ACM fourth international workshop on Data and text mining in biomedical informatics

(2010) pp. 49-56

DOI: 10.1145/1871871.1871881

Combining Syntactic Information and Domain-specific Lexical Patterns to Extract Drug-Drug Interactions from Biomedical Texts

Isabel Segura-Bedmar
Computer Science Department
Universidad Carlos III de Madrid
Avd. Universidad, 30
Leganés, Madrid, Spain
isegura@inf.uc3m.es

Paloma Martínez
Computer Science Department
Universidad Carlos III de Madrid
Avd. Universidad, 30
Leganés, Madrid, Spain
pmf@inf.uc3m.es

Cesar de Pablo-Sánchez
Computer Science Department
Universidad Carlos III de Madrid
Avd. Universidad, 30
Leganés, Madrid, Spain
cdepablo@inf.uc3m.es

ABSTRACT

A drug-drug interaction (DDI) occurs when one drug influences the level or activity of another drug. The increasing volume of the scientific literature overwhelms health care professionals trying to be kept up-to-date with all published studies on DDI. Information Extraction (IE) techniques can provide an interesting way of reducing the time spent by health care professionals on reviewing the literature. Nevertheless, no approach has been carried out to extract DDI from texts. To the best of our knowledge, this work proposes the first integral solution for the automatic extraction of DDI from biomedical texts.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms

Languages, Experimentation

Keywords

Information Extraction, Drug-Drug Interaction Extraction

1. INTRODUCTION

A DDI occurs when one drug influences the level or activity of another, for example, raising its blood levels and possibly intensifying its side effects or decreasing drug concentrations and thereby reducing its effectiveness. The detection of DDI is an important research area in patient safety since these interactions can become very dangerous and increase health care costs. Although there are different databases supporting health care professionals in the detection of DDI, these databases are rarely complete, since their update periods can reach three years [19]. Drug interactions are frequently reported in journals of clinical pharmacology and technical reports, making medical literature the most effective source for the detection of DDI. Thus, the management of DDI is a critical issue due to the overwhelming amount of information available on them [13].

Information Extraction (IE) can be of great benefit in the pharmaceutical industry allowing identification and extraction of relevant information on DDI and providing an interesting way of reducing the time spent by health care professionals on reviewing the literature. Moreover, the development of tools for automatically extracting DDI is essential for improving and updating the drug knowledge databases. Nevertheless, no approach has been carried out to extract DDI from biomedical texts.

Although many approaches have been proposed to extract biomedical relations, only a few of them achieve successful results. One important reason is that only a few approaches have dealt with the issue of the complexity of biomedical sentences [14]. However, language structures such as apposition, coordination and complex sentences are very common in the biomedical literature. We think that the detection of these linguistic phenomena is essential to successfully tackle the extraction of biomedical relations, in particular, DDI.

In this work, we propose a hybrid method that combines shallow parsing and pattern matching to extract relations between drugs from biomedical texts. A pharmacist defined a set of domain-specific lexical patterns to capture the most common expressions of DDI in texts, based on her professional experience and the corpus observation. Our method is based on the approach described in [14], which proposes a set of syntactic patterns to split the long sentences into clauses from which relations are extracted by a pattern matching algorithm. This approach works on the detection of appositions, coordinate constructions and relative clauses. Our contribution extends this approach dealing with any kind of subordinate and coordinate clause. Appositions and coordinate structures are interpreted based on shallow syntactic parsing provided by the UMLS MetaMap tool (MMTx) [3]. Subsequently, complex and compound sentences are broken down into clauses from which simple sentences are generated

Table 1: Main approaches for PPI extraction

System	Approach	Corpus	F_1
IntEx [1]	Link grammar + patterns	DIP	38.9%
AkanePPI [20]	dependency parsing + pattern matching	BioCreative-PPI	19%
Verspoora et al.[25]	semantic grammar + pattern matching	BioCreative-PPI	25.2%
BioPPISVMExtractor [29]	link grammar parser + SVM[1]	DIP	57.85%
Chen et al. [7]	SVM	BioCreative-PPI	57.8%
Airola et al., [2]	dependency-path kernel	Aimed, BioInfer, HPRD50, IEAP, LLL	56.4% (AIMed)

by a set of simplification rules. Finally, the lexical patterns are matched with the generated sentences in order to extract DDI.

The paper is organized as follows. Section 2 reviews the main approaches addressed for biomedical relation extraction. Section 3 describes the dataset used to develop and evalute our method. The treatment of coordinate structures and appositions is described in sections 4 and 5 respectively. Section 6 shows how clauses boundaries are identified using shallow syntactic information and how simple sentences are generated from the clauses. Section 7 introduces the set of domain-specific lexical patterns proposed by our pharmacist. Section 8 describes in detail the experiments and presents the experimental results. Finally, conclusions and future work are presented in Section 9.

2. RELATED WORK

Most investigation has centered around biological relationships (genetic and protein interactions (PPI)) due mainly to the availability of annotated corpora in the biological domain, a fact that facilitates the evaluation of approaches. In general, current approaches can be divided into three main categories: linguistic-based, pattern-based and machine learning-based approaches.

The general idea of *linguistic-based approaches* is to employ linguistic technology to grasp syntactic structures or semantic meanings that could be helpful to discover relations from unstructured texts. *Pattern-based approaches* design a set of domain-specific rules (also called patterns) that encode and capture the various forms of expressing a given relationship. As opposed to the previous approaches, which need a laborious effort to define grammars or a set of rules, the *machine learning* methods allow to automatically acquire and code all the necessary knowledge. Table 1 shows some of the main works for biomedical relation extraction.

The comparison among the different works is not always possible because many of them have been evaluated on different corpora. Therefore, it is risky to draw conclusions on the performance of the different techniques. 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 [30]. We believe that the performance of linguistic-based approaches is strongly influenced by the shortage of biomedical parsers. General purpose parsers, which have been trained on generic newswire texts, are not able to deal with the complexity of the biomedical sentences that tend to cause problems due to their long length and high degree of ambiguity [24].

Pattern-based approaches usually achieve high precision, but low recall. They are not capable of handling long and complex sentences, so common in biomedical texts. Furthermore, these approaches are limited by the extent of the patterns, since relations spanning several sentences cannot be detected by them. Linguistic phenomena including negation, modality and mood, which can alter or even reverse the meaning of the sentence, have hardly ever been studied by the pattern-based approaches. Thus, pattern-based approaches are not able to correctly process anything other than short and straightforward sentences [30], which, on the other hand, are quite rare in biomedical texts.

In general, machine learning-based approaches have achieved better performance than linguistic-based and pattern-based ones, as demonstrated in the last BioCreative challenge [17]. One important advantage of these approaches is that they can be easily extended to new set of data or a new task or domain. However, machine learning-based approaches depend heavily on the annotated corpora for training and testing. Corpus annotation is an expensive work, usually involving an extensive time and labor.

As can be observed in Table 1, most works adopt hybrid approaches. In particular, linguistic techniques such as tokenization, PoS tagging and syntactic parsing, are widely used by both pattern-based and machine learning-based approaches. This paper describes a hybrid approach to DDI extraction that combines shallow parsing and pattern matching.

3. THE DRUGDDI CORPUS

Most biomedical corpora (BioInfer [18], BioCreAtIvE-PPI [16] or AIMed [4]) have focus on describing genetic or protein interactions, but none contains DDI. While NLP techniques are relatively domain-portable, corpora are not. For this reason, we have created the first annotated corpus that studies the phenomena of interations among drugs.

The DrugDDI corpus consists of 579 documents describing DDI. These documents were randomly selected from the DrugBank database [28] and analyzed by the UMLS MetaMap Transfer (MMTx) tool [3] that performs sentence splitting, tokenization, POS-tagging, shallow syntactic parsing, and linking of phrases with Unified Medical Language System (UMLS) Metathesaurus concepts. Thus, MMTx allows to recognize a variety of biomedical entities, including drugs. The DrugDDI corpus consists of 66,021 phrases from which 22.6% (14,930) are drugs. The corpus contains 3,775 sentences with two or more drugs, although only 2,044 sentences have at least one interaction. A total of 3,160 DDI were annotated at sentence level with the assistance of a pharmacist. The average number of interactions per document is 5.46 and per sentence 0.54.

Table 2: Patterns to detect coordinate, correlative and appositive structures.

COORD	$([NP \mid PP \mid ADJ \mid UNK],)^* \ [NP \mid PP \mid ADJ \mid UNK] \ CONJ \ [NP \mid PP \mid ADJ \mid UNK]$
	$(VP,)^* \ VP \ CONJ \ VP$
CORRELATIVE	$[BOTH \mid EITHER \mid NEITHER][NP \mid PP \mid UNK] \ [AND \mid OR \mid NOR] \ [NP \mid PP \mid UNK]$
APPOSITIVE	$[NP \mid PP \mid UNK \mid APPOSITION]$
APPOSITION	$APPOSITIVE(,)? \ (()? \ MARKER \ [APPOSITIVE(,)?]+ \ (AND \mid OR)? \ (APPOSITIVE)? \ ())?$

4. DETECTING COORDINATE STRUCTURES

Coordination is an extremely common grammatical phenomenon in biomedical texts. Since coordinate constituents are semantically close and usually they play the same syntactic and grammatical roles in a sentence, it is necessary to assemble them together [14]. For example, the following sentence contains three DDI:

- Aspirin may decrease the effects [of probenecid]$_{PP}$, [sulfinpyrazone]$_{NP}$, and [phenylbutazone]$_{NP}$

In order to extract them, it is necessary to interpret the coordinate structure in it: *probenecid, sulfinpyrazone, and phenylbutazone*, in which the conjunction *and* coordinates the conjunct *probenecid* with *sulfinpyrazone* and with *phenylbutazone*.

Although a wide variety of structures can be conjoined, not all coordinations are acceptable. *Coordination of Likes Constraint (CLC)* [26] (also called *Law of Coordination of Likes*) asserts that syntactically different categories cannot be conjoined. However, based on the corpus observation, this constraint is too restrictive for the kind of parsing provided by MMTx. For example, the above sentence demonstrates that being of the same syntactic category is too strong requirement for conjuncts in a coordinate construction, since a prepositional phrase, *of probenecid*, can be conjoined with two noun phrases: *sulfinpyrazone* and *phenylbutazone*. In fact, we have observed in the corpus that coordinate structures involving constituents with different syntactic categories are very common. Sometimes it is due to the fact that MMTx is not able to determine the syntactic type of a phrase, classifying it as an unknown phrase (that is, with the tag *UNK*).

Table 2 presents a set of syntactic patterns to detect coordinate structures, where the first row shows a pattern in which different syntactic types can be combined to detect coordination at the phrase level. An exception is made for verb phrases, since the coordination between a verbal phrase and another type of syntactic phrase is a coordination between clauses (which is tackled in Section 6). Thus, the second pattern only allows to connect the verbal phrases with verbal phrases. Since this section focuses on coordination between phrases, we have only considered the coordinators *and, or, nor, and/or, as well as* as possible coordinators to link phrases. Table 2 also includes a syntactic pattern to detect correlative expressions such as *both midazolam and triazolam* (third row).

5. IDENTIFYING APPOSITIONS

There are divergent views within Linguistics with regard to what is or is not an apposition (also called appositional or appositive structure). [12] and [11] restrict the category of apposition to coreferential noun phrases (called appositives) that are juxtaposed and refer to the same extralinguistic entity. [8] and [15] expand this definition with the inclusion of constructions such as clauses and sentences as possible elements of an apposition. [5] admits as apposition only those constructions which can be linked by a marker of apposition.

Although the above approaches provide insights into the category of apposition, they provide either an inadequate or an incomplete description of apposition. The objective of this work is not to provide formal and complete description of apposition, but rather to identify appositions, in particular, those that contain drugs. Thus, we only deal with appositions that are linked by a marker of apposition since this kind of apposition appears frequently in the sentences that contain DDIs. Markers are helpful clues for detecting these structures. The markers of apposition that we have used in this approach are: *such as, like, including, for example, e.g. and i.e.*. Appositions that are not linked by any marker are also frequent in scientific texts, however, the lack of markers makes the detection of this kind of apposition extremely difficult. Moreover, we have observed they hardly ever occur in expressions describing DDI.

We have defined a set of syntactic patterns in order to identify the appositions (see table 2). Appositions comprise at least two contiguous phrases, the second of which is marked by clues such as parentheses or markers. This second phrase may be a coordinate structure. The *APPOSITIVE* pattern allows to recognize the intervening elements in an apposition, that is, their appositives. This pattern matches a phrase type (provided by MMTx) or another apposition. In this way, the pattern is able to recognize nested appositions. Regarding the phrase types, it has not considered types such as *VP*, *CONJ*, *ADV*, or, *ADJ*, since our main focus is to recognize appositions containing drugs (drugs only appear in noun, preposition and unknown phrases). The *APPOSITION* pattern is used to recognize appositions. This pattern matches an intervening element *APPOSITIVE* followed by a marker and by one or more intervening elements expressed by coordinate phrases. Parentheses are also included in the pattern.

Two different DDI can be extracted from the sentence:

- Catecholamine-depleting drugs]$_{NP}$, such as [Reserpine]$_{NP}$, may have an additive effect when given [with beta-blocking agents]$_{PP}$

(1) *Catecholamine-depleting drugs* with *beta-blocking agents*, and (2) *Reserpine* with *beta-blocking agents*. Thus, it is essential to detect and resolve the appositions occurring in sentences, prior to the application of the lexical patterns responsible for DDI extraction. The appositions are firstly encapsulated and then unfolded when the relation is obtained by any lexical pattern. Section 8 describes in detail the stage of matching.

6. CLAUSE SPLITTING

Biomedical texts usually consist of extremely long sentences. Long sentences are usually complex or compound-complex sentences, that is, contain two or more clauses. For example, the following sentence:

- Coadministration of CRIXIVAN and [other durgs that inhibit CYP3A4]$_{rel}$ [may decrease the clearance of indinavir]$_{clause1}$ and [may result in increased plasma concentrations of indinavir]$_{clause2}$.

contains two independent clauses (marked with *clause1* and *clause2*). Both clauses have the same subject: *Coadministration of CRIXIVAN and other drugs that inhibit CYP3A4*. This subject includes a relative clause (marked with *rel*) whose subject is *other drugs*. Parsing-based and pattern-based approaches are inefficient to deal with complex and compound sentences. Parsers are usually trained in common English text corpora and are difficult to extend to new domains. For this reason, they usually fail particularly in biomedical complex sentences. Regarding the pattern-based methods, relations are possibly extracted incorrectly when patterns are matched beyond the scope of one clause or other kinds of grammatical units [14]. For example, the previous example contains a relative clause (*that inhibit CYP3A4*), which hinders the matching between the sentence and the P_8 pattern (see Table 6).

This section proposes an algorithm for clause splitting that aims to reduce the complexity of sentences in biomedical texts, in order to improve the performance of our pattern-based method for DDI extraction. Clause splitting is the task of dividing a complex or compound sentence into several clauses. The algorithm exploits syntactic and lexical information provided by MMTx. Once sentences have been split into clauses, a set of simplification rules is used in order to generate new independent sentences from the clauses. Finally, the lexical patterns defined by the pharmacist can be applied to the generated sentences in order to extract DDI.

We now explain how the sentences are broken into clauses. First of all, it is necessary to ensure that the sentence is actually a compound or a complex sentence. It is not enough to check that there is some coordinator or subordinator in the sentence since sometimes they do not function like connectors between clauses, but as prepositions, adverbs, etc. A possible heuristic is to count the number of verb phrases included in the sentence. To give a definition of verb phrase is not an easy task. In fact, linguists have not even reached an agreement on what the verb phrase should include: only the words that are verbs, or also the complements of the verb. While the generative grammarians propose that a verb phrase consists of various combinations of the main verb and any auxiliary verbs, plus optional specifiers, complements, and adjuncts (for example, *Anagrelide* [*may interacts with any of these compounds*]$_{VP}$), for functionalist linguists the verb phrases consist only of main verbs, auxiliary verbs, and other infinitive or participle constructions [6] (for example, *Anagrelide* [*may interacts*]$_{VP}$ [*with any of these compounds*]$_{PP}$). We have decided to adopt the last definition, that is, we define a verb phrase as a syntactic structure that is composed of a main verb and, optionally, of auxiliary and modal verbs, but the complements are excluded of this structure. Unfortunately, MMTx offers an even simpler definition of verb phrase, because MMTx labels each verb as a *VP*. Forms of *to be* are labeled as $V/_{be}$. In order to group the main verb, its auxiliary or modal verbs, as well as its adverbial complements in the same verb phrase, we define the VP-pattern as: [VP|V/$_{be}$|VPG] (V/$_{be}$)? (NOT)? (ADV)? (VP|V/$_{be}$|VPG)? (TO VP)?. The *VP-pattern* is applied to sentences in order to merge their adjacent verb phrases into an extended verb phrase. If a sentence contains two or more extended VPs, then we can conclude that it is a complex or compound sentence. However, if a sentence only contains an extended *VP*, it is a simple sentence despite containing any conjunction. First column in Table 3 shows some sentences parsed by MMTx, while the second column shows the result of applying our *Vp-pattern* to them.

Once it has been determined that the sentence contains two or more clauses, the following step is to determine the type of sentence. Such information will be very useful in detecting the clause boundaries. In the English language, a *compound sentence* is composed of two or more independent clauses joined by a conjunction that can be a coordinator (coordinating conjunction: *for, and, nor, but, or,yet, so*), a correlative conjunction (*both, either, whether... or; not only... but also*) or an independent marker word (*however, moreover, furthermore, consequently, nevertheless, therefore*). Semicolons and commas can also function as conjunctions. If an independent marker occurs at the beginning of the sentence, then a semicolon or a comma should separate the clauses. If the second independent clause starts with an independent marker, then a semicolon or a comma is needed before the marker [27]. The independent markers can also occur in simple sentences, as in the following sentence: *However, initial dose modification is generally not necessary.*

A *complex sentence* has an independent clause joined with one or more subordinate clauses. Subordinate clauses contain both a subject and a verb, but do not express a complete thought. A complex sentence always has a relative pronoun (*who, that, which, whoever, whom, whomever, whose, whichever, whatever*) or a subordinator (*after, although, as, as if, because, before, even if, even though, if, in order to, since, though, unless, until, whatever, whether, when, whenever, while.*) that links the clauses. If the complex sentence begins with a subordinator, that is, the subordinate clause is at the beginning of the sentence, then the subordinate clause should end with a comma. On the other hand, if the independent clause is attached at the beginning of the main sentence and the subordinator is in the middle, then no comma is required [27].

Taking into account the above clues, we initially defined a set of lexical patterns for detecting clauses boundaries in compound and complex sentences (see Table 4). Relative clauses are a especial case, since, they often appear in the middle of a main clause, splitting it into two parts. If a sentence matches some of these patterns, then its clauses can be easily extracted from the matching.

However, these patterns are not always enough. Determining where a clause ends is not always a trivial task, since there might be commas or conjunctions internal to the clause. Moreover, some conjunctions can also function as prepositions (for example *for*) or as adverbs (for example *yet, so*). The problem regarding adverbs is easily resolved (at least in most of cases) because MMTx labels them as *CONJ* phrases when they function as coordinators (though sometimes MMTx mistakes the phrases or is not able to determine the types). The previous identification of appositions and coordinate structures allows to reduce the number

Table 3: How does MMTx label the verb phrases?.

Verb phrases detected by MMTx	Verb phrases joined by the VP-pattern
[Formal drug interaction studies]$_{NP}$ [have]$_{VP}$ [not]$_{ADV}$ [been]$_{V/be}$ [conducted]$_{VP}$ [with ORENCIA.]$_{PP}$	[Formal drug interaction studies]$_{NP}$ **[have not been conducted]**$_{VP}$ [with ORENCIA.]$_{PP}$
[The combination]$_{NP}$ [of methotrexate]$_{PP}$ [with acitretin]$_{PP}$ [is]$_{V/be}$ [also]$_{ADV}$ [contraindicated]$_{VP}$	[The combination]$_{NP}$ [of methotrexate]$_{PP}$ [with acitretin]$_{PP}$ **[is also contraindicated]**$_{VP}$

Table 4: Initial patterns for clause splitting

<table>
<tr><td rowspan="2">Compound sentences</td><td>CLAUSE$_1$(,|;)? [<i>indepMarker</i>|<i>coordinator</i>|;|,] CLAUSE$_2$</td></tr>
<tr><td><i>indepMarker(,)?</i> CLAUSE$_1$[,|;] CLAUSE$_2$</td></tr>
<tr><td rowspan="2">Complex sentences</td><td><i>depMarker(,)?</i> CLAUSE$_{subordinate}$, CLAUSE$_{main}$</td></tr>
<tr><td>CLAUSE$_{main}$ [<i>depMarker</i>|;|,] CLAUSE$_{subordinate}$</td></tr>
<tr><td>Relative Clauses</td><td><i>relativePronoun</i> (NP|PP|UNK|ADJ|APOS|COORD)? VP [NP|PP|UNK|ADJ|APOS|COORD]</td></tr>
</table>

of commas and conjunctions internal to a clause. However, for each comma or coordinator not included in any apposition or coordinate structure, it is required to know whether the clause ends or not in it. Therefore, the above patterns have been replaced with a set of heuristics based on the observation of fifty compound and complex sentences. These heuristics are encoded in algorithm 1.

Algorithm 1 Clause splitting in a compound or complex sentence S

```
Require: $S$!=NULL and its verbs have been joined into VPs by the VP-pattern.
    {$S$ is a sentence.}
Define $NUMVP$ as the number of verb phrases in $S$.
if $NUMVP$==1 then
    $S$ is a simple sentence {S only contains a indepedent clause.}
    return
end if
$INI$:=0. {This is the position where $S$ begins}
Look for a separator marker from $INI$ in $S$, that is, a coordinator, a subordinator, an independent marker, a semicolon or a comma
    {The coordinator or independent marker must be classified as a $CONJ$ phrase by MMTx.}
Save the found marker into the variable $MARKER$.
Define $FIN$ as the position where $MARKER$ begins.
while $MARKER$!=NULL do
    Define $CLAUSE$ as the substring between $INI$ and $FIN$.
    if $CLAUSE$ has any VP then
        Mark $CLAUSE$ as a clause in $S$. {The algorithm has found a clause. It must continue with the search of the rest of clauses}.
        Initialize $CLAUSE$ to NULL.
        To re-define $INI$ as the position where $MARKER$ ends.
    else
        Look for a separator marker from $FIN$ in $S$.
    end if
    Save the found marker into the variable $MARKER$.
    Define $FIN$ as the position where $MARKER$ begins.
end while
if CLAUSE!=NULL then
    Mark $CLAUSE$ as a clause.
end if
```

The input of the algorithm is the sentence in which its verb phrases have been joined by the VP-pattern. First of all, the algorithm must check that the sentence contains two or more clauses. Then, the sentence is reviewed while it contains any separator marker. A separator marker can be a coordinator, a independent marker, a dependent marker, a semicolon or a comma. The coordinators and subordinators must be labeled by MMTx as *CONJ* phrases, otherwise, they are not considered as conjunctions. Then, the algorithm iteratively finds candidate clauses, that is, a substring of the sentence between markers. If the candidate clause contains a verb phrase, then it is considered as clause. The algorithm is able to decide the kind of clause, that is, independent or subordinate.

6.1 Rules for Sentence Simplification

Once appositions and coordinate propositions have been recognized, and compound and complex sentences have been split into clauses, it is possible to apply a set of rules for sentence simplification. These rules allow to simplify the complex and compound sentences in simple sentences. Then, the pattern-based approach for DDI extraction will be applied to these simpler sentences.

We have adapted some of the simplification rules presented in [24]. This work also recognized relative clauses, apposition, coordination and subordination, however its goal was not relation extraction, but to provide syntactic simplification of sentences for improving the performance of NLP applications such as text summarization or machine translation. [24] proposes seven simplification rules to generate new simplified sentences from the clauses of the complex and compound sentences. Table 5 presents the rules adapted in our approach and some sentences broken up into simpler sentences by these rules.

7. LEXICAL PATTERNS FOR DDI EXTRACTION

Despite the richness of natural language expressions, in practice, DDI are often expressed by a limited number of constructions. This fact favors the use of patterns as an excellent method for their extraction. Based on her professional experience and the corpus observation, our pharmacist defined a set of lexical patterns (see Table 6) to capture the various language constructions used to express DDI in pharmacological texts. Moreover, the pharmacist provided a set of synonyms for the verbs that can indicate a possible DDI.

8. EVALUATION

This section explains in detail the experiments that we have carried out. We consider as baseline system, so called

Table 5: Rules to generate new simplified sentences from the clauses. The clause *CLAUSE*$_{REL(NP)}$ means that it is attached to the noun phrase *NP*.

Simplification Rules	Generated sentences
MARKER(,)? CLAUSE$_1$, CLAUSE$_2$	(1) CLAUSE$_1$ (2) CLAUSE$_2$
CLAUSE$_1$(,)? MARKER CLAUSE$_2$	(1) CLAUSE$_1$ (2) CLAUSE$_2$
CLAUSE$_1$ NP CLAUSE$_{REL(NP)}$ CLAUSE$_2$	(1) CLAUSE$_1$ NP CLAUSE$_2$ (2) NP CLAUSE$_{REL(NP)}$

Table 6: Lexical patterns to extract DDIs.

Id	Pattern
P1	DRUG *MODAL*? *ADV*? (INTERACT|INTERFERE) WITH WORD$_{0..5}$ (OF)? DRUG
P2	DRUG *MODAL*? *ADV*? INCREASE$_{syn}$ WORD$_{0..5}$ (OF)? DRUG
P3	DRUG *MODAL*? *ADV*? DECREASE$_{syn}$ WORD$_{0..5}$ (OF)? DRUG
P4	DRUG *MODAL*? *ADV*? ALTER$_{syn}$ WORD$_{0..5}$ (OF)? DRUG
P5	DRUG *MODAL*? BE *ADV*? INCREASE$_{syn}$ WORD$_{0..5}$ (BY)? DRUG
P6	DRUG *MODAL*? BE *ADV*? DECREASE$_{syn}$ WORD$_{0..5}$ (BY)? DRUG
P7	DRUG *MODAL*? BE *ADV*? ALTER$_{syn}$ WORD$_{0..5}$ (BY)? DRUG
P8	COADMINISTRATION OF DRUG (WITH|AND|PLUS) DRUG *MODAL*? *ADV*? [INCREASE$_{syn}$|DECREASE$_{syn}$INTERACT$_{syn}$||ALTER$_{syn}$]
P9	COADMINISTRATION OF DRUG (WITH|AND|PLUS) DRUG *MODAL*? *BE*? *ADV*? RESULT$_{syn}$ (TO|WITH|IN) [INCREASE$_{syn}$|DECREASE$_{syn}$INTERACT$_{syn}$||ALTER$_{syn}$]
P10	CAUTION *MODAL*? *ADV*? *BE*? USED WHEN DRUG *WORD*? (WITH|AND|PLUS) DRUG *BE*? ADMINISTERED *CONCURRENTLY*?
P11	PATIENTS TREATED (WITH)? DRUG (WITH|AND|PLUS) DRUG (CONCURRENTLY)? MODAL BE OBSERVED
P12	INTERACTION (OF|BETWEEN) DRUG (AND|WITH|PLUS) DRUG MODAL? (BE)? WORD$_{0..3}$ (OBSERVED|INCREASE|DECREASE|ALTER)

allDDIs, the case in which every pair of drugs that co-occur in a sentence are assumed to interact. This baseline yields the maximum recall, but a low precision (11%) and a baseline F-measure of 19%. Let us start with describing the most basic experiment in which neither coordinations, appositions nor clauses are tackled, that is, the lexical patterns are directly applied to the text of sentences. First of all, sentences are parsed by MMTx and drug names are identified by the DrugNer system [23]. Then, only those sentences that contain two or more drug names are selected and the drug names are replaced by the label *DRUG*$._{index}$, where *index* shows the order of each drug in the list of drugs that occur in sentence. Finally, the set of lexical patterns is applied to the text of the sentence. When a sentence has been correctly matched with a pattern, it must be checked if the matching string includes the negative adverb (*NOT*). If it is not included, then a possible interaction has been found. Drug names that occur in the matching are retrieved, and the pair of drug names is proposed as a DDI.

In the second experiment, appositions and coordinate structures are identified in text by the set of syntactic patterns described in sections 5 and 4. The lexical patterns were modified to consider these structures, that is, they are extended for including the labels *APPOSITION* and *COORD* as possible elements participating in the interactions. Thus, for this experiment, *DRUG*:= [*DRUG*|*APPOSITION*|*COORD]*. The procedure of matching pattern for this experiment is explained in algorithm 2.

Table 7 shows the global and individual pattern performance. The basic experiment achieves a reasonable precision (67.30%), but very low recall (14.07%). The average number of DDI detected by each pattern is 35.5 (the number total of DDI in the DrugDDI corpus is 3,160). Regarding the individual pattern performance, the highest recall is achieved by the pattern *P2* and the highest precision by the pattern *P8*. Regarding the second experiment, recall is improved by the inclusion of the appositions and coordinate structures, however, precision is lower. The average number of DDI detected by each pattern is 64.83. The pattern *P2* still achieves the highest recall, and the highest precision is obtained by the pattern *P10*. Therefore, the detection of these structures achieves to improve the recall (almost 12%) with a significant decrease in precision of almost 19%. This decrease can be attributed to the errors introduced during the syntactic processing.

We now explain the last experiment that combines the detection of appositions, coordinate structures, clause splitting and simplification rules. First of all, appositions and coordinate clauses are detected by applying the previous described procedure (algorithm 2) step by step until the sixth step. Then, the algorithm 1 (described in Section 6) is applied to sentences in order to split the complex and compound sentences into their clauses. New sentences are generated from these clauses by the simplification rules described in subsection 6.1. Finally, the previous procedure of matching pattern (algorithm 2) is applied to these new sentences from the seventh step. Results are shown in Table 7. While the inclusion of appositions and coordinate structures achieved to improve the recall, and therefore, the f-measure, the detection of clauses has not improved the performance. This is mainly due to the fact that many interactions occurring in complex sentences often span several clauses (for example, *The Cmax of norethindrone was 13% higher when it was coadministered with gabapentin*). The lexical patterns are not able to capture these interactions.

Table 7: Results

	Patterns			Coord+Apos			Coord+Apos+Clauses		
Id	**P(%)**	**R(%)**	**F**$_{\beta\,=\,1(\%)}$	**P(%)**	**R(%)**	**F**$_{\beta\,=\,1(\%)}$	**P(%)**	**R(%)**	**F**$_{\beta\,=\,1(\%)}$
P1	60.71	0.56	1.11	59.17	2.35	4.51	59.17	2.35	4.51
P2	69.51	3.77	7.15	54.78	7.00	12.42	55.75	6.41	11.50
P3	53.28	2.15	4.13	44.74	3.93	7.23	46.18	4.00	7.36
P4	68.64	2.68	5.15	52.67	4.56	8.39	52.69	4.53	8.34
P5	79.17	0.63	1.25	48.19	1.32	2.57	52.00	1.29	2.51
P6	60.00	0.30	0.59	39.13	39.39	0.43	0.85	0.30	0.59
P7	77.42	0.79	1.57	60.00	0.99	1.95	58.33	0.93	1.82
P8	100.00	0.50	0.99	57.45	0.89	1.76	52.54	1.02	2.01
P9	73.81	1.02	2.02	68.18	1.98	3.85	68.18	1.98	3.85
P10	85.71	0.20	0.40	50.00	73.33	0.36	0.72	0.10	0.20
P11	87.88	0.96	1.90	19.69	1.26	2.36	20.21	1.29	2.42
P12	47.06	0.53	1.05	35.19	0.63	1.23	35.19	0.63	1.23
GLOBALS	67.30	14.07	23.28	48.69	25.70	33.64	48.89	24.81	32.92

Algorithm 2 Pattern Matching including the detection of appositions and coordinate structures

1: The text is split into sentences. Each sentence is treated separately.

2: Each sentence is parsed by MMTx providing lexical information, POS tags, syntactic types, and semantic information on its words, tokens and phrases.

3: The DrugNer identifies the drug names in the sentence.

4: Select those sentences that contains two or more drug names.

5: The shallow syntactic information provided by MMTx is used to generate a sequence of the syntactic types of the phrases in the sentence.

6: The syntactic patterns are applied to the sequence in order to detect its appositions and coordinate structures. If some structure is detected, this will be replaced with the label $APPOSITION._{index}$ or $COORD._{index}$, depending on the case.

7: The drug names are replaced by the label *DRUG.index*

8: The text of the sentence is generated by concatenating their text phrases (except the text of the appositions and coordinate structures).

9: If generated text is matched by some pattern and the matching string does not contain the negative adverb, a candidate interaction has been found.

10: If the matching string contains appositions or coordinate structures, these must be unfolded in order to obtain the individual interacting elements (as many as the number of elements which make up each structure) and build the list of interactions.

11: The list of interactions is evaluated on the DrugDDI corpus.

9. CONCLUSIONS

In this paper, we have proposed a hybrid method that combines the resolution of complex linguistic constructions and matching pattern.

Regarding the resolution of the linguistic constructions, as it was pointed out in Section 8, most of the errors are due to mistakes introduced in the MMTx level and the difficulty of resolving nested clauses, so frequent in biomedical texts. Also, we are aware that our clause splitting method is too simplistic to deal with the complexity of biomedical sentences. Another shortcoming of our approach is that the negation has been addressed by an heuristic too simplistic. So, the following sentence matches the pattern *P1*, however, it does not represent any interaction:

- While studies have not shown $DRUG_1$ *interact with* $DRUG_2$, caution should be exercised.

This is due to the previous negation *studies have not shown* has not been detected. This could be avoid by a deeper treatment of the negation.

Future directions include trying to identify and resolve the errors of MMTx, improving our clause splitting algorithm, proposing new suitable simplification rules to regenerate the simple sentences from clauses, checking what occurs if the resolutions are applied in a different order, studying the utility of other corpora such as Genia-GR or Penn Treebank in the evaluation, and increasing the size of the corpus and annotating it with these linguistic constructions in order to apply machine learning methods. In addition, we will carry out a more exhaustive treatment of negation and modality in sentences.

Concerning the performance in the extraction of DDI, the variability of natural language expression makes it difficult for our method to accurately detect all semantic relations occurring in text since sentences conveying the same relation may be composed lexically and syntactically differently. Inversely, sentences that are lexically common may not necessarily convey the same relation. Thus, our lexical patterns are not enough to identify many of the interactions. Future work will include application of the SPINDEL system [9] to semi-automatically learn new patterns from biomedical texts. SPINDEL is a bootstrapping method which has achieved good results for named entity recognition task in general domain.

10. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish research projects: MA2VICMR consortium (S2009/TIC-1542, www.mavir.net), a network of excellence funded by the Madrid Regional Government and TIN2007-67407-C03-01 (BRAVO: Advanced Multimodal and Multilingual Question Answering). The authors are grateful to María Segura Bedmar, manager of the Drug Information Center of the Móstoles University Hospital, Spain, for her valuable assistance in the annotation of the corpus and evaluation of the system.

11. REFERENCES

[1] S. Ahmed, D. Chidambaram, H. Davulcu, and C. Baral. IntEx: A Syntactic Role Driven Protein-Protein Interaction Extractor for Bio-Medical Text. *Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 54–61, 2005.

[2] A. Airola, S. Pyysalo, J. Bjorne, T. Pahikkala, F. Ginter, and T. Salakoski. A graph kernel for protein-protein interaction extraction. In *Proceedings of BioNLP*, pages 1–9, 2008.

[3] A. Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17, 2001.

[4] R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, 2005.

[5] N. Burton-Roberts. Nominal apposition. *Foundations of language*, 13(3):391–419, 1975.

[6] N. Calzolari, A. Lenci, and A. Zampolli. The EAGLES/ISLE computational lexicon working group for multilingual computational lexicons. In *Proceedings of the First International Workshop on Multimedia Annotation. Tokyo (Japan)*, 2001.

[7] Y. Chen, F. Liu, and B. Manderick. Normalizing Interactor Proteins and Extracting Interaction Protein Pairs using Support Vector Machines. In *Proceedings of the BioCreative II. 5 Workshop 2009 on Digital Annotations*, page 29, 2009.

[8] G. Curme and G. Curme. *A grammar of the English language: syntax*. Verbatim Books, 1977.

[9] C. de Pablo-Sánchez and P. Martínez. Building a Graph of Names and Contextual Patterns for Named Entity Classification. In *31st European Conference on Information Retrieval*, 2009.

[10] S. Duda, C. Aliferis, R. Miller, A. Statnikov, and K. Johnson. Extracting drug-drug interaction articles from MEDLINE to improve the content of drug databases. In *AMIA Annual Symposium Proceedings*, volume 2005, page 216, 2005.

[11] W. Francis. The Structure of American English. *New York*, pages 409–17, 1958.

[12] C. Fries. *The structure of English: An introduction to the construction of English sentences*. Harcourt, Brace, 1952.

[13] P. Hansten. Drug interaction management. *Pharmacy World & Science*, 25(3):94–97, 2003.

[14] M. Huang, X. Zhu, and M. Li. A hybrid method for relation extraction from biomedical literature. *International Journal of Medical Informatics*, 75(6):443–455, 2006.

[15] O. Jespersen and J. McCawley. *Analytic syntax*. University of Chicago Press, 1984.

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

[17] M. Krallinger, F. Leitner, and A. Valencia. The BioCreative II.5 challenge overview. In *Proceedings of the BioCreative II. 5 Workshop 2009 on Digital Annotations*, page 19, 2009.

[18] S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50, 2007.

[19] A. Rodríguez-Terol, C. Camacho, et al. Calidad estructural de las bases de datos de interacciones. *Farmacia Hospitalaria*, 33(03):134, 2009.

[20] R. Sætre, K. Sagae, and J. Tsujii. Syntactic features for protein-protein interaction extraction. In *Proceedings of the International Symposium on Languages in Biology and Medicine (LBM short oral presentations)*, 2007.

[21] I. Segura-Bedmar, M. Crespo, and C. de Pablo-Sánchez. Score-based approach for Anaphora Resolution in Drug-Drug Interactions Documents. In *Natural Language Processing and Information Systems*, volume 5040, 2009.

[22] I. Segura-Bedmar, M. Crespo, C. de Pablo-Sánchez, and P. Martínez. Resolving anaphoras for the extraction of drug-drug interactions in pharmacological documents. *BMC bioinformatics*, 11(Suppl 2):S1, 2010.

[23] I. Segura-Bedmar, P. Martínez, and M. Segura-Bedmar. Drug name recognition and classification in biomedical texts A case study outlining approaches underpinning automated systems. *Drug Discovery Today*, 13(17-18):816–823, 2008.

[24] A. Siddharthan. Syntactic simplification and text cohesion. *Research on Language & Computation*, 4(1):77–109, 2006.

[25] K. Verspoora, C. Roeder, H. Johnson, K. Cohen, W. Baumgartner, and L. Hunter. Information Extraction of Normalized Protein Interaction Pairs Utilizing Linguistic and Semantic Cues. In *Proceedings of the BioCreative II. 5 Workshop 2009 on Digital Annotations*, page 37, 2009.

[26] E. Williams. Across-the-board rule application. *Linguistic Inquiry*, pages 31–43, 1978.

[27] J. Wingersky, J. Boerner, and D. Holguin-Balogh. *Writing paragraphs and essays: Integrating reading, writing, and grammar skills*. Heinle, 2008.

[28] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(Database issue):D901–6, Jan. 2008.

[29] Z. Yang, H. Lin, and Y. Li. BioPPISVMExtractor: A protein–protein interaction extractor for biomedical literature using SVM and rich feature sets. *Journal of Biomedical Informatics*, 2009.

[30] D. Zhou and Y. He. Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, 2007.