I, Martínez P, Segura-Bedmar M. Drug name recognition and classification in biomedical texts: a case study outlining approaches underpinning automated systems. Drug Discov Today 2008;13(17–18):816–23.

[38] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Annual AMIA symposium; 2001. p. 17–21.

[39] Kolarik C, Hofmann-Apitius M, Zimmermann M, Fluck J. Identification of new drug classification terms in textual resources. Bioinformatics 2007;23(13):i264.

[40] Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, et al. DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res 2008;36(Database issue):D901–6. doi:10.1093/nar/gkm958.

[41] Garcia-Blasco S, Danger R, Rosso P. Drug–drug interaction detection: a new approach based on maximal frequent sequences. in: SEPLN, vol. 45; 2010. p. 263–6.

[42] Rubin DL, Thorn CF, Klein TE, Altman RB. A statistical approach to scanning the biomedical literature for pharmacogenetics knowledge. J Am Med Inform Assoc 2005;12(2):121–9.

[43] Danger R, Segura-Bedmar I, Martínez P, Rosso P. A comparison of machine learning techniques for detection of drug target articles. J Biomed Inform 2010;46(6):902–13.

[44] Nedellec C. Learning language in logic–genic interaction extraction challenge. In: Proceedings of the ICML05 workshop: learning language in logic (LLL05), vol. 18; 2005. p. 97–9.

[45] McCray A, Srinivasan S, Browne A. Lexical methods for managing variation in biomedical terminologies. Annual symposium on computer application in medical care, vol. 18. IEEE Computer Society Press; 1994. p. 235–9.

[46] Cutting D, Kupiec J, Pedersen J, Sibun P. A practical part-of-speech tagger. In: Proceedings of the third conference on applied natural language processing; 1992.

[47] Aronson A. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the AMIA symposium, American Medical Informatics Association; 2001. p. 17.

[48] Yu H. Towards answering biological questions with experimental evidence: automatically identifying text that summarize image content in full-text articles. In: Annual AMIA symposium proceedings; 2006. p. 834–8.

[49] Porter MF. An algorithm for suffix stripping. Program 1980;14(3):130–7.

[50] Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001.

[51] Lavelli A, Califf ME, Ciravegna F, Freitag D, Giuliano C, Kushmerick N, et al. Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. Lang Resour Eval 2008;42:361–93. doi:10.1007/s10579-008-9079-3.

[52] Provost F, Fawcett T, Kohavi R. The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the fifteenth international conference on machine learning, vol. 445, Citeseer; 1998.

[53] Davis J, Goadrich M. The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on machine learning. ACM; 2006. p. 233–40.

[54] Sing T, Sander O, Beerenwinkel N, Lengauer T. ROCR: visualizing classifier performance in R. Bioinformatics 2005;21(20):3940.

[55] Jiang Y, Cukic B, Ma Y. Techniques for evaluating fault prediction models. Empirical Software Eng 2008;13(5):561–95.

[56] McNemar Q. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 1947;12(2):153–7.

[57] Dietterich T. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 1998;10(7):1895–923.

[58] He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowledge Data Eng 2009;21(9):1263.

[59] Van Hulse J, Khoshgoftaar T. Knowledge discovery from imbalanced and noisy data. Data Knowledge Eng 2009;68(12):1513–42.

[60] Ferner RE, Aronson JK. Communicating drug safety. JBM 2006;333:1435.

[61] Aronson JK. Drug interactions-information, education, and the British National Formulary. Br J Clin Pharmacol 2004;57(4):473–86.

[62] Aronson JK. Communicating information about drug interactions. Br J Clin Pharmacol 2007;63(6):637–9. doi:10.1111/j.1365-2125.2007.02948.x.