The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts

Conference Paper · January 2011

CITATIONS
85

READS
552

3 authors:

Isabel Segura-Bedmar
University Carlos III de Madrid
62 PUBLICATIONS **1,073** CITATIONS
SEE PROFILE

Paloma Martinez
University Carlos III de Madrid
236 PUBLICATIONS **1,902** CITATIONS
SEE PROFILE

Daniel Sanchez-Cisneros
University Carlos III de Madrid
6 PUBLICATIONS **100** CITATIONS
SEE PROFILE

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction.

Huelva, Spain, September 7th, 2011.

Edited by:

Isabel Segura-Bedmar, Universidad Carlos III de Madrid, Spain

Paloma Martínez, Universidad Carlos III de Madrid, Spain

Daniel Sánchez Cisneros, Universidad Carlos III de Madrid, Spain

Welcome

We are pleased to welcome to the DDIExtraction 2011 workshop (First Challenge Task on Drug-Drug Interaction Extraction) being held in Huelva, Spain on September 7 and co-located with the 27th Conference of the Spanish Society for Natural Language Processing, SEPLN 2011. On behalf of the organizing committee, we would like to thank you for your participation and hope you enjoy the workshop.

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. 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.

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. Most investigation has focused on biological relationships (genetic and protein interactions (PPI)) due mainly to the availability of annotated corpora in the biological domain, facilitating the evaluation of approaches. Few approaches have focused on the extraction of DDIs.

The DDIExtraction (Extraction of drug-drug interactions) task focuses on the extraction of drug-drug interactions from biomedical texts and aims to promote the development of text mining and information extraction systems applied to the pharmacological domain in order to reduce time spent by the health care professionals reviewing the literature for potential drug-drug interactions. Our main goal is to have a benchmark for the comparison of advanced techniques, rather than competitive aspects.

We would like to thank all the participating teams for submitting their runs and panelists for presenting their work. We also acknowledge all the members of the program committee for providing their support in reviewing contributions. Finally, we would like to thank to Universidad de Huelva, especially the organizers of the SEPLN 2011 conference and all the people that help us to make this workshop possible.

The DDIExtraction 2011 Workshop was partially supported by MA2VICMR consortium (S2009/TIC-1542) and MULTIMEDICA research project (TIN2010-20644-C03-01).

The DDIExtraction 2011 organizing committee

Committees

Organizing Committee

Isabel Segura-Bedmar, Universidad Carlos III de Madrid, Spain

Paloma Martínez, Universidad Carlos III de Madrid, Spain

Daniel Sánchez Cisneros, Universidad Carlos III de Madrid, Spain

Program Committee

Manuel Alcántara, Universidad Autónoma de Madrid, Spain

Manuel de Buenaga, Universidad Europea de Madrid (UEM), Spain

Cesar de Pablo-Sánchez, Innosoft Factory S.L., Spain

Alberto Díaz, Universidad Complutense de Madrid, Spain

Ana García Serrano, Universidad Nacional Educación a Distancia (UNED), Spain

Ana Iglesias, Universidad Carlos III de Madrid, Madrid, Spain

Antonio J. Jimeno Yepes, National Library of Medicine (NLM), Washington DC, USA

Jee-Hyub Kim, EMBL-EBI, UK.

Florian Leitner, Structural Computational Biology Group, CNIO, Spain

Paloma Martínez Fernández, Universidad Carlos III de Madrid, Spain

Jose Luís Martínez Fernández, Universidad Carlos III de Madrid, Spain

Antonio Moreno Sandoval, Universidad Autónoma de Madrid, Spain

Roser Morante, CLiPS - Linguistics Department, University of Antwerp, Belgium

Paolo Rosso, Universidad Politécnica de Valencia, Spain

Isabel Segura-Bedmar, Universidad Carlos III de Madrid, Spain

Table of contents

The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts 1

Isabel Segura-Bedmar, Paloma Martínez, and Daniel Sánchez-Cisneros

Relation Extraction for Drug-Drug Interactions using Ensemble Learning 11

Philippe Thomas, Mariana Neves, Illes Solt, Domonkos Tikk, and Ulf Leser

Two Different Machine Learning Techniques for Drug-drug Interaction Extraction 19

Md. Faisal Mahbub Chowdhury, Asma Ben Abacha, Alberto Lavelli, and Pierre Zweigenbau

Drug-drug Interaction Extraction Using Composite Kernels 27

Md. Faisal Mahbub Chowdhury, and Alberto Lavelli

Drug-Drug Interaction Extraction with RLS and SVM Classiffers 35

Jari Björne, Antti Airola, Tapio Pahikkala, and Tapio Salakoski

Feature selection for Drug-Drug Interaction detection using machine-learning based approaches 43

Anne-Lyse Minard, Anne-Laure Ligozat, Brigitte Grau, and Lamia Makour

Automatic Drug-Drug Interaction Detection: A Machine Learning Approach With Maximal Frequent Sequence Extraction 51

Sandra Garcia-Blasco, Santiago M. Mola-Velasco, Roxana Danger, and Paolo Rosso

A machine learning approach to extract drug–drug interactions in an unbalanced cataset.. 59

Jacinto Mata Vázquez, Ramón Santano, Daniel Blanco, Marcos Lucero, and Manuel J. Maña López

Drug-Drug Interactions Discovery Based on CRFs SVMs and Rule-Based Methods 67

Stefania Rubrichi, Matteo Gabetta, Riccardo Bellazzi, Cristiana Larizza, and Silvana Quaglini

An experimental exploration of drug-drug interaction extraction from biomedical texts..... 75

Man Lan, Jiang Zhao, Kezun Zhang, Honglei Shi, and Jingli Cai

Extraction of drug-drug interactions using all paths graph kernel 83

Shreyas Karnik, Abhinita Subhadarshini, Zhiping Wang, Luis Rocha and Lang Li

The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts

Isabel Segura-Bedmar, Paloma Martínez, and Daniel Sánchez-Cisneros

Universidad Carlos III de Madrid, Computer Science Department,
Avd. Universiad, 30, 28911 Leganés, Madrid, Spain
{isegura,pmf,dscisner}@springer.com
http://labda.inf.uc3m.es/

Abstract. We present an evaluation task designed to provide a framework for comparing different approaches to extracting drug-drug interactions from biomedical texts. We define the task, describe the training/test data, list the participating systems and discuss their results. There were 10 teams who submitted a total of 40 runs.

Keywords: Biomedical Text Mining, Drug-Drug Interaction Extraction

1 Task Description and Related Work

A drug-drug interaction (DDI) occurs when one drug influences the level or activity of another drug. 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 [12]. 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. The management of DDIs is a critical issue, therefore, due to the overwhelming amount of information available [8].

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.

Different systems have been developed for the extraction of biomedical relations, particularly PPIs, from texts. Nevertheless, few approaches have been proposed to the problem of extracting DDIs in biomedical texts. We developed two different approaches for DDI extraction. Since no benchmark corpus was available to evaluate our approaches to DDI extraction, we created the DrugDDI corpus annotated with 3,160 DDIs. Our first approach is a hybrid linguistic

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 1–9,
Huelva, Spain, September 2011.

approach [13] that combines shallow parsing and syntactic simplification with pattern matching. This system yielded a precision of 48.69%, a recall of 25.70% and an F-measure of 33.64%. Our second approach [14] is based on a supervised machine learning technique, more specifically, the shallow linguistic kernel proposed in Giuliano et al. (2006) [7]. It achieved a precision of 51.03%, a recall of 72.82% and an F-measure of 60.01%.

In order to stimulate research in this direction, we have organized the challenge task DDIExtraction2011. Likewise the BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) challenge evaluation has devoted to provide a common frameworks for evaluation of text mining driving progress in text mining techniques applied to the biological domain, our purpose is to create a benchmark dataset and evaluation task that will enable researchers to compare their algorithms applied to the extraction of drug-drug interactions.

2 The DrugDDI corpus

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. We hope that the corpus serves to encourage the NLP community to conduct further research in the field of pharmacology.

As source of unstructured textual information on drugs and their interactions, we used the DrugBank database[17]. This database is a rich resource combining chemical and pharmaceutical information of approximately 4,900 pharmacological substances. 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, toxicity, etc. Of particular interest to this study, DrugBank offers the field 'Interactions' (it is no longer available) that contained a link to a document describing DDIs in unstructured texts. DrugBank provides a file with the names of approved drugs[1], approximately 1,450. We randomly chose 1,000 drug names and used the RobotMaker[2], a screen-scrapper application, to download the interaction documents for these drugs. We only retrieved a total of 930 documents since some drugs did not have any linked document. Due to the cost-intensive and time consuming nature of the annotation process, we decided to reduce the number of documents to be annotated and only considered 579 documents. We believe that these texts are a reliable and representative source of data for expressing DDI since the language used is mostly devoted to descriptions of DDIs. Additionally, the highly specialized pharmacological language is very similar to that found in the Medline pharmacology abstracts.

These documents were then analyzed by the UMLS MetaMap Transfer (MMTx) [2] tool performing sentence splitting, tokenization, POS-tagging, shal-

[1] http://www.drugbank.ca/downloads
[2] http://openkapow.com/

low syntactic parsing (see Figure 1) and linking of phrases with UMLS Metathesaurus concepts. 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. An experienced pharmacist reviewed the UMLS Semantic Network as well as the semantic annotation provided by MMTx and recommended us the inclusion of the following UMLS semantic types as possible types of interacting drugs: Clinical Drug (clnd), Pharmacological Substance (phsu), Antibiotic (antb), Biologically Active Substance (bacs), Chemical Viewed Structurally (chvs) and Amino Acid, Peptide, or Protein (aapp).

The principal value of the DrugDDI corpus undoubtedly comes from its DDIs annotations. To obtain these annotations, all documents 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. Only sentences with two or more drugs were considered and the annotation was made sentence by sentence. Figure 1 shows an example of an annotated sentence that contains 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 containing these interacting drugs are also annotated, providing an easily access to the related concepts provided by MMTx. As mentioned, Figure 1 shows three DDIs: the first DDI represents an interaction between *Aspirin* and *probenecid*, the second one an interaction between *aspirin* and *sulfinpyrazone*, and the last one a DDI between *aspirin* and *phenylbutazone*.

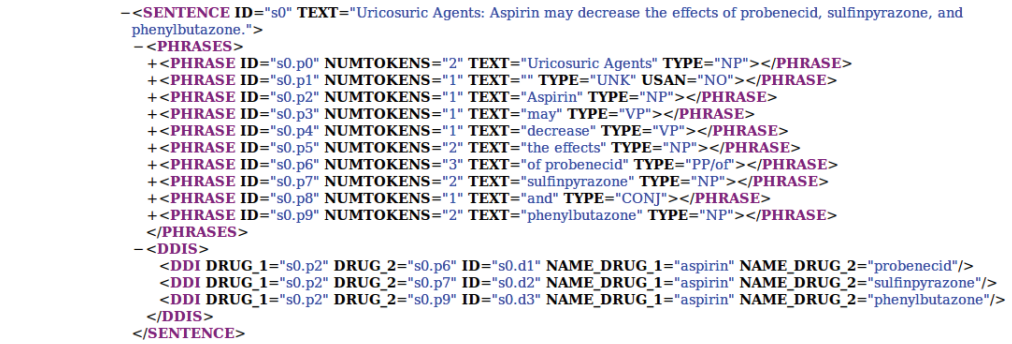


Fig. 1. Example of DDI annotations.

The DrugDDI corpus is also provided in the unified format for PPI corpora proposed in Pyysalo et al. [11] (see Figure 2). This shared format could attract attention of groups studying PPI extraction because they could easily adapt their systems to the problem of DDI extraction. The unified XML format does not contain any linguistic information provided by MMTx. The unified format only

```
−<sentence id="DrugDDI.d346.s0" origId="s0" text="Uricosuric Agents: Aspirin may decrease the effects of probenecid,
 sulfinpyrazone, and phenylbutazone.">
   <entity id="DrugDDI.d346.s0.e0" origId="s0.p0" charOffset="0-17" type="drug" text="Uricosuric Agents"/>
   <entity id="DrugDDI.d346.s0.e1" origId="s0.p2" charOffset="19-26" type="drug" text="Aspirin"/>
   <entity id="DrugDDI.d346.s0.e2" origId="s0.p6" charOffset="55-65" type="drug" text="probenecid"/>
   <entity id="DrugDDI.d346.s0.e3" origId="s0.p7" charOffset="67-81" type="drug" text="sulfinpyrazone"/>
   <entity id="DrugDDI.d346.s0.e4" origId="s0.p9" charOffset="87-101" type="drug" text="phenylbutazone"/>
   <pair id="DrugDDI.d346.s0.p0" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e1" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p1" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e2" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p2" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e3" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p3" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e4" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p4" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e2" interaction="true"/>
   <pair id="DrugDDI.d346.s0.p5" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e3" interaction="true"/>
   <pair id="DrugDDI.d346.s0.p6" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e4" interaction="true"/>
   <pair id="DrugDDI.d346.s0.p7" e1="DrugDDI.d346.s0.e2" e2="DrugDDI.d346.s0.e3" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p8" e1="DrugDDI.d346.s0.e2" e2="DrugDDI.d346.s0.e4" interaction="false"/>
   <pair id="DrugDDI.d346.s0.p9" e1="DrugDDI.d346.s0.e3" e2="DrugDDI.d346.s0.e4" interaction="false"/>
</sentence>
```

Fig. 2. The unified XML format.

Table 1. Basic statistics on the DrugDDI corpus.

	Number	Avg. per document
Documents	579	
Sentences	5,806	10.03
Phrases	66,021	114.02
Tokens	127,653	220.47
Sentences with at least one DDI	2,044	3.53
Sentences with no DDI	3,762	6.50
DDIs	3,160	5.46 (0.54 per sentence)

provides the sentences, their drugs and their interactions. Each entity (drug) includes reference (origId) to its id phrase in the MMTX format corpus text in which the corresponding drug appears. For each sentence from the DrugDDI corpus represented in the unified XML format, its DDI candidate pairs should be generated from the different drugs appearing therein. Each DDI candidate pair is represented as a *pair* node in which the ids of the interacting drugs are registered in its *e1* and *e2* attributes. If the pair is a DDI, the *interaction* attribute must be set to *true*, and *false* value otherwise.

Table 1 shows basic statistics of the DrugDDI corpus. In general, the size of biomedical corpora is quite small and usually does not exceed 1,000 sentences. The average number of sentences per MedLine abstract was estimated at 7.2 ± 1.9 [18]. Our corpus contains 5,806 sentences with 10.3 sentences per document on average. MMTx identified a total of 66,021 phrases of which 12.5% (8,260) are drugs. The average number of drug mentions per document was 24.9, and the average number of drug mentions per sentence was 2.4. The corpus contains a total of 3,775 sentences with two or more drug mentions, although only 2,044 sentences contain at least one interaction. With the assistance of a pharmacist, a total of 3,160 DDIs were with an average of 5.46 DDIs per document and 0.54 per sentence.

DDI extraction can be 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. 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: $\{(D_i, D_j) : D_i, D_j \epsilon D, 1 <= i, j <= N, i \neq j, i < j\}$. 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 task. 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 $C_{N,2} = \binom{N}{2}$.

Table 2 shows the total number of relation examples or instances generated from the DrugDDI corpus. Among the 30,757 candidate drug pairs, only 3,160 (10.27%) were marked as positive interactions (i.e., DDIs) while 27,597 (89.73%) were marked as negative interactions (i.e., non-DDIs).

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

Set	Documents	Examples	Positives	Negatives
Train	437 (75.5%)	25,209	2,421 (9.6%)	22,788 (90.4%)
Final Test	142 (24.5%)	5,548	739 (13.3%)	4,809 (86.7%)
Total	579	30,757	3,160 (10.27%)	27,597 (89.73%)

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 systems. In order to build the training dataset used for development tests, 75% of the DrugDDI corpus files (435 files) were randomly selected for the training dataset and the remaining 25% (144 files) is 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).

3 The participants

The task of extracting drug-drug interactions from biomedical texts has attracted the participation of 10 teams who submitted 40 runs. Table 4 lists the teams,

Table 3. Training and testing datasets.

Set	Documents	Sentences	Drugs	DDIs
Training	435	4,267	11,260	2,402
Final Test	144	1,539	3,689	758
Total	579	5,806	14,949	3,160

their affiliations, the number of runs submitted and the description of their systems.

The runs' performance information in terms of precision, recall, F-measure and accuracy, appears in Table 5.

Table 4. Short description of the teams.

Team	Institution	Runs	Description
WBI	Humboldt-Universitat Berlin	5	combination of several kernels and a case-based reasoning (CBR) system using a voting approach
FBK-HLT	Fondazione Bruno Kessler - HLT	5	composite kernels using the MEDT, PST and SL kernels
LIMSI-FBK	LIMSI - Fondazione Bruno Kessler	1	a feature-based method using SVM and a composite kernel-based method.
UTurku	University of Turku	4	machine learning classifiers such as SVM and RLS; DrugBank and MetaMap
LIMSI-CNRS	LIMSI-CNRS	5	a feature-based method using libSVM and SVMPerf
bnb_nlel	Universidad Politécnica de Valencia	1	a feature-based method using Random Forests
laberinto-uhu	Universidad de Huelva	5	a feature-based method using classical classifiers such as SVM, Nave Bayes, Decision Trees, Adaboost
DrIF	University of Pavia (Department Mario Stefanelli)	4	two machine learning-based (CFFs and SVMs) and one hybrid approach which combines CRFs and a rule-based technique.
ENCU	East China Normal University	5	a feature-based method using SVM.
IUPUITMGroup	Indiana University-Purdue University Indianapolis	5	all paths graph (APG) kernel

Table 5. Precision, recall, F-measure and accuracy over each run's performance.

Team	run	TP	FP	FN	TN	P	R	F	Acc
WBI	5	543	354	212	5917	0.6054	0.7192	0.6574	0.9194
WBI	4	529	332	226	5939	0.6144	0.7007	0.6547	0.9206
WBI	2	568	465	187	5806	0.5499	0.7523	0.6353	0.9072
WBI	1	575	585	180	5686	0.4957	0.7616	0.6005	0.8911
WBI	3	319	362	436	5909	0.4684	0.4225	0.4443	0.8864
LIMSI-FBK	1	532	376	223	5895	0.5859	0.7046	0.6398	0.9147
FBK-HLT	4	529	377	226	5894	0.5839	0.7007	0.6370	0.9142
FBK-HLT	1	513	344	242	5927	0.5986	0.6795	0.6365	0.9166
FBK-HLT	2	560	458	195	5813	0.5501	0.7417	0.6317	0.9071
FBK-HLT	3	534	423	221	5848	0.5580	0.7073	0.6238	0.9083
FBK-HLT	5	544	674	211	5597	0.4466	0.7205	0.5514	0.8740
Uturku	3	520	376	235	5895	0.5804	0.6887	0.6299	0.9130
Uturku	4	370	179	385	6092	0.6740	0.4901	0.5675	0.9197
Uturku	2	368	197	387	6074	0.6513	0.4874	0.5576	0.9169
Uturku	1	350	172	405	6099	0.6705	0.4636	0.5482	0.9179
LIMSI-CNRS	1	490	398	265	5873	0.5518	0.6490	0.5965	0.9056
LIMSI-CNRS	2	491	402	264	5869	0.5498	0.6503	0.5959	0.9052
LIMSI-CNRS	4	462	380	293	5891	0.5487	0.6119	0.5786	0.9042
LIMSI-CNRS	5	373	264	382	6007	0.5856	0.4940	0.5359	0.9081
LIMSI-CNRS	3	388	470	367	5801	0.4522	0.5139	0.4811	0.8809
BNBNLEL	1	420	266	335	6005	0.6122	0.5563	0.5829	0.9145
laberinto-uhu	1	335	335	420	5936	0.5000	0.4437	0.4702	0.8925
laberinto-uhu	2	324	371	431	5900	0.4662	0.4291	0.4469	0.8859
laberinto-uhu	3	368	551	387	5720	0.4004	0.4874	0.4397	0.8665
laberinto-uhu	4	238	153	517	6118	0.6087	0.3152	0.4154	0.9046
laberinto-uhu	5	193	107	562	6164	0.6433	0.2556	0.3659	0.9048
DrIF	1	369	545	386	5725	0.4037	0.4887	0.4422	0.8675
DrIF	4	369	545	386	5726	0.4037	0.4887	0.4422	0.8675
DrIF	3	317	456	438	5815	0.4101	0.4199	0.4149	0.8728
DrIF	2	196	110	559	6161	0.6405	0.2596	0.3695	0.9048
ENCU	5	351	836	404	5435	0.2957	0.4649	0.3615	0.8235
ENCU	3	324	830	431	5441	0.2808	0.4291	0.3394	0.8205
ENCU	1	580	3456	175	2815	0.1437	0.7682	0.2421	0.4832
ENCU	2	713	4781	42	1490	0.1298	0.9444	0.2282	0.3135
ENCU	4	206	424	549	5847	0.3270	0.2728	0.2975	0.8615
IUPUITMGroup	4	193	1457	562	4814	0.1170	0.2556	0.1605	0.7126
IUPUITMGroup	1	237	2005	518	4266	0.1057	0.3139	0.1582	0.6409
IUPUITMGroup	2	127	943	628	5328	0.1187	0.1682	0.1392	0.7764
IUPUITMGroup	3	125	937	630	5334	0.1177	0.1656	0.1376	0.7770
IUPUITMGroup	5	110	770	645	5501	0.1250	0.1457	0.1346	0.7986

4 Discussion

The best performance is achieved by the team WBI [15]. Its system combines several kernels (APG [1], SL [7], kBSPS [16]) and a case-based reasoning (CBR) (called MOARA [10]) using a voting approach. In particular, the combination

of the kernels APG, SL and the MOARA system yields the best F-measure (0.6574).

The team FBK-HLT [5] proposes new composite kernels using well-known kernels such as MEDT [6], PST [9] and SL [7]. Similarly, the team LIMSI-FBK [4] combines the same kernels (MEDT, PST and SL) and a feature-based method using SVM. This system achieves an F-measure of 0.6398.

The team Uturku [3] proposes a feature-based method using the classifiers SVM and RLS. Features used by the classifiers include syntactic information (tokens, dependency types, POS tags, text, stems, etc) and semantic knowledge from DrugBank and MetaMap. This system achieves an F-measure of 0.6299.

In general, approaches based on kernels methods achieved better results than the classical feature-based methods. Most systems have used primarily syntactic information, however semantic information has been poorly used.

5 Conclusion

This paper describes a new semantic evaluation task, Extraction of drug-drug interactions from biomedical texts. We have accomplished our goal of providing a framework and a benchmark data set to allow for comparisons of methods for this task. The results that the participating systems have reported show successful approaches to this difficult task, and the advantages of kernel-based methods over classical machine learning classifiers.

The success of the task shows that the framework and the data are useful resources. By making this collection freely accessible, we encourage further research into this domain. Moreover, next SemEval-3 (6th International Workshop on Semantic Evaluations[3]) to be held in summer 2013 has scheduled the "Extraction of drug-drug interactions from biomedical Texts" task [4]. In order to accomplish this new task, the current corpus is being extended to collect new data test.

Acknowledgements

This study was funded by the projects MA2VICMR (S2009/TIC-1542) and MULTIMEDICA (TIN2010-20644-C03-01). The organizers are particularly grateful to all participants who contributed to detect annotation errors in the corpus.

References

1. Airola, A., Pyysalo, S., Bjorne, J., Pahikkala, T., Ginter, F., Salakoski, T.: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC bioinformatics 9(Suppl 11), S2 (2008)

[3] http://www.cs.york.ac.uk/semeval/

[4] http://www.cs.york.ac.uk/semeval/proposal-16.html

2. Aronson, A.R.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Annual AMIA Symposium pp. 17–21 (Jan 2001)
3. Björne, J., Airola, A., Pahikkala, T., Salakoski, T.: Drug-drug interaction extraction with rls and svm classiffers. In: Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) (2011)
4. Chowdhury, M., Abacha, A., Lavelli, A., P., Z.: Two different machine learning techniques for drug-drug interaction extraction. In: Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) (2011)
5. Chowdhury, M., Lavelli, A.: Drug-drug interaction extraction using composite kernels. In: Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) (2011)
6. Chowdhury, M., Lavelli, A., Moschitti, A.: A study on dependency tree kernels for automatic extraction of protein-protein interaction. ACL HLT 2011 p. 124
7. 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). pp. 401–408 (2006)
8. Hansten, P.D.: Drug interaction management. Pharmacy World & Science 25(3), 94–97 (2003)
9. Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. pp. 335–es. Association for Computational Linguistics (2004)
10. Neves, M., Carazo, J., Pascual-Montano, A.: Extraction of biomedical events using case-based reasoning. In: Proceedings of the Workshop on BioNLP: Shared Task. pp. 68–76. Association for Computational Linguistics (2009)
11. Pyysalo, S., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC bioinformatics 9(Suppl 3), S6 (2008)
12. Rodríguez-Terol, A., Camacho, C., Others: Calidad estructural de las bases de datos de interacciones. Farmacia Hospitalaria 33(03), 134 (2009)
13. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchez, C.: A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. BMC Bioinformatics 12(Suppl 2), S1 (2011)
14. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchez, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics In Press, Corrected Proof (2011)
15. Thomas, P., Neves, M., Solt, I., Tikk, D., Leser, U.: Relation extraction for drug-drug interactions using ensemble learning. In: Proceedings of the First Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) (2011)
16. Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., Leser, U.: A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. PLoS Computational Biology 6(7), e1000837 (2010)
17. Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., Hassanali, M.: DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic acids research 36(Database issue), D901–6 (Jan 2008)
18. Yu, H.: Towards answering biological questions with experimental evidence: automatically identifying text that summarize image content in full-text articles. Annual AMIA Symposium proceedings pp. 834–8 (Jan 2006)

Relation Extraction for Drug-Drug Interactions using Ensemble Learning

Philippe Thomas[1], Mariana Neves[1], Illés Solt[2], Domonkos Tikk[2], and Ulf Leser[1]

[1] Humboldt-Universität zu Berlin, Knowledge Management in Bioinformatics, Unter den Linden 6, 10099 Berlin, Germany, {thomas, neves, leser}@informatik.hu-berlin.de

[2] Budapest University of Technology and Economics, Department of Telecommunications and Media Informatics, Magyar tudósok körútja 2, 1117 Budapest, Hungary {solt, tikk}@tmit.bme.hu

Abstract. We describe our approach for the extraction of drug-drug interactions from literature. The proposed method builds majority voting ensembles of contrasting machine learning methods, which exploit different linguistic feature spaces. We evaluated our approach in the context of the DDI Extraction 2011 challenge, where using document-wise cross-validation, the best single classifier achieved an F_1 of 57.3 % and the best ensemble achieved 60.6 %. On the held out test set, our best run achieved an F_1 of 65.7 %.

Keywords: Text mining; Relation extraction; Machine learning; Ensemble learning

1 Introduction

Most biomedical knowledge appears first as research results in scientific publications before it is distilled into structured knowledge bases. For researchers and database curators there is an urgent need to cope with the fast increase of biomedical literature [6]. Biomedical text mining currently achieves good results for named entity recognition (NER), *e.g.* gene/protein-names and recognition of single nucleotide polymorphisms [3, 11]. A recent trend is the extraction of simple or complex relations between entities [7].

In this work, we describe our approach for the extraction of drug-drug interactions (DDI) from text that was also the core task of the DDI Extraction 2011 challenge[1]. DDIs describe the interference of one drug with another drug and usually lead to an enhanced, reduced, neutralized, or even toxic drug effect. For example: "Aspirin administered in combination with Warfarin can lead to bleeding and has to be avoided." DDI effects are thus crucial to decide when (not) to administer specific drugs to patients.

[1] http://labda.inf.uc3m.es/DDIExtraction2011/

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 11–18, Huelva, Spain, September 2011.

1.1 Problem definition

The DDI challenge[1] consisted of one task, namely the identification of interactions between two drugs. This interaction is binary and undirected, as target and agent roles are not labeled. In the challenge setting, recognition of drug names was readily available.

2 Methods

Binary relation extraction is often tackled as a pair-wise classification problem between all entities mentioned within one sentence. Thus a sentence with n entities contains at most $\binom{n}{2}$ interacting pairs.

Corpus annotations have been made available in two different formats. (1) contained only the documents with respective drug annotations in a format previously used for protein-protein interactions (PPIs) [13]. (2) additionally contained linguistic information such as part-of-speech tags and shallow parses. Further phrases were annotated with corresponding UMLS concepts. This information has been automatically derived using MetaMap and incorporated by the organizers. We exclusively used (1) and extended it with linguistic information as described in the following subsection.

2.1 Preprocessing

Sentences have been parsed using Charniak–Lease parser [8] with a self-trained re-ranking model augmented for biomedical texts [10]. Resulting constituent parse trees have been converted into dependency graphs using the Stanford converter [4]. In the last step we created an augmented XML following the recommendations of [2]. This XML encompasses tokens with respective part-of-speech tags, constituent parse tree, and dependency parse tree information. Properties of the training and test corpora are shown in Table 1. Please note that the number of positive and negative instances in the test set has been made available after the end of the challenge. A more detailed description of the DDI corpus can be found in [14].

Corpus	Sentences	Pairs		
		Positive	Negative	Total
Training	4,267	2,402	21,425	23,827
Test	1,539	755	6,271	7,026

Table 1: Basic statistics of the DDI corpus training and test sets.

2.2 Kernel based approaches

Tikk *et al.* [17] systematically analyzed 9 different machine learning approaches for the extraction of undirected binary protein-protein interactions. In their analysis, three kernel have been identified of being superior to the remaining six approaches, namely all-paths graph (APG) [2], k-band shortest path spectrum (kBSPS) [17], and the shallow linguistic (SL) [5] kernel. The SL kernel uses only shallow linguistic features, *i.e.* word, stem, part-of-speech tag and morphologic properties of the surrounding words. kBSPS builds a classifier on the shortest dependency path connecting the two entities. It further allows for variable mismatches and also incorporates all nodes within distance k from the shortest path. APG builds a classifier using surface features and a weighting scheme for dependency parse tree features. For a more detailed description of the kernel we refer to the original publications. The advantage of these three methods has been replicated and validated in a follow up experiment during the i2b2 relation extraction challenge [15]. In the current work we also focus on these three methods.

Experiments have been done using an open-source relation extraction framework.[2] Entities were blinded by replacing the entity name with a generic string to ensure the generality of the approach. Without entity blinding a classifier uses drug names as features, which clearly affects its generalization abilities on unseen entity pairs.

2.3 Case-based reasoning

In addition to kernel classifiers, we also used a customized version of Moara, an improvement of the system that participated in the BioNLP'09 Event Extraction Challenge [12]. It uses case-based reasoning (CBR) for classifying the drug pairs. CBR [1] is a machine learning approach that represents data with a set of features. In the training step, first the cases from the training data are learned and then saved in a knowledge base. During the testing step, the same representation of cases is used for the input data, the documents are converted to cases and the system searches the base for cases most similar to the case-problem.

Each drug pair corresponds to one case. This case is represented by the local context, *i.e.*, the tokens between a drug pair. We have limited the size of the context to 20 tokens (pairs separated by more tokens are treated as false). The features may be related to the context as a whole or to each of the tokens that is part of the context. Features may be set as mandatory or optional, here no feature was defined as mandatory. As features we considered part-of-speech tag, role and lemma.

The part-of-speech tag is the one obtained during the pre-processing of the corpus. The role of the token is set to *DRUG* in case that the token is annotated as drug that takes part in the interaction. No role is set to drugs which are part of the context and are not part of the interaction pair, as well as the remaining

[2] `http://informatik.hu-berlin.de/forschung/gebiete/wbi/ppi-benchmark`

tokens. The lemma feature is only assigned for the non-role tokens using the Dragon toolkit [18], otherwise the feature is not set. See Table 2 for an example.

Context	Features		
	Lemma	POS	Role
Buprenorphine	*drug*	NN	DRUG
is	be	VBZ	–
metabolized	metabolized	VBN	–
to	to	TO	–
norbuprenorphine	norbuprenorphine	NN	–
by	by	IN	–
cytochrome	*drug*	NN	DRUG

Table 2: Example of features for the two interacting drugs described in the sentence "Buprenorphine is metabolized to norbuprenorphine by cytochrome." The lemma *drug* is the result of entity blinding.

During the searching step, Moara uses a filtering strategy in which it looks for a case with exactly the same values for the features, *i.e.*, it tries to find cases with exactly the same values for the mandatory features and matching as many optional features as possible. For the case retrieved in this step, a similarity between those and the original case is calculated by comparing the values of the corresponding features using a global alignment. This methodology was proposed as part of the CBR algorithm for biomedical term classification in the MaSTerClass system [16]. By default, for any feature, the insertion and deletion costs are 1 (one) and the substitution cost is 0 (zero) for equal features with equal values, and 1 (one) otherwise. However, we have also defined specific costs for the part-of-speech tag feature which were based on the ones used in the MaSTerClass system. We decided to select those cases whose global alignment score is below a certain threshold, automatically defined as proposed in [16]. The final solution, *i.e.*, whether the predicted category is "positive" or "negative", is given by a voting scheme among the similar cases. When no similar case if found for a determined pair, or if the pair was not analyzed at all due to its length (larger than 20), the "negative" category is assigned by default.

2.4 Ensemble learning

Previous extraction challenges showed that combinations of classifiers may achieve better results than any single classifier itself [7, 9]. Thus we experimented with different combinations of classifiers by using a majority voting scheme.

3 Results

3.1 Cross validation

In order to compare the different approaches, we performed document-wise 10-fold cross validation on the training set (see Table 3). It has been shown that such a setting provides more realistic performance estimates than instance-wise cross validation [2]. All approaches have been tested using the same splits to ensure comparability. For APG, kBSPS, and SL; we followed the parameter optimization strategy described in [17].

Method		Performance		
Type	Name	P	R	F_1
Kernel	APG	53.4	63.1	57.3
	kBSPS	35.9	53.5	42.7
	SL	45.4	**71.6**	55.3
Case-based reasoning	Moara	43.3	40.7	41.6
Ensemble	APG/Moara/SL	**59.0**	63.0	**60.6**
	APG/kBSPS/SL	53.2	65.2	58.3

Table 3: Document-wise cross-validation results on the training set for selected methods.

3.2 Test dataset

For the test set we submitted results for APG, SL, Moara, and the two majority voting ensembles. Results for kBSPS have been excluded, as only 5 submissions were permitted and kBSPS and Moara achieve similar results in F_1. The official results achieved on the test set are shown Table 4.

Run	Method	P	R	F_1
WBI-2	APG	55.0	75.2	63.4
WBI-1	SL	49.6	**76.2**	60.1
WBI-3	Moara	46.8	42.3	44.4
WBI-5	APG/Moara/SL	60.5	71.9	**65.7**
WBI-4	APG/kBSPS/SL	**61.4**	70.1	65.5

Table 4: Relation extraction results on the test set.

4 Discussion

4.1 Cross-validation

The document-wise cross-validation results show that SL and APG outperform the remaining methods. kBSPS and Moara are on a par with each other but F_1 is about 15 percentage points (pp) inferior to SL or APG. Even though the results of kBSPS and Moara are inferior, as ensemble members they are capable of improving F_1 on the training corpus. The combination APG/Moara/SL performs about 2.3 pp better in F_1 than the APG/kBSPS/SL ensemble and yields an overall improvement of 3.3 pp in comparison to the best single classifier (APG). Single method results are in line with previously published results using these kernel for other domains [15, 17]. Again the SL kernel, which uses only shallow linguistic information, achieves considerably good results. This indicates that shallow information is often sufficient for relation extraction.

We estimated the effect of entity blinding by temporarily disabling it. This experiment has been performed for SL exclusively and yielded an increase of 1.7 pp in F_1. This effect was accompanied by an increase of 3.6 pp in precision and a decrease of 3 pp in recall. We did not disable entity blinding for the submitted runs, as such classifiers would be biased towards known DDIs and less capable of finding novel DDIs, the ultimate goal of DDI extraction.

4.2 Test dataset

For the challenge all four classifier have been retrained using the whole training corpus using the parameter setting yielding the highest F_1in the training phase. Our best run achieved 65.7 % in F_1.

Between training and test results we observe a perfect correlation for F_1 (Kendall's tau (τ) of 1.0). Thus the evaluation corpus affirms the general ranking of methods determined on the training corpus. The effect of ensemble learning is less pronounced on the test set but with 2.3 pp still notable.

4.3 Error analysis

To have an impression about the errors generated by these classifiers, we manually analyzed drug mention pairs that were not correctly classified by any method (APG, kBSPS, Moara, and SL). Performing cross-validation, the DDI training corpus contained 442 (1.85 %) such pairs, examples are given in Figure 1.

We identified a few situations that may have caused difficulties: issues with the annotated corpus and linguistic constructs not or incorrectly handled by our methods. Annotation inconsistencies we encountered include dubious drug entity annotations (B1, B6 in Figure 1), and ground truth annotations that were either likely incorrect (B3) or could not be verified without the context (A4, B4). As for linguistic constructs, our methods lack co-reference resolution (A1, B5) and negation detection (A6, B7), and they also fail to recognize complex formulations (A5, B2). As a special case, conditional constructs belong to both

A1 **Probenecid** interferes with renal tubular secretion of *ciprofloxacin* and produces an increase in the level of **ciprofloxacin** in serum.

A2 *Drugs* which may enhance the neuromuscular blocking action of **TRACRIUM** include: **enflurane**;

A3 While not systematically studied, certain *drugs* may induce the metabolism of **bupropion** (e.g., **carbamazepine**, *phenobarbital*, *phenytoin*).

A4 **Auranofin** should not be used together with *penicillamine* (**Depen**, *Cuprimine*), another *arthritis medication*.

A5 These **drugs** in combination with very high doses of **quinolones** have been shown to provoke convulsions

A6 **Diclofenac** interferes minimally or not at all with the protein binding of *salicylic acid* (20% decrease in binding), *tolbutamide*, **prednisolone** (10% decrease in binding), or *warfarin*.

(a) False negatives

B1 **Dofetilide** is eliminated in the kidney by **cation**ic secretion.

B2 Use of **sulfapyridine** with these **medicines** may increase the chance of side effects of these *medicines*.

B3 **Haloperidol** blocks dopamine receptors, thus inhibiting the central stimulant effects of **amphetamines**.

B4 This interaction should be given consideration in patients taking **NSAIDs** concomitantly with **ACE inhibitors**.

B5 No dose adjustment of **bosentan** is necessary, but increased effects of **bosentan** should be considered.

B6 **Epirubicin** is extensively metabolized by the **liver**.

B7 **Gabapentin** is not appreciably metabolized nor does it interfere with the metabolism of commonly coadministered **antiepileptic drugs**.

(b) False positives

Fig. 1: Examples of drug mention pairs not classified correctly by any of our methods. The two entities of the pair are typeset in bold, others in italic.

groups, they are nor consistently annotated nor consistently classified by our methods (A2, A3, B2). Furthermore, we found several examples that are not affected by any of the above situations.

5 Conclusion

In this paper we presented our approach for the DDI Extraction 2011 challenge. Primarily, we investigated the re-usability of methods previously proven efficient for relation extraction in other biomedical sub-domains, notably protein-protein interaction (PPI) extraction. In comparison to PPI extraction corpora, the training corpus is substantially larger and also exhibits a higher class imbalance towards negative instances. Furthermore, we experimented with basic ensembles to increase overall performance and conducted a manual error analysis to pinpoint weaknesses in the applied methods. Our best result consisted of a majority voting ensemble of three methodically different classifiers.

Acknowledgments

PT was supported by German Federal Ministry of Education and Research (grant No 0315417B), MN by German Research Foundation, and DT by Alexander von Humboldt Foundation.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39–59 (1994)
2. Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., Salakoski, T.: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics 9 Suppl 11, S2 (2008)
3. Caporaso, J.G., Baumgartner, W.A., Randolph, D.A., Cohen, K.B., Hunter, L.: MutationFinder: a high-performance system for extracting point mutation mentions from text. Bioinformatics 23(14), 1862–1865 (Jul 2007)
4. De Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: LREC 2006. vol. 6, pp. 449–454 (2006)
5. Giuliano, C., Lavelli, A., Romano, L.: Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In: Proc. of EACL'06. Trento, Italy (2006)
6. Hunter, L., Cohen, K.B.: Biomedical language processing: what's beyond PubMed? Mol Cell 21(5), 589–594 (Mar 2006)
7. Kim, J., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J.: Overview of BioNLP'09 shared task on event extraction. In: Proc. of BioNLP'09. pp. 1–9 (2009)
8. Lease, M., Charniak, E.: Parsing biomedical literature. In: Proc. of IJCNLP'05. pp. 58–69 (2005)
9. Leitner, F., Mardis, S., Krallinger, M., Cesareni, G., Hirschman, L., Valencia, A.: An overview of BioCreative II. 5. IEEE IEEE/ACM Transactions on Computational Biology and Bioinformatics pp. 385–399 (2010)
10. McClosky, D.: Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Ph.D. thesis, Brown University (2010)
11. Morgan, A.A., Lu, Z., Wang, X., Cohen, A.M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., Sun, C., Liu, H., Torres, R., Krauthammer, M., Lau, W.W., Liu, H., Hsu, C.N., Schuemie, M., Cohen, K.B., Hirschman, L.: Overview of BioCreative II gene normalization. Genome Biol 9 Suppl 2, S3 (2008)
12. Neves, M., Carazo, J.M., Pascual-Montano, A.: Extraction of biomedical events using case-based reasoning. In: Proc. of NAACL 2009. pp. 68–76 (2009)
13. Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics 9 Suppl 3, S6 (2008)
14. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchezti, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. J Biomed Inform (Apr 2011)
15. Solt, I., Szidarovszky, F.P., Tikk, D.: Concept, Assertion and Relation Extraction at the 2010 i2b2 Relation Extraction Challenge using parsing information and dictionaries. In: Proc. of i2b2/VA Shared-Task. Washington, DC (2010)
16. Spasic, I., Ananiadou, S., Tsujii, J.: MaSTerClass: a case-based reasoning system for the classification of biomedical terms. Bioinformatics 21(11), 2748–2758 (Jun 2005)
17. 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 6 (2010)
18. Zhou, X., Zhang, X., Hu, X.: Dragon Toolkit: Incorporating Auto-Learned Semantic Knowledge into Large-Scale Text Retrieval and Mining. In: Proc. of ICTAI'07. vol. 2, pp. 197–201 (2007)

Two Different Machine Learning Techniques for Drug-Drug Interaction Extraction

Md. Faisal Mahbub Chowdhury[2,3], Asma Ben Abacha[1],
Alberto Lavelli[2], and Pierre Zweigenbaum[1]

[1] LIMSI-CNRS, BP 133 - F-91403 Orsay Cedex, France
[2] HLT Research Unit, Fondazione Bruno Kessler (FBK), Trento, Italy
[3] Department of Information Eng. and Computer Science, University of Trento, Italy
chowdhury@fbk.eu, abacha@limsi.fr, lavelli@fbk.eu, pz@limsi.fr

Abstract. Detection of drug-drug interaction (DDI) is an important task for both patient safety and efficient health care management. In this paper, we explore the combination of two different machine-learning approaches to extract DDI: *(i)* a feature-based method using a SVM classifier with a set of features extracted from texts, and *(ii)* a kernel-based method combining 3 different kernels. Experiments conducted on the DDIExtraction2011 challenge corpus (unified format) show that our method is effective in extracting DDIs with 0.6398 F_1.

Keywords: Drug-Drug Interaction, machine learning, feature-based method, kernel-based method, tree kernel, shallow linguistic kernel.

1 Introduction

The *drug-drug interaction (DDI)* is a condition when one drug influences the level or activity of another. Detection of DDI is crucial for both patient safety and efficient health care management.

The objective of the *DDIExtraction2011 challenge*[4] was to identify the state of the art for automatically extracting DDI from biomedical articles. We participated in this challenge with a system combining two different machine learning methods to extract DDI: a feature-based method and a kernel-based one. The first approach uses a SVM classifier with a set of lexical, morphosyntactic and semantic features (e.g. trigger words, negation) extracted from texts. The second method uses a kernel which is a composition of a *mildly extended dependency tree (MEDT)* kernel [3], a *phrase structure tree (PST)* kernel [9], and a *shallow linguistic (SL)* kernel [5]. We obtained 0.6398 F-measure on the unified format of the challenge corpus.

In the rest of the paper, we first discuss related works (Section 2). In Section 3, we briefly discuss the dataset. Then in Section 4, we describe the feature-based system. Following that, in Section 5, the kernel-based system is presented. Evaluation results are discussed in Section 6. Finally, we summarize our work and discuss some future directions (Section 7).

[4] http://labda.inf.uc3m.es/DDIExtraction2011/

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 19–26, Huelva, Spain, September 2011.

2 Related Work

Several approaches have been applied to biological relation extraction (e.g. protein-protein interaction). Song et al. [13] propose a protein-protein interaction (PPI) extraction technique called PPISpotter by combining an active learning technique with semi-supervised SVMs to extract protein-protein interaction. Chen et al. [2] propose a PPI Pair Extractor (PPIEor), a SVM for binary classification which uses a linear kernel and a rich set of features based on linguistic analysis, contextual words, interaction words, interaction patterns and specific domain information. Li et al. [8] use an ensemble kernel to extract the PPI information. This ensemble kernel is composed with feature-based kernel and structure-based kernel using the parse tree of a sentences containing at least two protein names.

Much less approaches have focused on the extraction of DDIs compared to biological relation extraction. Recently, Segura-Bedmar et al. [11] presented a hybrid linguistic approach to DDI extraction that combines shallow parsing and syntactic simplification with pattern matching. The lexical patterns achieve 67.30% precision and 14.07% recall. With the inclusion of appositions and coordinate structures they obtained 25.70% recall and 48.69% precision. In another study, Segura-Bedmar et al. [12] used shallow linguistic (SL) kernel [5] and reported as much as an F_1 score of 0.6001.

3 Dataset

The DDIExtraction2011 challenge task required the automatic identification of DDIs from biomedical articles. Only the intra-sentential DDIs (i.e. DDIs within single sentence boundaries) are considered. The challenge corpus [12] is divided into training and evaluation dataset. Initially released training data consist of *435* abstracts and *4,267* sentences, and were annotated with *2,402* DDIs. During the evaluation phase, a dataset containing *144* abstracts and *1,539* sentences was provided to the participants as the evaluation data. Both datasets contain drug annotations, but only the training dataset has DDI annotations.

These datasets are made available in two formats: the so-called *unified* format and the *MMTx* format. The unified format contains only the tokenized sentences, while the MMTx format contains the tokenized sentences along with POS tag for each token.

We used the unified format data. In both training and evaluation datasets, there are some missing special symbols, perhaps due to encoding problems. The position of these symbols can be identified by the presence of the question mark *"?"* symbol. For example:

> *<sentence id="DrugDDI.d554.s14" origId="s14" text="Ergotamine or dihydroergotamine**?**acute ergot toxicity characterized by severe peripheral vasospasm and dysesthesia.">*

4 Feature-based Machine Learning Method

In this approach, the problem is modeled as a supervised binary classification task. We used a SVM classifier to decide whether a candidate DDI pair is an authentic DDI or not. We used the LibSVM tool [1] to test different SVM techniques (nu-SVC, linear kernel, etc.) and the script grid.py, provided by LibSVM, to find the best C and gamma parameters. We obtained the best results by using a C-SVC SVM with the Radial Basis kernel function with the following SVM parameters: c=1.0, g=0.0078125 and the set of features described in sections 4.1 and 4.2.

4.1 Features for DDI Extraction

We choose the following feature set to describe each candidate DDI pair (D1,D2):

- **Word Features.** Include Words of D1, words of D2, words between D1 and D2 and their number, 3 words before D1, 3 words after D2 and lemmas of all these words.
- **Morphosyntactic Features.** Include Part-of-speech (POS) tags of each drug words (D1 and D2), POS of the previous 3 and next 3 words. We use TreeTagger [5] to obtain lemmas and POS tags.
- **Other Features.** Include, among others, verbs between D1 and D2 and their number, first verb before D1 and first verb after D2.

4.2 Advanced features

In order to improve the performance of our system, we also incorporated some more advanced features related to this task. We used lists of interacting drugs, constructed by extracting drug couples that are related by an interaction in the training corpus. We defined a feature to represent the fact that candidate drug couples are declared in this list.

However, such lists are not sufficient to identify an interaction between new drug pairs. We also worked on detecting keywords expressing such relations in the training sentences. The following examples of positive (1,2) and negative (3) sentences show some of the keywords or trigger words that may indicate an interaction relationship.

1. *The oral bioavailability of enoxacin is **reduced** by 60% with **coadministration** of ranitidine.*
2. *Etonogestrel may **interact** with the following medications: acetaminophen (Tylenol) ...*
3. *There have been **no** formal studies of the **interaction** of Levulan Kerastick for Topical Solution with any other drugs ...*

To exploit these pieces of semantic information, we defined the following features:

[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

– **Trigger words.** This category of features indicates whether a specific trigger word occurs in the sentence (e.g. induce, inhibit). The trigger words were collected manually from the training corpus.
– **Negation.** This category of features indicates if a negation is detected (e.g. not, no) at a limited distance of characters before, between and after the two considered drugs.

5 Kernel-based Machine Learning Method

In this approach, the DDI extraction task was addressed using a system that exploits kernel-based method. Initially, the data had been pre-processed to obtain relevant information of the tokens of the sentences.

5.1 Data pre-processing

We used the Stanford parser[6] [7] for tokenization, POS-tagging and parsing of the sentences. Having *"?"* in the middle of a sentence causes parsing errors since the syntactic parser often misleadingly considers it as a sentence ending sign. So, we replace all *"?"* with *"@"*. To reduce tokenization errors, if a drug name does not contain an empty space character immediately before and after its boundaries, we inserted blank space characters in those positions inside the corresponding sentence. The SPECIALIST lexicon tool[7] was used to normalize tokens to avoid spelling variations and also to provide lemmas. The dependency relations produced by the parser were used to create dependency parse trees for corresponding sentences.

5.2 System description

Our system uses a composite kernel K_{SMP} which combines multiple tree and feature based kernels. It is defined as follows:

$$K_{SMP}(R_1, R_2) = K_{SL}(R_1, R_2) + w_1{*}K_{MEDT}(R_1, R_2) + w_2{*}K_{PST}(R_1, R_2)$$

where K_{SL}, K_{MEDT} and K_{PST} represent respectively shallow linguistic (SL) [5], mildly extended dependency tree (MEDT) [3] and PST [9] kernels, and w_i represents multiplicative constant(s). The values for all of the w_i used during our experiments are equal to 1.[8] The composite kernel is valid according to the kernel closure properties.

A dependency tree (DT) kernel, pioneered by Culotta et al. [4], is typically applied to the minimal or smallest common subtree of a dependency parse tree

[6] http://nlp.stanford.edu/software/lex-parser.shtml
[7] http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html
[8] Due to time constraints, we have not been able to perform extensive parameter tuning. We are confident that tuning of the multiplicative constant(s) (i.e. w_i) might produce even better performance.

that includes a target pair of entities. Such subtree reduces unnecessary information by placing word(s) closer to its dependent(s) inside the tree and emphasizes local features of the corresponding relation. However, sometimes a minimal subtree might not contain important cue words or predicates. The MEDT kernel addresses this issue using some linguistically motivated expansions. We used the best settings for the MEDT kernel reported by Chowdhury et al. [3] for protein-protein interaction extraction.

The PST kernel is basically the path-enclosed tree (PET) proposed by Moschitti [9]. This tree kernel is based on the smallest common subtree of a phrase structure parse tree, which includes the two entities involved in a relation.

The SL kernel is perhaps the best feature based kernel used so far for biomedical RE tasks (e.g. PPI and DDI extraction). It is a combination of global context (GC) and local context (LC) kernels. The GC kernel exploits contextual information of the words occurring before, between and after the pair of entities (to be investigated for RE) in the corresponding sentence; while the LC kernel exploits contextual information surrounding individual entities.

The jSRE system[9] is the implementation of these kernels using the support vector machine (SVM) algorithm. It should be noted that, by default, the jSRE system uses the ratio of negative and positive examples as the value of the cost-ratio-factor[10] parameter during SVM training.

Segura-Bedmar et al. [12] used the jSRE system for DDI extraction on the same corpus (in the MMTx format) that has been used during the DDIExtraction2011 challenge. They experimented with various parameter settings, and reported as much as an F_1 score of 0.6001. We used the same parameter settings (n-gram=3, window-size=3) with which they obtained their best result.

To compute the feature vectors of SL kernel, we used the jSRE system. The tree kernels and composite kernel were computed using the SVM-LIGHT-TK toolkit[11] [10, 6]. Finally, the ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter.

6 Results

We split the original training data into two parts by documents. One part contains around 63% of documents (i.e. 276 docs) that have around 67% of the "true" DDI pairs (i.e. 1603). The remaining documents belong to the other part. Both of the systems used these splits.

The first part is used for tuning the systems, while the second part is used as a test corpus for performance evaluation. The results on this test corpus are shown in Table 1. As we can see, the union (on the positive DDIs) of the outputs of each approach is higher than the individual output of the systems. We also calculated results for the intersection (only common positive DDIs) of

[9] http://hlt.fbk.eu/en/technology/jSRE

[10] This parameter value is the one by which training errors on positive examples would outweight errors on negative examples.

[11] http://disi.unitn.it/moschitti/Tree-Kernel.htm

the outputs which decreased the outcome. It is also important to note that the feature-based method (FBM) provides higher precision while the kernel-based method (KBM) obtains higher recall.

	FBM	**KBM**	**Union**	**Intersection**
Precision	0.5910	0.4342	0.4218	0.6346
Recall	0.3640	0.5277	0.6083	0.2821
F_1 Score	0.4505	0.4764	**0.4982**	0.3906

Table 1. Experimental results when trained on 63% of the original training documents and tested on the remaining.

Table 2 shows the evaluation results for the proposed approaches on the final challenge's evaluation corpus. The union of outputs of the systems has produced an F_1 score of **0.6398** which is better than the individual results. The behaviour of precision and recall obtained by the two approaches is the same as observed on the initial corpus (better precision for the feature-based approach and better recall for the kernel-based approach), however, the F_1 score of the kernel-based approach is quite close (F_1 score of *0.6365*) to that of the union.

	FBM	**KBM**	**Union**
True Positive	319	513	532
False Positive	133	344	376
False Negative	436	242	223
True Negative	6138	5927	5895
Precision	**0.7058**	0.5986	0.5859
Recall	0.4225	0.6795	**0.7046**
F_1 Score	0.5286	0.6365	**0.6398**

Table 2. Evaluation results provided by the challenge organisers.

7 Conclusion

In this paper, we have proposed the combination of two different machine learning techniques, a feature-based method and a kernel-based one, to extract DDIs. The feature-based method uses a set of features extracted from texts, including lexical, morphosyntactic and semantic features. The kernel-based method does not use features explicitly, but rather use a kernel composition of MEDT, PST and SL kernels. We have combined these two machine learning techniques and presented a simple union system in the DDIExtraction2011 challenge which obtained encouraging results. We plan to test and add more features in our first

method (e.g. UMLS semantic types), and to test the kernel-based method by assigning different weights to the individual kernels of the composite kernel. We also plan to perform further tests with other type of approaches like rule-based methods using manually constructed patterns. Another interesting future work can be to test other algorithms for the combination of different approaches (e.g. ensemble algorithms).

Acknowledgments

One of the authors, MFM Chowdhury, is supported by the PhD research grant of the project "eOnco - Pervasive knowledge and data management in cancer care".
One of the authors, Asma Ben Abacha, is partially supported by OSEO under the Quaero program.

References

[1] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
[2] Chen, Y., Liu, F., Manderick, B.: Extract protein-protein interactions from the literature using support vector machines with feature selection. Biomedical Engineering, Trends, Research and Technologies (2011)
[3] Chowdhury, M.F.M., Lavelli, A., Moschitti, A.: A study on dependency tree kernels for automatic extraction of protein-protein interaction. In: Proceedings of BioNLP 2011 Workshop. pp. 124–133. Association for Computational Linguistics, Portland, Oregen, USA (June 2011)
[4] Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04). Barcelona, Spain (2004)
[5] Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2006). pp. 401–408. Trento, Italy (2006)
[6] Joachims, T.: Making large-scale support vector machine learning practical. In: Advances in kernel methods: support vector learning, pp. 169–184. MIT Press, Cambridge, MA, USA (1999)
[7] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03). pp. 423–430. Association for Computational Linguistics, Sapporo, Japan (2003)
[8] Li, L., Ping, J., Huang, D.: Protein-protein interaction extraction from biomedical literatures based on a combined kernel. Journal of Information and Computational Science 7(5), 1065–1073 (2010)
[9] Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04). Barcelona, Spain (2004)
[10] Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Machine Learning: ECML 2006, Lecture Notes in Computer Science, vol. 4212, pp. 318–329. Springer Berlin / Heidelberg (2006)

[11] Segura-Bedmar, I., Martínez, P., Pablo-Sánchez, C.d.: Extracting drug-drug interactions from biomedical texts. BMC Bioinformatics 11(Suppl 5), 9 (2010)
[12] Segura-Bedmar, I., Martínez, P., Pablo-Sánchez, C.d.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics In Press, Corrected Proof, Available online (24 April, 2011)
[13] Song, M., Yu, H., Han, W.: Combining active learning and semi-supervised learning techniques to extract protein interaction sentences. In: International Workshop on Data Mining in Bioinformatics (2010)

Drug-drug Interaction Extraction Using Composite Kernels

Md. Faisal Mahbub Chowdhury[1,2] and Alberto Lavelli[1]

[1] HLT Research Unit, Fondazione Bruno Kessler (FBK), Trento, Italy
[2] Department of Information Eng. and Computer Science, University of Trento, Italy
{chowdhury,lavelli}@fbk.eu

Abstract. Detection of drug-drug interaction (DDI) is crucial for identification of adverse drug effects. In this paper, we present a range of new composite kernels that are evaluated in the DDIExtraction2011 challenge. These kernels are computed using different combinations of tree and feature based kernels. The best result that we obtained is an F_1 score of 0.6370 which is higher than the already published result on this same corpus.

Keywords: drugs, kernels, dependency tree, phrase structure tree, local context, global context.

1 Introduction

The *DDIExtraction2011 challenge*[3] provides a platform to identify the state of the art for drug-drug interaction (DDI) extraction from biomedical articles. We have participated in this challenge applying a range of new composite kernels. These kernels combine different combinations of *mildly extended dependency tree (MEDT)* kernel [2], *phrase structure tree (PST)* kernel [7], *local context (LC)* kernel [4], *global context (GC) kernel* [4] and *shallow linguistic (SL)* kernel [4].

The best result we have obtained is an F_1 score of **0.6370** by combining MEDT, PST and GC kernels on the unified format of the data. From the preprocessing of data to the extraction of DDIs using kernel compositions, our objective is to exploit the maximum information that could be learned from different representations of the data.

In the remaining of this paper, we discuss how we have addressed the DDI extraction task. In Section 2, we briefly discuss the dataset. Then in Section 3, the pre-processing steps are described. Following that, in Section 4, we mention the individual kernels which are the building blocks for our kernel compositions. Section 5 defines the proposed composite kernels. Evaluation results are discussed in Section 6. Finally, in Section 7 we summarize our work and present ideas for future work.

[3] http://labda.inf.uc3m.es/DDIExtraction2011/

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 27–33, Huelva, Spain, September 2011.

2 Dataset

The DDIExtraction2011 challenge task requires the automatic identification of DDIs from biomedical articles. Only the intra-sentential DDIs (i.e. DDIs within single sentence boundaries) are considered. The challenge corpus [9] is divided into training and evaluation datasets. Initially released training data consists of *435* abstracts and *4,267* sentences, and is annotated with *2,402* DDIs. During the evaluation phase, a dataset containing *144* abstracts and *1,539* sentences is provided to the participants as the evaluation data. Both the datasets contain drug annotations, but only the training dataset has DDI annotations.

These datasets are made available in two formats: the so-called *unified* format and the *MMTx* format. The unified format contains only the tokenized sentences, while the MMTx format contains the tokenized sentences along with POS tag for each token.

We have used the unified format data. We have found out that, in both training and evaluation datasets, there are some missing special symbols, perhaps due to encoding problems. The position of these symbols can be identified by the presence of the question mark *"?"* symbol. For example:

> *<sentence id="DrugDDI.d554.s14" origId="s14" text="Ergotamine or dihydroergotamine**?***acute ergot toxicity characterized by severe peripheral vasospasm and dysesthesia.">*

We have tried to randomly check whether the unified format and MMTx format datasets contain the same sentences. We have found that one of the randomly chosen sentences [4] does not include a *">"* character which exists as a token of the corresponding sentence inside the corresponding MTMx file. This suggests that there might be missing characters inside some sentences due to conversion errors of the html/xml special characters.

3 Data pre-processing

Our system is trained and evaluated on the unified format. We use the Stanford parser[5] [6] for tokenization, POS-tagging and parsing of the sentences.

Some of the characteristics of the data sets have required pre-processing steps to correctly handle the texts. Having *"?"* in the middle of a sentence causes parsing errors since the syntactic parser often misleadingly considers it as a sentence ending sign. So, we replaced all *"?"* with *"@"*. Additionally, to reduce tokenization errors, if a drug name does not contain an empty space character immediately before and after its boundaries, we insert space characters in those positions inside the corresponding sentence.

The SPECIALIST lexicon tool is used to normalize tokens to avoid spelling variations and also to provide lemmas. The dependency relations produced by the parser are used to create dependency parse trees for corresponding sentences.

[4] DrugDDI.d151.s11 of the file Flumazenil_ddi.xml.
[5] http://nlp.stanford.edu/software/lex-parser.shtml

4 Individual kernel approaches that we exploit

The approach adopted for our participation to the challenge is to exploit systems (or methodologies) that already obtained state-of-the-art results in the protein-protein interaction (PPI) extraction task and also in other RE tasks in domains such as newspaper articles. One of these systems [4] uses feature based kernels and is shown to be very effective for PPI extraction. We also consider tree kernel based approaches since they are the state of the art for various RE tasks especially from newspaper texts. All of the systems (or methodologies) are based on the support vector machine (SVM) algorithm for supervised machine learning.

4.1 Feature based kernels

Giuliano et al. [4] proposed a so called Shallow Linguistic (SL) kernel which is so far one of the best performing kernels used for biomedical RE. The SL kernel is defined as follows:

$$K_{SL}\ (R_1,\ R_2) = K_{LC}\ (R_1,\ R_2) + K_{GC}\ (R_1,\ R_2)$$

where K_{SL}, K_{GC} and K_{LC} correspond to SL, global context (GC) and local context (LC) kernels respectively. The GC kernel exploits contextual information of the words occurring before, between and after the pair of entities (to be investigated for RE) in the corresponding sentence; while the LC kernel exploits contextual information surrounding individual entities.

The jSRE system[6] provides an implementation of these kernels. It should be noted that, by default, jSRE uses the ratio of negative and positive examples as the value of the cost-ratio-factor[7] parameter during SVM training.

Segura-Bedmar et al. [9] used the jSRE system for DDI extraction on the same corpus (in the MMTx format) that has been used during the DDIExtraction2011 challenge. They experimented with various parameter settings, and reported an F_1 score of 0.6001. We used the same parameter settings (n-gram=3, window-size=3) with which they obtained their best result.

4.2 Tree kernels

One of the tree kernels that we have used is called mildly extended dependency tree (MEDT) kernel, proposed by Chowdhury et al. [2]. A dependency tree (DT) kernel, pioneered by Culotta et al. [3], is typically applied to the minimal or smallest common subtree of a dependency parse tree that includes a target pair of entities. Such subtree reduces unnecessary information by placing word(s) closer to its dependent(s) inside the tree and emphasizes local features of the

[6] http://hlt.fbk.eu/en/technology/jSRE

[7] This parameter value is the one by which training errors on positive examples would outweight errors on negative examples.

corresponding relation. However, sometimes a minimal subtree might not contain important cue words or predicates.

The MEDT kernel addresses this issue using some linguistically motivated extensions. The best settings for the MEDT kernel, that we used in our experiments for DDI extraction, observed by the authors on the AIMed protein-protein interaction dataset [1] is by expanding the minimal subtree with the following rule, and then by using unlexicalized partial trees (uPTs) [10] for similarity matching.

> *If the root of the minimal subtree is the head word of one of the interacting entities, then add the parent node (in the original DT tree) of the root node as the new root of the subtree.*

Apart from that, we have also used a phrase structure tree (PST) kernel which is basically the path-enclosed tree (PET) proposed by Moschitti [7]. This tree kernel is based on the smallest common subtree of a phrase structure parse tree, which includes the two entities involved in a relation.

5 Proposed kernel compositions

We propose the following composite kernels for DDI extraction:

- $K_{MP}\ (R_1, R_2) = w_1 * K_{MEDT}\ (R_1, R_2) + w_2 * K_{PST}\ (R_1, R_2)$
- $K_{LMP}\ (R_1, R_2) = K_{LC}\ (R_1, R_2) + w_3 * K_{MP}$
- $K_{GMP}\ (R_1, R_2) = K_{GC}\ (R_1, R_2) + w_3 * K_{MP}$
- $K_{SMP}\ (R_1, R_2) = K_{SL}\ (R_1, R_2) + w_3 * K_{MP}$

where K_{SL}, K_{MEDT}, K_{PST}, K_{LC} and K_{GC} represent SL, MEDT, PST, LC and GC kernels respectively, and w_i represents multiplicative constant(s). The values for all of the w_i used during our experiments are equal to 1. All the composite kernels are valid according to the closure properties of kernels.

The motivation behind using these new composite kernels is to combine varying representations (i.e. tree structures and flat structures) of different types of information (i.e. dependencies, syntactic information, and shallow linguistic features), and to see whether they can complement each other to learn a more robust model.

To compute the feature vectors of K_{SL}, K_{LC} and K_{GC}, we used the jSRE system. The tree kernels and composite kernels are computed using the SVM-LIGHT-TK toolkit[8] [8, 5]. Finally, the ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter.

[8] http://disi.unitn.it/moschitti/Tree-Kernel.htm

6 Evaluation Results

We have tuned all the composite kernels on the training data using 10-fold cross validation[9]. The results of these experiments are shown in Table 1. The experiments show that the best result is gained using the K_{GMP}. Both the K_{GMP} and K_{SMP} perform much better than the other kernels.

	K_{MP}	K_{SL}	K_{LMP}	K_{GMP}	K_{SMP}
Precision	0.4836	0.5109	0.5150	0.5522	0.5616
Recall	0.6249	0.6607	0.6507	0.6520	0.6336
F_1 Score	0.5452	0.5762	0.5750	**0.5980**	0.5954

Table 1. Results of 10-fold cross validation on the training data.

Table 2 shows the official evaluation results of our proposed kernels in the challenge. The results show a trend similar to the one of the cross-validation, with the composite tree kernel K_{MP} obtaining an F_1 score much lower than that of the other kernels. The combination of the tree and feature based kernels produces better results as the K_{SMP} got a better F_1 score than that of the K_{SL} or K_{MP} alone. But this combination also caused a drop in true and false positives. This suggests that such a combination produces a conservative model that requires more similarities in the features and structures of the candidate relations to be identified as DDIs than that of the kernels they are composed of.

As expected, the highest result is obtained by the K_{GMP} kernel. Implicitly, the tree kernels already exploit local contextual features as part of the tree structures. For example, the lemma of a (relevant) token is considered as a node inside the MEDT structure, while the order of the neighbouring tokens of an entity (along with their POS tags) are inherited inside the PST structure. So, excluding the LC kernel in the composite kernel might have been allowed to avoid data overfitting. Furthermore, entity blinding (i.e. generalizing the named entities instead of using their original names) is not considered for the basic feature set construction of LC kernel (please refer to the original paper of Giuliano et al. [4]). This might have caused systematic bias and resulted in lower performance.

7 Conclusion

In this paper, we have applied new composite kernels that exploit different types of tree structures and features. These include dependency trees and phrase structure trees as well as local and global contexts of the relevant entities. The kernels

[9] To obtain the overall performance we sum up the true positives, false positives, and false negatives of all the 10 folds, and then measure precision, recall and F_1 score from these figures.

	K_{MP}	K_{SL}	K_{LMP}	K_{GMP}	K_{SMP}
True Positive	544	560	534	529	513
False Positive	674	458	423	377	344
False Negative	211	195	221	226	242
True Negative	5597	5813	5848	5894	5927
Precision	0.4466	0.5501	0.558	0.5839	0.5986
Recall	0.7205	0.7417	0.7073	0.7007	0.6795
F_1 Score	0.5514	0.6317	0.6238	**0.6370**	0.6365

Table 2. Evaluation results on the test data provided by the challenge organisers.

have been evaluated on the DDIExtraction2011 challenge, and have achieved encouraging results.

Due to time constraints, we have not been able to perform extensive parameter tuning. We are confident that tuning of the multiplicative constant(s) (i.e. w_i) might produce even better performance. We also predict these kernels would be able to learn more accurate training models using a bigger training data, and would produce results better than that of the individual kernels which are their building blocks.

Acknowledgments

This work was carried out in the context of the project "eOnco - Pervasive knowledge and data management in cancer care".

References

[1] Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K., Wong, Y.W.: Comparative experiments on learning information extractors for proteins and their interactions. Artificial Intelligence in Medicine (Special Issue on Summarization and Information Extraction from Medical Documents) 33(2), 139–155 (2005)

[2] Chowdhury, M.F.M., Lavelli, A., Moschitti, A.: A study on dependency tree kernels for automatic extraction of protein-protein interaction. In: Proceedings of BioNLP 2011 Workshop. pp. 124–133. Association for Computational Linguistics, Portland, Oregen, USA (June 2011)

[3] Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, Spain (2004)

[4] Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'2006). pp. 401–408. Trento, Italy (2006)

[5] Joachims, T.: Making large-scale support vector machine learning practical. In: Advances in kernel methods: support vector learning, pp. 169–184. MIT Press, Cambridge, MA, USA (1999)

[6] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03). pp. 423–430. Association for Computational Linguistics, Sapporo, Japan (2003)
[7] Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. ACL '04, Barcelona, Spain (2004)
[8] Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Machine Learning: ECML 2006, Lecture Notes in Computer Science, vol. 4212, pp. 318–329. Springer Berlin / Heidelberg (2006)
[9] Segura-Bedmar, I., Martínez, P., Pablo-Sánchez, C.d.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics In Press, Corrected Proof, Available online (24 April, 2011)
[10] Severyn, A., Moschitti, A.: Fast cutting plane training for structural kernels. In: Proceedings of ECML-PKDD (2010)

Drug-Drug Interaction Extraction from Biomedical Texts with SVM and RLS Classifiers

Jari Björne,[1,2] Antti Airola,[1,2] Tapio Pahikkala[1] and Tapio Salakoski[1]

[1] Department of Information Technology, University of Turku
[2] Turku Centre for Computer Science (TUCS)
Joukahaisenkatu 3-5, 20520 Turku, Finland
firstname.lastname@utu.fi

Abstract. We introduce a system developed to extract drug-drug interactions (DDI) for drug mention pairs found in biomedical texts. This system was developed for the DDI Extraction First Challenge Task 2011 and is based on our publicly available Turku Event Extraction System, which we adapt for the domain of drug-drug interactions. This system relies heavily on deep syntactic parsing to build a representation of the relations between drug mentions. In developing the DDI extraction system, we evaluate the suitability of both text-based and database derived features for DDI detection. For machine learning, we test both support vector machine (SVM) and regularized least-squares (RLS) classifiers, with detailed experiments for determining the optimal parameters and training approach. Our system achieves a performance of 62.99% F-score on the DDI Extraction 2011 task.

1 Introduction

Biomedical Natural Language Processing (BioNLP) is the application of natural language processing methods to analyse textual data on biology and medicine, often research articles. Information extraction techniques can be used to mine large text datasets for relevant information, such as relations between specific types of entities.

In drug-drug interactions (DDI) one administered drug has an impact on the level or activity of another drug. Knowing all potential interactions is very important for physicians prescribing varying combinations of drugs for their patients. In addition to existing databases, drug-drug information could be extracted from textual sources, such as research articles. The DDI Extraction 2011 Shared Task[3] is a competitive evaluation of text mining methods for extraction of drug-drug interactions, using a corpus annotated for the task [13]. In the DDI corpus drug-drug interactions are represented as pairwise interactions between two drug mentions in the same sentence.

The DDI Extraction task organizers have also developed a shallow linguistic kernel method for DDI extraction, demonstrating the suitability of the dataset

[3] http://labda.inf.uc3m.es/DDIExtraction2011/

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 35–42,
Huelva, Spain, September 2011.

for machine learning based information extraction [13]. They have also extended this work into an online service for retrieving drug-drug interactions from the Medline 2010 database [12].

We apply for the DDI Shared Task our open source Turku Event Extraction System, which was the best performing system in the popular BioNLP 2009 Shared Task on Event Extraction, and which we have recently upgraded for the BioNLP 2011 Shared Task, demonstrating again competitive performance [1]. Event extraction is the retrieval of complex, detailed relation structures, but these structures are ultimately comprised of pairwise relations between text-bound entities. The Turku Event Extraction System has modules for extraction of full complex events, as well as for direct pairwise relations, which we use for DDI extraction.

The DDI corpus is provided in two formats, in a MetaMap (MTMX) [2] XML format, and a unified Protein-Protein Interaction XML format [10]. The Turku Event Extraction System uses the latter format as its native data representation, making it a suitable system for adapting to the current task.

In this work we test several feature representations applicable for DDI extraction. We test two different classification methods, and demonstrate the importance of thorough parameter optimization for obtaining optimal performance on the DDI Shared Task.

2 Methods

2.1 System Overview

The Turku Event Extraction System abstracts event and relation extraction by using an extendable graph format. The system extracts information in two main steps: detection of trigger words (nodes) denoting entities in the text, and detection of their relationships (edges). Additional processing steps can e.g. refine the resulting graph structure or convert it to other formats. In the DDI Extraction 2011 task all entities, the drug mentions, are given for both training and test data. Thus, we only use the *edge detector* part of the Turku Event Extraction System. Each undirected drug entity pair in a sentence is a drug-drug interaction candidate, marked as a positive or negative example by the annotation. In the graph format, the drug entities are the nodes, and all of their pairs, connected through the dependency parse, are the edge examples to be classified (See Figure 1).

We adapt the Turku Event Extraction System to the DDI task by extending it with a new example builder module, which converts the DDI corpus into machine learning classification examples, taking into account information specific for drug-drug interactions.

2.2 Data Preparation

The DDI corpus provided for the shared task was divided into a training corpus of 4267 sentences for system development, and a test corpus of 1539 sentences

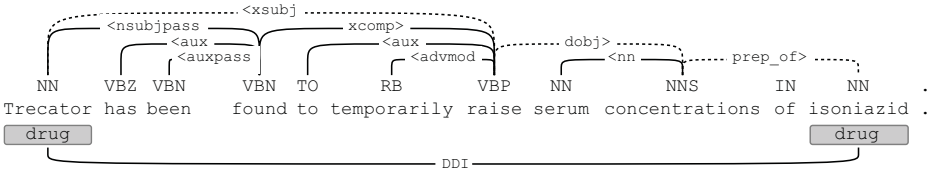


Fig. 1. A potential drug-drug interaction (DDI) can exist for each pair of drug entities in the sentence. This example sentence from the DrugDDI corpus training set has one positive interaction. The automatically generated syntactic deep dependency parse is shown above the sentence. Our system classifies drug entity pairs based on features built primarily from the *shortest path of dependencies*, shown with the dotted line.

that was only made available without labels for running the final results (labels are available now that the task has ended). To develop a machine learning based system, one needs to optimize the parameters of the learned model, by testing experimentally a number of values on data not used for learning. We therefore divided the training corpus into a 3297 sentence *learning set* and a 970 sentence parameter *optimization set* in roughly a ratio of 3:1.

The training corpus was provided in a set of 436 files, each containing sentences relevant to a specific drug. We put all sentences from the same file into either the learning or the optimization set, to prevent the classifier relying too much on specific drug names. The number of negative pairs varied from 1 to 87 and positive pairs from 0 to 29 per file. To maintain a balanced class distribution, and to ensure a representative sample of interactions in both the learning and optimization sets, we divided the files by first sorting by positive interactions, then distributing the files every 3 to learning set and 1 to optimization set. This division distributed the positive interactions almost exactly 3:1 (1156 vs. 374) between the learning and optimization sets. The positive/negative class ratio for the learning set was 54% (1156/2138) vs. 63% (374/596) for the optimization set, a difference we considered acceptable for optimizing the class ratio dependent F-score.

2.3 Parsing

Before we could build the machine learning examples, all sentences needed to be processed with deep syntactic parsing, using the tool chain implemented for the Turku Event Extraction System in previous work [1]. This tool chain performs parsing in two steps: the Charniak-Johnson parser [3] first generates a PENN parse tree, which is converted to a dependency parse with the Stanford parser

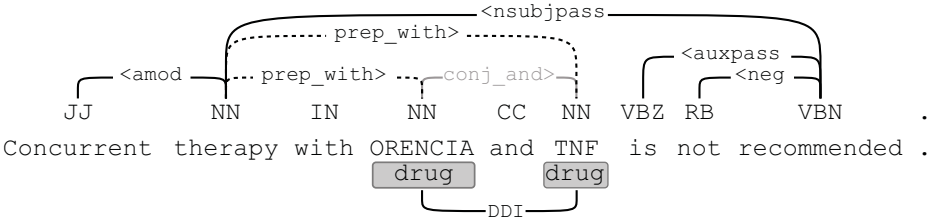


Fig. 2. Skipping the conj_and dependencies when determining the shortest path (dotted line) allows more tokens relevant for the potential interaction to be included in the path.

tools [7]. A dependency parse represents syntax in a form useful for semantic information extraction [8]. With the Charniak-Johnson parser, we used David McClosky's domain-adapted biomodel trained on the biomedical GENIA corpus and unlabeled PubMed articles [6].

2.4 Feature Representations

We use a component derived from the event argument detector of the Turku Event Extraction System. This module is designed to detect relations between two known entities in text, which in this task are the drug-drug pairs. We use the module in the undirected mode, since the drug-drug interactions do not have a defined direction in the current task. Our basic feature representation is the one produced by this system, comprised of e.g. token and dependency n-grams built from the shortest path of dependencies (See Figure 1), path terminal token attributes and sentence word count features. The token and dependency types, POS tags and text, also stemmed with the Porter stemmer [9], are used in different combinations to build variations of these features.

As a modification of the Turku Event Extraction System event argument detector we remove *conj_and* type dependencies from the calculation of the shortest path. The event argument edges that the system was developed to detect usually link a protein name to a defined interaction trigger word (such as the verb defining the interaction). In the case of DDIs, such words are not part of the annotation, but can still be important for classification. Dependencies of type *conj_and* can lead to a shortest path that directly connects a drug entity pair, without travelling through other words important for the interaction (See Figure 2). Skipping *conj_and* dependencies increased the F-score on the optimization set by 0.42 percentage points.

We further improve extraction performance by using external datasets containing information about the drug-drug pairs in the text. DrugBank [16], the

database on which the DrugDDI corpus is based, contains manually curated information on known drug-drug interaction pairs. We mark as a feature for each candidate pair whether it is present in DrugBank, and whether it is there as a known interacting pair.

We also use the data from the MetaMap (MTMX) [2] version of the DDI corpus. For both entities in a candidate pair, we add as MetaMap features their CUI numbers, predicted long and short names, prediction probabilities and semantic types. We also mark whether an annotated drug name has not been given a known name by MetaMap, and whether both entities have received the same name. We normalize the prediction probabilities into the range [0,1] and sort them as the minimum and maximum MetaMap probability for the candidate pair. For the semantic types, we build a feature for each type of both entities, as well as each combination of the entities' types.

2.5 Classification

We tested two similar classifier training methods, the (soft margin) support vector machine (SVM) [15] and the regularized least-squares (RLS) [11]. Both of the methods are *regularized kernel methods*, and are known to be closely related both theoretically and in terms of expected classification performance [4, 11].

Given a set of m training examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $\mathbf{x}_i$ are n-dimensional feature vectors and y_1 are class labels, both methods under consideration can be formulated as the following regularized risk minimization problem [4]:

$$\mathbf{w}^* = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \sum_{i=1}^{m} l\left(\mathbf{x}_i^{\mathrm{T}}\mathbf{w}, y_i\right) + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w} \right\}, \tag{1}$$

where the first term measures with a loss function l how well $\mathbf{w}$ fits the training data, the second term is the quadratic regularizer that measures the learner complexity, and $\lambda > 0$ is a regularization parameter controlling the trade-off between the two terms. In standard SVM formulations, the regularization parameter is often replaced with its inverse $C = \frac{1}{\lambda}$. The hinge loss, defined as

$$l(f, y) = \max\left(1 - fy, 0\right), \tag{2}$$

leads to the SVM and the squared loss, defined as

$$l(f, y) = (f - y)^2, \tag{3}$$

to the RLS classifiers [11], when inserted into equation (1).

Natural language based feature representations are typically characterized by high dimensionality, where the number of possible features may correspond to the size of some vocabulary, or to some power of such number. Further, the data is typically sparse, meaning that most of the features are zero valued. Linear models are typically sufficiently expressive in such high dimensions. Further, efficient algorithms that can make use of the sparsity of the data, so that their computational and memory costs scale linearly with respect to the number of non-zero features in the training set, are known for both SVM [5] and RLS [11]. For these reasons, we chose to train the models using the linear kernel.

3 Results

In the experiments the optimization set was used for learner parameter selection. The final models were trained on all training data, using the learner parameters that resulted in best performance on the optimization set. For both SVM and RLS, the regularization parameter value was chosen using grid search on an exponential grid. The RLS experiments were run using our RLScore open source software[4], whereas the SVM experiments were implemented with the Joachims SVM^{multiclass} program[5] [14].

Both RLS and SVM models produce real-valued predictions. Typically, one assigns a data point to the positive class if the prediction is larger than zero, and to the negative if it is smaller than zero. Since the learning methods are based on optimizing an approximation of classification error rate, the learned models may not be optimal in terms of F-score performance. For this reason, we tested re-calibrating the learned RLS model. We set the threshold at which negative class predictions change to positive to the point on the precision-recall curve that lead to the highest F-score on the development set. The threshold was set to a negative value, indicating that the re-calibration trades precision in order to gain recall. Due to time constraints the same approach was tested with SVMs only after the final DDI Extraction 2011 task results had been submitted.

The RLS results of 62.99% F-score are clearly higher than any of the submitted SVM results. This is mostly due to the re-calibration of the RLS model, which leads to higher recall with some loss of precision, but overall better F-score. A corresponding experiment with an SVM, performed after the competition, confirms that this threshold optimization is largely independent of the classifier used (See Table 3), although the RLS still has a slightly higher performance. With 755 positives and 6271 negatives in the test set, the all-positive F-score for the test set is 19.41%, a baseline above which all of our results clearly are.

Adding features based on information from external databases clearly improves performance. Using known DrugBank interaction pairs increases performance by 0.94 percentage points and adding the MetaMap annotation a further 0.99 percentage points, a total improvement of 1.93 percentage points over result number 1 which uses only information extracted from the corpus text.

4 Discussion and Conclusions

The results demonstrate that combining rich feature representations with state-of-the art classifiers such as RLS or SVM provides a straightforward approach to automatically constructing drug-drug interaction extraction systems. The high impact of the threshold optimization on both RLS and SVM results outlines the importance of finding the optimal trade-off between precision and recall. The RLS slightly outperforms SVM in our experiments, resulting in our final DDI

[4] available at www.tucs.fi/rlscore

[5] http://svmlight.joachims.org/svm_multiclass.html

Table 1. DDI Extraction 2011 results. This table shows the extraction performance for the four *results* (1-4) submitted for the shared task, as well as a post-competition experiment (pce). The *features* are the baseline features, built only from the DDI corpus, features built from known DrugBank interaction pairs, and features based on the provided MetaMap annotation. For classification, either an SVM or an RLS *classifier* was used, potentially with an optimal *threshold* for parameter selection.

Result	Features	Classifier	Threshold	Precision	Recall	F-score
1	corpus	SVM	-	67.05	46.36	54.82
2	corpus+DrugBank	SVM	-	65.13	48.74	55.76
pce	corpus+DrugBank	SVM	+	62.53	62.12	62.33
3	corpus+DrugBank	RLS	+	58.04	68.87	62.99
4	corpus+DrugBank+MetaMap	SVM	-	67.40	49.01	56.75

Extraction 2001 task F-score of 62.99%. Using also MetaMap features with the RLS classifier setup might further improve performance.

Our results indicate that using additional sources of information, such as the DrugBank and the MetaMap can lead to gains in predictive performance. In the DDI Extraction 2011 task using any external databases was encouraged to maximise performance, but when applying such methods to practical text mining applications care must be exercised. In particular, using lists of known interactions can increase performance on well known test data, but could also cause a classifier to rely too much on this information, making it more difficult to detect the new, unknown interactions. Fortunately, while external databases increase performance, their contribution is a rather small part of the whole system performance, and as such can be left out in situations that demand it.

At the time of writing this paper, the other teams' results in the DDI Shared Task are not available, so we can't draw many conclusions from our performance. The F-score of 62.99% is clearly above the all-positive baseline of 19.41%, indicating that the basic machine learning model is suitable for this task. The performance is somewhat similar to Turku Event Extraction System results for comparable relation extraction tasks in the BioNLP'11 Shared Task, such as the Bacteria Gene Interactions (BI) task F-score of 77% and the Bacteria Gene Renaming (REN) task text-only features F-score of 67.85% [1].

For the DDI corpus, to the best of our knowledge, the only available point of comparison is the task authors' F-score of 60.01% using a shallow linguistic kernel [13]. For the DDI Extraction 2011 task the corpus has been somewhat updated and the training and test set division seems slightly different. Even if these results are not directly comparable, we can presume our result to be in roughly the same performance range.

We have extended the Turku Event Extraction System for the task of DDI extraction, and have developed optimized feature and machine learning models for achieving good performance. We hope our work can contribute to further developments in the field of DDI extraction, and will publish our software for download from `bionlp.utu.fi` under an open source license.

References

1. Björne, J., Salakoski, T.: Generalizing biomedical event extraction. In: Proceedings of BioNLP Shared Task 2011 Workshop. pp. 183–191. Association for Computational Linguistics, Portland, Oregon, USA (June 2011)
2. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research 32(suppl 1), D267–D270 (2004)
3. Charniak, E., Johnson, M.: Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). pp. 173–180. ACL (2005)
4. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. Advances in Computational Mathematics 13, 1–50 (April 2000)
5. Joachims, T.: Training linear SVMs in linear time. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.) Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2006). pp. 217–226. ACM Press, New York, NY, USA (2006)
6. McClosky, D.: Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Ph.D. thesis, Department of Computer Science, Brown University (2010)
7. de Marneffe, M.C., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC-06. pp. 449–454 (2006)
8. de Marneffe, M.C., Manning, C.: The Stanford typed dependencies representation. In: COLING Workshop on Cross-framework and Cross-domain Parser Evaluation (2008)
9. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
10. Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., Salakoski, T.: Comparative Analysis of Five Protein-protein Interaction Corpora. BMC Bioinformatics, special issue 9(Suppl 3), S6 (2008)
11. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In: Suykens, J., Horvath, G., Basu, S., Micchelli, C., Vandewalle, J. (eds.) Advances in Learning Theory: Methods, Model and Applications, NATO Science Series III: Computer and System Sciences, vol. 190, chap. 7, pp. 131–154. IOS Press, Amsterdam, Netherlands (2003)
12. Sánchez-Cisneros, D., Segura-Bedmar, I., Martínez, P.: DDIExtractor: A Web-Based Java Tool for Extracting Drug-Drug Interactions from Biomedical Texts. In: Muñoz, R., Montoyo, A., Métais, E. (eds.) Natural Language Processing and Information Systems, Lecture Notes in Computer Science, vol. 6716, pp. 274–277. Springer Berlin / Heidelberg (2011)
13. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchez, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics In Press, Corrected Proof, – (2011)
14. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research (JMLR) 6(Sep), 1453–1484 (2005)
15. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)
16. Wishart, D.S., Knox, C., Guo, A.C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., Hassanali, M.: DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Research 36(suppl 1), D901–D906 (2008), http://nar.oxfordjournals.org/content/36/suppl$_1$/D901.*abstract*

Feature Selection for Drug-Drug Interaction Detection Using Machine-Learning Based Approaches

Anne-Lyse Minard[1,2], Lamia Makour[1],
Anne-Laure Ligozat[1,3], and Brigitte Grau[1,3]

[1] LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France
[2] Université Paris-Sud 11, Orsay, France
[3] ENSIIE, Évry, France
firstname.lastname@limsi.fr

Abstract. This paper describes the systems developed for the DDI Extraction challenge. The systems use machine learning methods and are based on SVM by using LIBSVM and SVMPerf tools. Classical features and corpus-specific features are used, and they are selected according to their F-score. The best system obtained an F-measure of 0.5965.

Keywords: relation extraction, machine-learning methods, feature selection, drug-drug interaction, LIBSVM, SVMPerf

1 Introduction

In this paper [4], we present our participation to DDI Extraction challenge. The task was to detect if two drugs in the same sentence are in interaction or not. For example in (1) there is an interaction between *HUMORSOL* and *succinylcholine*, and between *HUMORSOL* and *anticholinesterase agents*, but not between *succinylcholine* and *anticholinesterase agents*.

(1) Possible drug interactions of HUMORSOL with succinylcholine or with other anticholinesterase agents.

The high number of features relevant to recognize the presence of an interaction between drugs in sentence, conducts us to propose systems based on machine-learning methods. We chose classifiers based on SVM because they are used in state-of-art systems for relation extraction. We tested two classifiers: LIBSVM [Chang and Lin2001] and SVMPerf [Joachims2005]. We thought that SVMPerf could improve the classification of the not well represented class, i.e. the interaction class (only 10% of drugs pairs are in interaction), because it gives more tolerance of false positives for the under-represented class. We also worked on feature selection in order to keep the most relevant features. In a first section,

[4] This work has been partially supported by OSEO under the Quaero program.

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 43–50, Huelva, Spain, September 2011.

we briefly describe the corpus and the knowledge it enables us to compute based on recurrent relations between same drugs. Then we describe our solution that makes use of LIBSVM and the studies we have done concerning first feature selection to improve the classification made by LIBSVM and second the use of another classifier SVMPerf. We then show the results obtained by our systems.

2 Corpus

2.1 Description

For the challenge we disposed of two corpora composed of biomedical texts collected from the DrugBank database and annotated with drugs [Segura-Bedmar et al.2011]. The development corpus was annotated with drug-drug interactions, and the evaluation corpus was annotated with drugs. We chose to use the corpora in the Unified format. The development corpus is composed of 435 files, which contain 23,827 candidate pairs of drugs including 2,402 drug-drug interactions. The evaluation corpus contains 144 files and 7,026 candidate pairs containing 755 interactions. We split the development corpus into training (1,606 interactions) and test (796 interactions) sub-corpora for the development of our models.

2.2 Knowledge Extracted from the Corpus

For each pair of entities in the development corpus, we searched if this pair is often found in interaction or never in interaction in the corpus. The results of this study are shown in table 1. Between brackets, we indicate the number of pairs that appear at least twice. For example, there are 91 pairs of drugs that always interact and appear more than twice in the corpus.

Table 1. Number of pairs in the development corpus

	training corpus
# entities couple	14,096
# never interact	12,163 (2,706)
# always interact	1,047 (91)
# interact and not	886

These results are kept in a knowledge base that will be combined with the results of the machine-learning method (see 5.1). We can see that the most relevant information coming from this kind of knowledge concerns the absence of interaction.

3 Classification with LIBSVM

We first applied LIBSVM with the features described in [Minard et al.2011] for the i2b2 2010 task about relation extraction. We wanted to verify their relevance

for this task. The system we developed use classical features ([Zhou et al.2005], [Roberts et al.2008]). We added to them some features related to the writing style of the corpus and some domain knowledge. For each pair of drugs all the features are extracted. If there are four drugs in the same sentence, we considered six pairs of drugs. In this section, we describe the sets of features and the classifier.

3.1 Features

We first defined a lot of features, and then with the training and test corpus we did several tests and we kept only the most relevant combination of features for this task. In this section we described the features kept for the detection of interaction.

3.1.1 Coordination

To reduce the complexity of sentences we processed sentences before feature extraction to delete entities (tagged as drug) in coordination with one of the two candidate drugs. We added three features: the number of deleted entities, the coordination words that are the triggers of the deletion (*or*, *and*, a comma), and a feature which indicates that the sentence was reduced. This reduction is applied on 33% pairs of drugs in the training corpus.

3.1.2 Surface Features

The surface features take into account the position of the two drugs in the sentence.

- **Distance** (i.e. number of words [5]) between the two drugs: in the development corpus 88% of drugs in interaction are separated by 1 to 20 words. The value of this feature is a number, and not one or zero like other features.
- **Presence of other concepts** between the two entities: for 82% of the entity pairs in relation in the development corpus there are no other drugs between them.

3.1.3 Lexical Features

The lexical features are composed by the words of the contexts of the two entities, including verbs and prepositions which often express interaction.

- **The words and stems** [6] which constitute the entities. The stems are used to group inflectional and derivational variations altogether.
- **The stems of the three words** at the left and right contexts of candidate entities. After several tests we chose a window of three words; with bigger or smaller windows, precision lightly increases but recall decreases.

[5] The words include also the punctuation signs.

[6] We use the PERL module lingua::stem to obtain the stem of the word: http://snowhare.com/utilities/modules/lingua-stem/.

- **The stems of the words** between candidate concepts, to consider all the words between concepts; the most important information for the classification is located here.
- **The stems of the verbs** in the three words at the left and right of candidate concepts and between them. The verb is often the trigger of the relation: for example in (2) the interaction is expressed by *interact*.

  (2) Beta-adrenergic blocking agents may also **interact** with sympathomimetics.
- **The prepositions** between candidate concepts, for example *with* in (3).

  (3) d-amphetamine **with** desipramine or protriptyline and possibly other tricyclics cause striking and sustained increases in the concentration of d-amphetamine in the brain;

3.1.4 Morpho-Syntactic Features

This features take into account syntactic information for expressing relations.

- **Morpho-syntactic tags** of the three words at the left and right of candidate entities: the tags come from the TreeTagger [Schmid1994].
- **Presence of a preposition** between the two entities, regardless of which preposition it is.
- **Presence of a punctuation sign** between candidate entities, if it is the only "word".
- **Path length in the constituency tree** between the two entities: the constituency trees are produced by the Charniak/McClosky parser [McClosky2010].
- **Lowest common ancestor** of the two entities in the constituency tree.

Figure 1 represents the constituency tree for example (2). The length of the path between *Beta-adrenergic blocking agents* and *sympathomimetics* is 9 and the common ancestor is *S*.

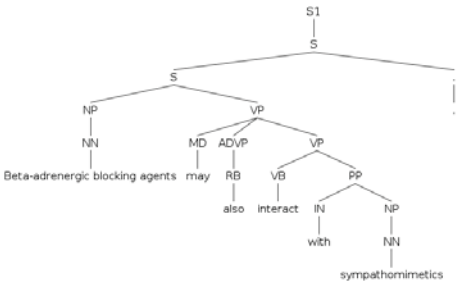


Fig. 1. Example of a constituency tree

3.1.5 Semantic Features

In order to generalize information given by some terms, we also give to the classifier their semantic types.

- **Semantic type (from the UMLS)** of the two entities. In the example (2) the entity *sympathomimetics* has the semantic type *pharmacologic substance*.
- **VerbNet classes** [7] (an expansion of Levin's classes) of the verbs in the three words at the left and right of candidate concepts and between them. For example *increase* is member of the same class as *enhance*, *improve*, etc.
- **Relation between the two drugs in the UMLS**: in the development corpus 57 kinds of relation are found. There is a relation in the UMLS for 5% of drugs pairs in the development corpus. For example, in the UMLS there is a relation *tradename of* between *Procainamide* and *Pronestyl* (4), so the two entities cannot be in interaction.
  (4) - Procainamide (e.g., Pronestyl) or

3.1.6 Corpus-Specific Features

These kinds of features are specific to the DDI corpus.

- A feature indicates **if one of the two drugs is the most frequent drug in the file**. Each file is about one particular drug, so most of the interaction described in the file is between it and another drug.
  A lot of sentences begin with a drug and a semi-colon, like sentence (5). A feature encodes **if one of the two drugs is the same as the first drug** in the sentence.
  (5) **Valproate**: Tiagabine causes a slight decrease (about 10%) in steady-state **valproate** concentrations.
- A feature is set **if one of the two entities is refered to by the term "drug"**: in the training corpus 520 entities are "drug". In this case the expression of the relation can be different (6).
  (6) Interactions between Betaseron and other **drugs** have not been fully evaluated.

3.2 Classifier

We used the LIBSVM tool with a RBF kernel. c and gamma parameters were chosen by the tool *grid.py* with the train corpus for test: c was set at 2 and gamma at 0.0078125. For each class we determined a weight on the parameter c to force the system to classify in the class of interaction. We did tests to choose the value of the weight: for the class of non-interaction the weight is 2 and for the interaction class the weight is 9.

4 Studies from LIBSVM results

This first system obtained 0.56 F-measure on the test corpus. We then made studies on two axes. As the number of features is great, we studied how to reduce it in order to improve the classification. We also studied the application of another classifier which could give more tolerance to false positive to improve the performance of prediction with unbalanced data.

[7] http://verbs.colorado.edu/∼mpalmer/projects/verbnet.html

4.1 Feature Selection

We did a selection of features thanks to the F-score of each feature computed as in [Chen and Lin2006] on the training corpus, prior to the training of the classifier. Given a data set X with m classes, X^k the set of instances in class k, and $|X^k| = l_k, k = 1, ..., m$. Assume $\bar{x}_j^k$ and $\bar{x}_j$ are the average of the j th feature in X^k and X, respectively. The Fisher score of the jth feature of this data set is defined as:

$$\hat{F}(j) = \frac{S_B(j)}{S_W(j)},$$

where

$$S_B(j) = \sum_{k=1}^{m} l_k(\bar{x}_j^k - \bar{x}_j)^2, S_W(j) = \sum_{k=1}^{m} \sum_{x \in X^k} (x_j - \bar{x}_j^k)^2$$

We used the tool *fselect.py*, provided with the LIBSVM library. We defined different thresholds under which we deleted the features. We classified the features in four classes: the semantic class, the morpho-syntactic class, the lexical class and a class with the other features (syntactic, surface, corpus-specific and coordination features). We did tests with different combinations of thresholds for each features class. The best combination of thresholds is described in table 2. This improvement lead to an F-measure of 0.59 on the test corpus. On the full training corpus, we have 368 fewer features after selection, i.e. a total of 9741 features.

Table 2. Best combination of thresholds for feature selection

Semantic class	0.001
Morpho-syntactic class	0.000001
Lexical class	0
Other	0.000004

4.2 SVMPerf

We also tested the SVMPerf tool with a linear kernel. This tool is faster than LIBSVM and optimizes different measures of performance like F1-score or ROC-Area in binary classification. This last measure (ROCArea) allows to choose between different training models. The model is optimal if ROCArea=1, which is the probability to affect the right class to each instance. After training, we changed the value of the threshold b from 1.5 to 1.2. This value was the optimal threshold between the different values that we tested; it increases the performance of prediction with more tolerance of false positives. The c parameter was set at 20 after test of several values with the training corpus.

5 Experimentations and Results

In this section we describe the particularity of each developed system, and finally we give the results obtained at DDI Extraction 2011.

5.1 Experimentations

1. LIMSI-CNRS_4: LIBSVM (baseline)
   This system is the baseline described in 3.
2. LIMSI-CNRS_2: LIBSVM + feature selection
   This system uses lIBSVM with feature selection.
3. LIMSI-CNRS_3: LIBSVM + feature selection (bis)
   This system is the same as the previous one, but the c and gamma parameters differ. The parameters are calculated on the development corpus. The c parameter was set at 2048 and the gamma parameter at 0.0001220703125.
4. LIMSI-CNRS_1: LIBSVM + feature selection + knowledge
   This system is based on LIBSVM. After the classification we combined the prediction of the classifier and the knowledge (cf. section 2.2) in case that their decisions differ. The combination is done as follows: for the class of non-interaction, if the couple exists in the knowledge base and the decision value provided by the classifier is lower than 0.1, the resulting class is the class of the knowledge base. For the interaction class, we keep the class of the knowledge base when the classifier decision value is lower than -0.5.
5. LIMSI-CNRS_5: LIBSVM + SVMPerf (+ feature selection)
   We combine the performance of SVMPerf and LIBSVM by comparing the decision values from each tool. If the two decision values are lower than 0.5, we use the LIBSVM prediction, otherwise we use the prediction with the highest decision value.

5.2 Results and Discussion

The results of the different runs are presented in table 3. The best F-measure is 0.5965 and was obtained by the system which used LIBSVM and combined the prediction of the classfier with the knowledge about pairs of drugs in the training corpus. This F-measure is not significantly different with the F-measure obtained by the system which used LIBSVM without using the knowledge about pairs of drugs in the corpus. So the use of information about the presence or not of the pairs of drugs in the training corpus is not useful for the identification of drugs interaction because the intersection of drugs pairs in the development and evaluation corpus is small (cf. Table 4). There are only 15 pairs that are always in interaction in the development corpus and the evaluation corpus. The best improvement is given by feature selection: without feature selection the system obtained an F-measure of 0.57 and with feature selection of 0.59. However, we can notice that the combination of the two classifiers improve precision.

6 Conclusion

For the DDI Extraction challenge, we developed several methods based on SVM. We showed that a selection of features according to their F-measure improve interaction detection. Reducing the number of features leads to a 0.02 increase of the F-measure. We also showed that SVMPerf is not as efficient as libSVM for this task on this kind of unbalanced data.

Table 3. Results

	Precision	**Recall**	**F-measure**
LIBSVM (baseline)	0.5487	0.6119	0.5786
LIBSVM + feature selection	0.5498	**0.6503**	0.5959
LIBSVM + feature selection (bis)	0.4522	0.5139	0.4811
LIBSVM + feature selection + knowledge	0.5518	0.6490	**0.5965**
LIBSVM (+ feature selection) and SVMPerf	**0.5856**	0.4940	0.5359

Table 4. Intersection between the pairs in the development and the evaluation corpus

		development corpus			
		# never interact	# always interact	# not in development corpus	total
evaluation corpus	# never interact	1,323	100	2,929	4,352
	# always interact	25	15	329	369
	# not in evaluation corpus	10,772	1,008		
	total	12,120	1,123		

References

[Chang and Lin2001] Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines.* Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[Chen and Lin2006] Y. W. Chen and C. J. Lin, 2006. *Combining SVMs with various feature selection strategies.* Springer.

[Joachims2005] Thorsten Joachims. 2005. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 377–384, New York, NY, USA. ACM.

[McClosky2010] David McClosky. 2010. Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. *PHD Thesis, Department of Computer Science, Brown University.*

[Minard et al.2011] Anne-Lyse Minard, Anne-Laure Ligozat, Asma Ben Abacha, Delphine Bernhard, Bruno Cartoni, Louise Delger, Brigitte Grau, Sophie Rosset, Pierre Zweigenbaum, and Cyril Grouin. 2011. Hybrid methods for improving information access in clinical documents: Concept, assertion, and relation identification.

[Roberts et al.2008] Angus Roberts, Robert Gaizauskas, and Mark Hepple. 2008. Extracting clinical relationships from patient narratives. In *BioNLP2008: Current Trends in Biomedical Natural Language Processing*, pages 10–18.

[Schmid1994] Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

[Segura-Bedmar et al.2011] Isabel Segura-Bedmar, Paloma Martinez, and Cesar de Pablo-Sanchez. 2011. Using a shallow linguistic kernel for drug-drug interaction extraction. *Journal of Biomedical Informatics*, In Press, Corrected Proof:–.

[Zhou et al.2005] GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 427–434.

Automatic Drug-Drug Interaction Detection: A Machine Learning Approach With Maximal Frequent Sequence Extraction

Sandra Garcia-Blasco[1], Santiago M. Mola-Velasco[1], Roxana Danger[2], and Paolo Rosso[3]

[1] bitsnbrains S.L. and Universidad Politécnica de Valencia, Spain – {*sandra.garcia,santiago.mola*}*@bitsnbrains.net*
[2] Imperial College London, UK – *rdanger@imperial.ac.uk*
[3] NLE Lab. - ELiRF, DSIC, Universidad Politécnica de Valencia, Spain – *prosso@dsic.upv.es*

Abstract. A Drug-Drug Interaction (DDI) occurs when the effects of a drug are modified by the presence of other drugs. DDIExtraction2011 proposes a first challenge task, Drug-Drug Interaction Extraction, to compare different techniques for DDI extraction and to set a benchmark that will enable future systems to be tested. The goal of the competition is for every pair of drugs in a sentence, decide whether an interaction is being described or not. We built a system based on machine learning based on bag of words and pattern extraction. Bag of words and other drug-level and character-level have been proven to have a high discriminative power for detecting DDI, while pattern extraction provided a moderated improvement indicating a good line for further research.

1 Introduction

A Drug-Drug Interaction (DDI) occurs when the effects of a drug are modified by the presence of other drugs. The consequences of a DDI may be very harmful for the patient's health, therefore it is very important that health-care professionals keep their databases up-to-date with respect to new DDI reported in the literature.

DDIExtraction2011 proposes a first challenge task, DDI Extraction, to compare different techniques for DDI extraction and to set a benchmark that will enable future systems to be tested. The goal of the competition is for every pair of drugs in a sentence, decide whether an interaction is being described or not. The corpus used was the DrugDDI corpus [1]. Two formats of the corpus were provided, MMTx format and Unified format. Our system uses Unified format, which only contains labels for drugs. Table 1 shows the corpus statistics[4].

The paper is structured as follows: Section 2 overviews related work. Section 3 describes the system used as well as its features. In section 4 we discuss the evaluation and results and in Section 5 we draw some conclusions.

[4] These statistics cover only documents and sentences that contain, at least, one drug pair.

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 51–58, Huelva, Spain, September 2011.

Table 1. DrugDDI corpus statistics.

	Training	Test	Total
Documents	399	134	533
Sentences	2812	965	3777
Pairs of drugs	23827	7026	30853
Interactions	2397	755	3152

2 Related Work

Even though the problem of DDI extraction is relatively new, some authors have already presented approximations to solve it. In [2], the author presents two approximations to face the problem: a hybrid approach, combining shallow parsing and matching of patterns described by a pharmacist; and an approximation based on kernel methods that obtained better results that the hybrid approach, reaching 55% precision and 84% recall.

In [3] the authors propose a first approximation for DDI detection based on automatically determining the patterns that identify DDI from a training set. The patterns extracted were *Maximal Frequent Sequences* (MFS), based on [4]. In this work, the identified MFS were used to determine whether a sentence contains or not a description of a DDI, without identifying the pair of interacting drugs. MFS have been useful in different tasks such as text summarization [5], measuring text similarities [6] and authorship attribution [7]. MFS will also be part of our approximation, and will be defined further on.

Protein-Protein Interaction (PPI) extraction is an area of research very similar to DDI extraction that has received a bigger attention from the scientific comunity. The BioCreative III Workshop hosted two tasks of PPI document classification and interaction extraction [8]. Some of the features present in a wide range of participants were bag-of-words, bigrams, co-occurrences and character ngrams. This kind of features will have a key role in our system. In [9] the authors use patterns as one of their main features to extract PPI. In [10], the authors use a hybrid approach with clustering and machine learning classification using Support Vector Machines (SVM).

3 Our System

We built a system based on machine learning[5], therefore we had to define a feature set to estimate the model. Each sample is one possible interaction, this is, each unique combination of two drugs appearing in a sentence of the corpus. Given the small size of the corpus and the difficulty of properly estimating the model, it was necessary to represent the features in a reduced space.

The first step was to preprocess the corpus. For doing so, each sentence was tokenized[6] with standard English tokenization rules (e.g. split by spaces, removal

[5] We used RapidMiner for every classification and clustering model. Available at `http://rapid-i.com/`.

[6] The tokenization was performed with Apache Lucene. Available at `http://lucene.apache.org`.

of apostrophes, conversion to lower case, removal of punctuation marks) with the following particularities:

- Each group of tokens that represent a drug were replaced by *#drug#*.
- Numbers were replaced by *_num_*.
- Stop words were not removed.
- Stemming was applied[7].
- Percentage symbols were preserved as independent tokens.

In the following subsections, we will describe the different features used in the system.

3.1 Bag of Words

From the set of all words appearing in the preprocessed corpus, we discarded those with a frequency lower than 3 and stop words. With the resulting set of words, we generated a dataset where each sample was a possible interaction in the corpus and each feature was the presence or not of each word between the two drugs of the potential interaction. Using this dataset, every word was ranked using information gain ratio with respect to the label [8]. Then, every word with an information gain ratio lower than 0.0001 was discarded. The presence of each of the remaining words was a feature in the final dataset. Finally, 1,010 words were kept.

Samples of words with a high gain ratio are: *exceed*, *add*, *solubl*, *amphetamin*, *below*, *lowest*, *second*, *defici*, *occurr*, *stimul* and *acceler*.

3.2 Word Categories

In biomedical literature complex sentences are used very frequently. MFS and bag of words are not able to capture relations that are far apart inside a sentence. To somehow reflect the structure of the sentence, we defined some word categories. This way, we can have some information about dependent and independent clauses, coordinate and subordinate structures, etc. Some of these categories were also included in [2]. We added two categories that include absolute terms and quantifiers, as well as a category for negations. Table 2 enumerates the words included in each category.

For each word category we defined two features. One indicating how many times the words in the category appeared in the sentence, and the other indicating how many times they appeared between the two drugs of the potential interaction.

[7] The stemming algorithm used was Snowball for English. Available at `http://snowball.tartarus.org`.

[8] Information Gain Ratio was calculated using Weka. Available at `http://www.cs.waikato.ac.nz/ml/weka/`.

Table 2. Word Categories.

Category	Words included
Subordinate	*after, although, as, because, before, if, since, though, unless, until, whatever, when, whenever, whether, while.*
Independent markers	*however, moreover, furthermore, consequently, nevertheless, therefore.*
Appositions	*like, including, e.g., i.e.*
Coordinators	*for, and, nor, but, or, yet, so.*
Absolute	*never, always.*
Quantifiers	*higher, lower.*
Negations	*no, not.*

3.3 Maximal Frequent Sequences

Similar to bag of words, we used sequences of words as features. For this, we used Maximal Frequent Sequences (MFS).[9] Following [4], a sequence is defined as an ordered list of elements, in this case, words. A *sequence* is *maximal* if it is not a subsequence of any other, this is, if it does not appear in any other sequence in the same order. Given a collection of sentences, a *sequence* is β*-frequent* if it appears in at least β sentences, where β is the defined frequency threshold. *MFS* are all the sequences that are β*-frequent* and maximal.

We extracted all the MFS from the training corpus, with a β of 10 minimum length of 2. Given the size of the corpus, sometimes very long MFS have no capability to generalize knowledge because they sometimes represent full sentences, instead of patterns that should be frequent in a kind of sentence. To avoid this, we restricted the MFS to a maximum length of 7 words. With this, we obtained 1.010 patterns. In order to reduce the feature space we calculated clusters of MFS.

Clusters were calculated with the Kernel K-Means algorithm [11], using radial kernel, with respect to the relative frequency of each MFS in the following contexts: a) sentences, b) sentences containing an interaction, c) MFS appearing between two drugs, c) MFS appearing before the first drug of an interaction and d) MFS appearing after the last drug of an interaction. Clustering helped to avoid pattern redundancy. This was necessary because some patterns could be considered equivalent since they only differed in one or a few words not relevant in the context of DDI. We obtained 274 clusters. Each of this clusters is a feature of the final dataset which is set to 1 if, at least, one of the MFS of the cluster matches with the potential interaction. The matching algorithm is shown in Algorithm 1.

3.4 Token and Char Level Features

At the token and char level, several features were defined. We must recall that, during preprocessing, every token or group of tokens labeled as drugs where replaced by the token *#drug#*. Table 3 describes this subset of features. Each one of these features appears twice in the final dataset, once computed on the

[9] We used a proprietary library by bitsnbrains, `http://bitsnbrains.net`.

Algorithm 1: MFS matching algorithm.

Input: mfs, $sentence$, $drug1index$, $drug2index$
Output: $match$
$startThreshold \leftarrow 0$
$endThreshold \leftarrow 0$
if *"#drug#"* $\in mfs$ **then**
    $startThreshold \leftarrow$ First index of "#drug#" in mfs
    $endThreshold \leftarrow length(mfs) -$ last index of "#drug#" in mfs
$startIndex \leftarrow drug1index - startThreshold$
if $startIndex < 0$ **then**
    $startIndex \leftarrow 0$
$endIndex \leftarrow drug2index + endThreshold$
if $endIndex > length(sentence)$ **then**
    $endIndex \leftarrow length(sentence)$
$textBetweenDrugs \leftarrow$ Substring of $sentence$ from index $startIndex$ to $endIndex$
if mfs *is subsequence of* $textBetweenDrugs$ **then**
    $match \leftarrow 1$
else
    $match \leftarrow 0$

whole sentence and once computed only in the text between the two drugs of the potential interaction.

Table 3. Token and char level features.

Feature	Description
Tokens	Number of tokens.
Token *#drug#*	Number of times the *#drug#* token appears.
Chars	Number of chars.
Commas	Number of commas.
Semicolons	Number of semicolons.
Colons	Number of colons.
Percentages	Number of times the character % appears.

3.5 Drug Level Features

With the features defined so far, we have not taken into account the two drugs of the potential interaction. We believe this is important in order to have more information when deciding wether if they interact or not.

For each document, we calculated the *main drug* as the drug after which the document was named, this is, the name of the article of the DrugBank database where the text was extracted from. In the case of scientific articles, the main drug would be calculated as the drug or drug names appearing in the title of the article, if any. Also for each document, we calculated the *most frequent drug* as the token labeled as drug that appeared more times in the document.

We noticed that, sometimes, drugs are referred to using their trade names. To ensure good treatment of drugs in the drug level features, we replaced each trade name with the original drug name[10]. Table 4 describes the drug level features.

Table 4. Drug level features for candidate interactions (CI)

Feature	Description
Main drug	True if one of of the two drugs in the CI is the document name.
Most frequent drug	True if one of the two drugs in the CI is the most frequent drug in the document.
Cross reference	True if, at least, one of the two drugs in the CI is *drug*, *medication* or *medicine*.
Alcohol	True if, at least, one of the two drugs in the CI is *alcohol* or *ethanol*.
Is same drug	True if both drugs in the CI are the same.

3.6 Classification Model

During preliminary research, we explored the performance of a wide range of classification models, notably Support Vector Machines, Decision Trees and multiple ensemble classifiers such as Bagging, MetaCost and Random Forests [12]. Our best choice was Random Forest with 100 iterations and 100 attributes per iteration.

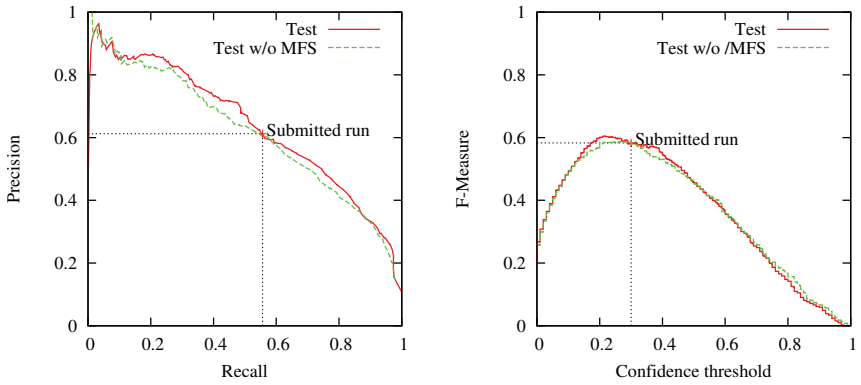
4 Evaluation

We evaluated our model with standard performance measures for binary classification: Precision (P), Recall (R) and F-Measure (F). For each label, our model outputs a confidence value. In order to decide the label, we define a confidence threshold above which the decision will be positive and below which it will be negative. A quick way to visualize every possible set up of the system is the PR curve, where P and R are ploted for different confidence thresholds. Analogously, we can plot F-Measure and confidence thresholds to visualize the optimum threshold with respect F-Measure. AUC-PR is defined as the area under the PR curve. AUC-PR is a very stable measure to compare binary classification models.

We are evaluating the performance of our system for the test set, with and without MFS. Figure 1 shows PR and F curves for both settings. The PR curves are convex, which makes the decision of an optimum threshold much easier and less risky. Table 5 shows Precision, Recall, F-Measure, AUC-PR, precision at recall 0.8 and recall at precision 0.8 for test with MFS.

[10] Trade names were extracted from the KEGG DRUG database, from the Kyoto Encyclopedia of Genes and Genomes. Available at `http://www.genome.jp/kegg/drug/`

MFS improve moderately the performance of the system, increasing about 0.02 in AUC-PR. We expected more influence of MFS. Patterns were extracted using all sentences, even the ones that did not include any drug interaction. We believe that this could have reduced the performance.



(a) Precision-Recall curves (b) F-Measure curves

Fig. 1. PR and F curves for test with and without MFS.

Table 5. Performance measures for test with and without MFS.

	P	R	F	AUC-PR	P@R 0.8	R@P 0.8
Test	0.6122	0.5563	0.5829	0.6341	0.4309	0.3205
Test w/o MFS	0.6069	0.5563	0.5805	0.6142	0.4113	0.2808

5 Conclusions

We presented a system for DDI extraction based on bag-of-words and Maximal Frequent Sequences, as used for the DDIExtraction2011 competition. Our submission obtained a F-Measure of 0.5829 and a AUC-PR of 0.6341 for the test corpus. Our system can be set up to reach recall of 0.3205 with a precision of 0.8, or precision of 0.4309 and a recall 0.8. The use of MFS increased AUC-PR by 0.02.

One of the main problems we have encountered is the complexity of the language structures used in biomedical literature. Most of the sentence contained appositions, coordinators, etc. Therefore it was very difficult to reflect those structures using MFS. The reduced size of the corpus is also a serious limitation for our approach.

Our system should be improved by complementing it with other state-of-the-art techniques used in the PPI field that have not been explored yet during

our participation, such as character n-grams and co-occurrences. It could also be improved by extracting MFS with reduced restrictions and improving the clustering step.

Acknowledgments. This work has been done in the framework of the VLC/ CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems. Contributions of first and second authors have been supported and partially funded by bitsnbrains S.L. Contribution of fourth author has been partially funded by the European Commission as part of the WIQEI IRSES project (grant no. 269180) within the FP 7 Marie Curie People Framework, by MICINN as part of the Text-Enterprise 2.0 project (TIN2009-13391-C04-03) within the Plan I+D+i. Computational resources for this research have been kindly provided by Daniel Kuehn from Data@UrService.

References

1. Segura-Bedmar, I., Martinez, P., de Pablo-Sanchez, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. J. Biomed. Inform. In Press, Corrected Proof, Available online 24 April 2011, DOI 10.1016/j.jbi.2011.04.005.
2. Segura-Bedmar, I.: Application of Information Extraction techniques to pharmacological domain: Extracting drug-drug interactions. PhD thesis, UC3M, Madrid, Spain (April 2010)
3. García-Blasco, S., Danger, R., Rosso, P.: Drug-Drug Interaction Detection: A New Approach Based on Maximal Frequent Sequences. SEPLN **45** (2010) 263–266
4. Ahonen-Myka, H.: Discovery of Frequent Word Sequences in Text. In: Pattern Detection and Discovery. Volume 2447 of LNCS., London, UK, Springer (2002) 180–189
5. García, R.A.: Algoritmos para el descubrimiento de patrones secuenciales maximales. PhD thesis, INAOE, Mexico (September 2007)
6. García-Blasco, S.: Extracción de secuencias maximales de una colección de textos. Final degree project, UPV, Valencia, Spain (December 2009)
7. Coyotl-Morales, R.M., Villaseñor Pineda, L., Montes-y Gómez, M., Rosso, P.: Authorship Attribution using Word Sequences. In: Proc. 11th Iberoamerican Congress on Pattern Recognition, CIARP 2006. Volume 4225 of LNCS., Springer (2006) 844–853
8. Arighi, C., Cohen, K., et al., eds.: Proceedings of BioCreative III Workshop, Bethesda, MD USA (2010)
9. Sullivan, R., Miller, C., Tari, L., Baral, C., Gonzalez, G.: Efficient extraction of protein-protein interactions from full-text articles. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(3) (2010) 481–494
10. Bui, Q.C., Katrenko, S., Sloot, P.M.A.: A hybrid approach to extract proteinprotein interactions. Bioinformatics **27**(2) (2011) 259–265 Code available at http://staff.science.uva.nl/˜bui/PPIs.zip.
11. Zhang, R., Rudnicky, A.I.: A Large Scale Clustering Scheme for Kernel K-Means. In: 16th Conference on Pattern Recognition. Volume 4., Los Alamitos, CA, USA, IEEE Computer Society (2002) 289–292
12. Breiman, L.: Random Forests. Machine Learning **45**(1) (2001) 5–32

A Machine Learning Approach to Extract Drug – Drug Interactions in an Unbalanced Dataset

Jacinto Mata, Ramón Santano, Daniel Blanco,
Marcos Lucero, Manuel J. Maña

Escuela Técnica Superior de Ingeniería. Universidad de Huelva
Ctra. Huelva - Palos de la Frontera s/n. 21819 La Rábida (Huelva)
{jacinto.mata, manuel.mana}@dti.uhu.es,
{ramon.santano, daniel.blanco, marcos.lucero}@alu.uhu.es

Abstract. Drug-Drug Interaction (DDI) extraction from the pharmacological literature is an emergent challenge in the text mining area. In this paper we describe a DDI extraction system based on a machine learning approach. We propose distinct solutions to deal with the high dimensionality of the problem and the unbalanced representation of classes in the dataset. On the test dataset, our best run reaches an F-measure of 0.4702.

Keywords: Drug-drug interaction, machine learning, unbalanced classification, feature selection.

1 Introduction

One of the most relevant problems in patient safety is the adverse reaction caused by drugs interactions. In [3], it is claimed that 1.5 million adverse drug events and tens of thousands of hospital admissions take place each year. A Drug-Drug Interaction (DDI) occurs when the effect of a particular drug is altered when it is taken with another drug. The most updated source to know DDI is the pharmacological specialized literature. However, the automatic extraction of DDI information from this huge document repository is not a trivial problem. In this scenario, text mining techniques are very suitable to deal with this kind of problems.

Different approaches are used in DDI extraction. In [9], the authors propose a hybrid method based on linguistic and pattern rules to detect DDI in the literature. Linguistic rules grasp syntactic structures or semantic meanings that could discover relations from unstructured texts. Pattern-based rules encode the various forms of expressing a given relationship. As far as we know, there are not many works applying machine learning approaches to this task due to the inexistence of available corpora. In [10] a SVM classifier was used to extract DDI into the DrugDDI corpus. However, in the similar problem of protein-protein interaction (PPI) has been widely used obtaining promising effectiveness, as in [7]. The main advantages of this

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 59–65,
Huelva, Spain, September 2011.

approach are that they can be easily extended to new set of data and the development effort is considerably lower than manual encoding of rules and patterns.

In this paper we present a machine learning approach to extract DDI using the DrugDDI corpus [10]. Natural Language Processing (NLP) techniques are used to analyze documents and extracting features which represent them. The unbalanced proportion between positive and negative classes in the corpus suggest us the application of sampling techniques. We have experimented with several machine learning algorithms (SVM, Naïve Bayes, Decision Trees, Adaboost) in combination with feature selection techniques in order to reduce the dimensionality of the problem.

The paper is organized as follows. The system architecture is presented in section 2. In Section 3 we describe the set of features that represents each pair of drugs which appears in the documents. Also we present the feature selection methods used to reduce the initial set of attributes. Next, Section 4 describes the techniques that we have used to deal with this unbalanced classification problem. In Section 5 we evaluate the results obtained with the training corpus. The results on the test corpus are presented in Section 6. Finally, the conclusions are in Section 7.

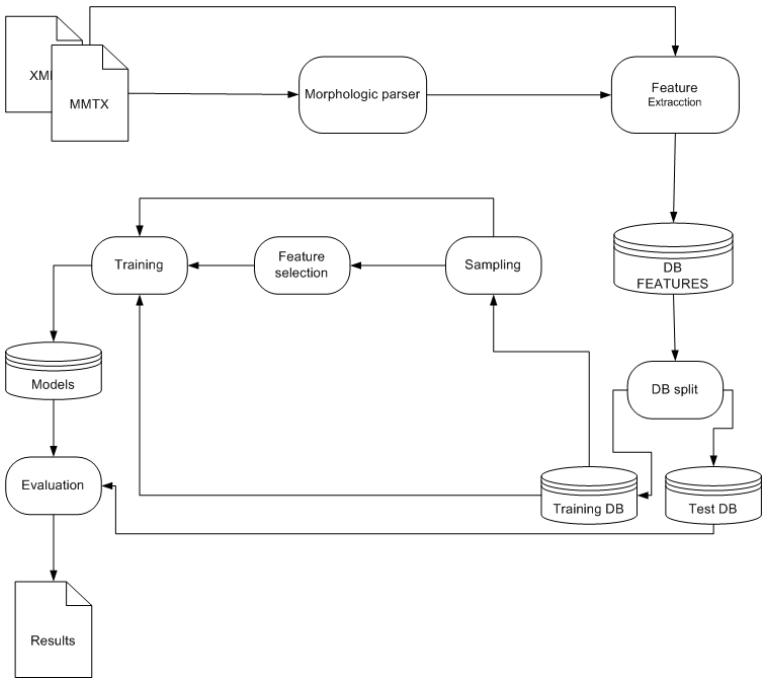


Fig. 1. System Architecture Diagram.

2 System Architecture

Two different document formats has been provided by the organizers, the Unified format and the MMTx format. We have used this last one to develop and testing our system.

The words around the drugs in a sentence have been selected as attributes of the database because they could provide clues about the existence of interaction between two drugs. We have experimented using the words as they appear in the documents and, in other cases, with the lemmas provided by the Stanford University morphologic parser[1].

For each drug pair in a sentence a set of features was extracted. The main features were focused on keywords, distances between drugs and drug semantic types. In the next section, a more detailed description of each attribute is done.

In order to carry out the experimentation, the DB of Features was split in two datasets for training and testing. We have used 2/3 of the original DB for training the classifier. The remaining 1/3 was used to test the system during the development phase.

Before training the classifier we have experimented with two preprocessing techniques. Because this problem is an unbalanced classification task we have carried out sampling techniques. Also, to reduce the dimensionality of the dataset a feature selection technique was performed. To obtain the model, we have experimented with several machine learning algorithms (SVM, Naïve Bayes, Decision Trees, Adaboost).

With each obtained model an evaluation was completed using the test dataset. The results obtained in this evaluation are shown in Section 5.

3 Feature Extraction and Selection

The most important part in this kind of classifying problem is to choose the set of features that represents as well as possible each pair of drugs. It means that we need to find those features that provide important information for differentiating pairs of drugs with interaction of pairs without interactions.

In this section we describe the features we have chosen to build the dataset.

3.1 Features

Firstly, we have extracted the drug ID, which indicates the sentence and the phrase of the dataset to which the drug belongs to.

Secondly, a feature subset composed by keywords was chosen. Each attribute is represented by a binary value that means the presence or absence of this keyword. Three windows of tokens have been considered to locate the keywords: between the first and the second drug, before the first drug and after the second drug. In the last two cases, only three tokens were taken into account.

[1] http://nlp.stanford.edu/index.shtml

In this work, a keyword is a word that could provide relevant information about whether a pair of drugs interacts or not. In order to build the list of keywords we extracted all the words between each pair of drugs, before the first drug or after the second drug, according the case. This set of words was filtered by a short list of stop-words. The POS tag of each word has been taken into account to make the selection. In this sense, we thought that verbs have an important semantic content, so we decided to include all of them into the final list. With respect to the nouns, we did a manual selection choosing those nouns that could be related semantically with drug interactions. Finally, in the case of prepositions, adverbs and conjunctions, we selected those that could be related with negation or frequency.

We have experimented using the keywords as they appear in the documents and, in other cases, with the lemmas provided by the Stanford University morphologic analyzer. In this case, the number of keywords was reduced because distinct verb tenses or plurals of a word were reduced to their lemmas, obtaining a total of 459 attributes.

Next, we added to the feature set the distance, in number of words and phrases, between the drugs. Also we included two features that represent the semantic type of each drug (represented by integer numbers).

Finally, the feature set is completed with the class, a binary value, where 1 means drug interaction and 0 if the pair does not interact.

As we can see in Table 1, we have extracted a total of 600 features from the original dataset to build the develop dataset.

Table 1. Feature set without lemmatization of the keywords.

Feature	Type	Number of features
Drugs ID	Integer	2
Keywords before first drug	Binary	153
Keywords between drugs	Binary	243
Keywords after second drug	Binary	197
Number of words between drugs	Integer	1
Number of phrases between drugs	Integer	1
Drug semantic types	Integer	2
Class	Binary	1
Total		600

3.2 Feature selection

Due to the high dimensionality of the training dataset, we have experimented with chi-squared feature selection method [8]. This method returns a ranking of the features in decreasing order by the value of the chi-squared statistic with respect to the class. We selected the attributes which the statistic had a value greater than 0. The resulting dataset, in the case of keywords without lemmatization, had 496 attributes.

4 Unbalanced Classification

As shown in Table 2, there are 23827 drug pairs in the develop dataset and only 2409 are real drug interactions. Therefore, the positive class is nearly the 10% (9.89%) of the total number of instances. It is a classification task with unbalanced classes. To deal with this problem we have used the SMOTE algorithm [2] in order to balance the classes.

Several classification algorithms have been selected in order to obtain the best effectiveness results with respect to the F-measure of the positive class. We have used the Weka [4] implementation of the following algorithms: RandomForest [1], Naïve Bayes [5], SMO [6] and MultiBoosting [11].

In some cases, to build the classification model, we have applied a cost sensitive matrix in order to penalize false positives.

5 Experimentation on Training Corpus

The develop corpus contains a collection of pharmacological texts labeled with drug interactions. This collection consists of 4267 sentences extracted from a total of 435 documents, which describe the interactions between drugs (Drug Drug Interactions or DDI). From these documents we have extracted 23827 drug pairs as possible cases of interaction. In total, there are 2409 instances corresponding to drug interactions and 21418 instances where there is no interaction between drugs.

Table 2 summarizes the training corpus statistics.

Table 2. Training corpus statistics.

Total different documents (files)	435
Number of documents containing, at least, one drug	412
Number of documents containing, at least, one drug pair	399
Total number of sentences	4267
Total number of drugs	11260
Total number of drug pairs	23827
Number of drug interactions	2409
Total entities that participate in a pair	10374
Average drug per document (documents and sentences with pairs)	25.88
Average drug per sentence (sentences with pairs)	4.67

In the experiment phase, we divided the dataset into two new datasets for training and testing, respectively. The training dataset consists of 2/3 of the total instances (15885). The test dataset consists of the remaining instances (7942).

The distribution of the instances for training and test datasets was done at random, keeping the percentage of instances with drug interaction and no interaction (10% and 90%, respectively).

Table 3 shows the effectiveness results for precision, recall and F-measure on the positive class of the 10 best evaluations. Each row of the table indicates a different

combination of classification algorithm, cost sensitive training, feature selection, sampling and keyword lemmatization.

As can be seen, the best results are obtained with the RandomForest algorithm. Moreover, the cost sensitive training, feature selection, sampling and lemmatization of the keywords contribute to achieve the best F-measures.

Table 3. Evaluation on training corpus. The second column is the classification algorithm. For RandomForest algorithm, the *I* parameter means the number of trees used to train the model. The *CST* column indicates whether the model has been built using a cost sensitive training. Different cost sensitive matrixes have been used in the experimentation phase. The *FS* column shows when feature selection has been carried out. The *Sampling* column has the same meaning with the application of SMOTE algorithm. Finally, *KW Lem.* column shows a lemmatization process has been performed.

RUN	Classification algorithm	CST	FS	Sampling	KW Lem.	Precision	Recall	F-Measure
1	RandomForest (*I* = 50)	X	X	X		0.573	0.617	0.595
2	RandomForest (*I* = 50)	X	X	X	X	0.578	0.610	0.594
3	RandomForest (*I* = 10)	X	X	X	X	0.500	0.654	0.567
4	RandomForest (*I* = 10)	X		X	X	0.492	0.644	0.558
5	RandomForest (*I* = 10)	X	X	X		0.565	0.548	0.556
6	RandomForest (*I* = 10)	X	X	X		0.469	0.677	0.554
7	RandomForest (*I* = 50)	X			X	0.645	0.472	0.545
8	MultiBoosting		X	X		0.674	0.443	0.535
9	RandomForest (*I* = 10)	X		X		0.544	0.520	0.532
10	RandomForest (*I* = 10)	X				0.587	0.471	0.523

6 Results on Test Corpus

In order to send runs with different characteristics, we didn't send the five runs with higher value of F-measure. According to Table 3, runs 1, 2, 4, 7 and 8 were submitted. We chose this strategy because we did not know the characteristics of the test corpus.

In Table 4, we present the results obtained for the five submitted runs. The approaches that obtain the best results on the training dataset coincide with the obtained on the test dataset. Although there are not significant differences between precisions on training and test datasets, a greater decrement in the recall measure do that the F-measure falls a 10% approximately. We think that this decrement in the effectiveness measures is due to a possible overfitting of the classification models.

7 Conclusions

In this paper we have presented a DDI extraction system based on a machine learning approach. We have proposed distinct solutions to deal with the high dimensionality of the problem and the unbalanced representation of classes in the dataset. The results obtained on both datasets are promising and we think that this could be a good starting point for future improvements.

Table 4. Evaluation on test corpus. The second column is the classification algorithm. For RandomForest algorithm, the *I* parameter means the number of trees used to train the model. The *CST* column indicates whether the model has been built using a cost sensitive training. Different cost sensitive matrixes have been used in the experimentation phase. The *FS* column shows when feature selection has been carried out. The *Sampling* column has the same meaning with the application of SMOTE algorithm. Finally, *KW Lem.* column shows a lemmatization process has been performed.

RUN	Classification algorithm	CST	FS	Sampling	KW Lem.	Precision	Recall	F-Measure
1	RandomForest (*I* = 50)	X	X	X		0.5000	0.4437	0.4702
2	RandomForest (*I* = 50)	X	X	X	X	0.4662	0.4291	0.4669
3	RandomForest (*I* = 10)	X		X	X	0.4004	0.4874	0.4397
4	RandomForest (*I* = 50)	X			X	0.6087	0.3152	0.4154
5	MultiBoosting		X	X		0.6433	0.2556	0.3659

References

1. Breiman, L. Random Forests. Machine Learning, 2001. Vol. 45(1):5-32.
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 2002. Vol. 16:321-357.
3. Classen, D.C., Phansalkar, S., Bates, D.W. Critical drug-drug interactions for use in electronic health records systems with computerized physician order entry: review of leading approaches. J. Patient Safety 2011 ,Jun;7(2):61-5.
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten. I.H. The WEKA Data Mining Software: An Update; SIGKDD Explorations 2009, Vol. 11, Issue 1.
5. John, G.H., Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.
6. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K. Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation, 2001. 13(3):637-649.
7. Krallinger, M., Leitner F., Valencia, A. The BioCreative II.5 challenge overview. Proceedings of the BioCreative II. 5 Workshop 2009 on Digital Annotations 2009, 19.
8. Liu, H., Setiono, R., Chi2. Feature selection and discretization of numeric attributes, Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, 338-391, 1995.
9. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchez, C. A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents, March, 2011, BMC BioInformatics, Vol. 12 (Suppl 2):S1.
10. Segura-Bedmar, I., Martinez, P., de Pablo-Sanchez, C. Using a shallow linguistic kernel for drug-drug interaction extraction, Journal of Biomedical Informatics, In Press, Corrected Proof, Available online 24 April 2011, DOI: 10.1016/j.jbi.2011.04.005.
11. Webb, G.I. MultiBoosting: A Technique for Combining Boosting and Wagging. Machine Learning 2000. Vol.40(No.2).

Drug-Drug Interactions Discovery Based on CRFs, SVMs and Rule-Based Methods

Stefania Rubrichi, Matteo Gabetta, Riccardo Bellazzi, Cristiana Larizza, and Silvana Quaglini

Laboratory for Biomedical Informatics "Mario Stefanelli",
Department of Computers and Systems Science,
University of Pavia, Pavia, Italy

Abstract. Information about medications is critical in improving the patients' safety and quality of care. Most adverse drug events are predictable from the known pharmacology of the drugs and many represent known interactions and are, therefore, likely to be preventable. However, most of this information is locked in free-text and, as such, cannot be actively accessed and elaborated by computerized applications. In this work, we propose three different approaches to the problem of automatic recognition of drug-drug interactions that we have developed within the "First Challenge Task: Drug-Drug Interaction Extraction" competition. Our approaches learn to discriminate between semantically interesting and uninteresting content in a structured prediction framework as well as a rule-based one. The systems are trained using the DrugDDI corpus provided by the challenge organizers. An empirical analysis of the three approaches on this dataset shows that the inclusion of rule-based methods is indeed advantageous.

Keywords: Drug-Drug Interactions, Information Extraction, Conditional Random Fields, Support Vector Machines, Adverse Drug Events

1 Background

The use of medications has a central role in health care provision, yet on occasion it may endanger patients' safety and account for increased health care costs, as result of adverse drug events (ADEs). Many of these injuries are inevitable, but at least a quarter may be secondary to medication errors [7] that can be avoidable. That is the case of ADEs due to drug-drug interactions (DDIs), since many of them are due to disregarded known interactions and are therefore likely to be preventable. Over the 6.5% of drug-related hospital admissions are a consequence of DDIs.

DDIs are a common problem during drug treatment. Widely, a drug interaction represents the situation in which a substance affects the activity of an active ingredient, resulting in various effects such as alterations in absorption,

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 67–74, Huelva, Spain, September 2011.

metabolism, excretion, and pharmacodynamics (i.e. the drug effects are decreased or increased, or the drug produces a new effect that neither produces on its own). Safe medication use requires that prescribers receive clear information on the medication itself including information about any potential interactions. This information is constantly changing, and while most of the necessary updated knowledge is available somewhere, it is not always readily accessible. In particular, most of this information is locked in free-text, then cannot be actively used by health information systems. Reliable access to this comprehensive information, by Natural Language Processing (NLP) systems, can represent a useful tool for preventing medication errors and, more specifically, DDIs. Over the last two decades there has been an increase of interest in applying NLP, in particular information extraction (IE) techniques, to biomedical text. Excellent efforts have been documented in the medication domain literature on IE from textual clinical documents [4,5,9,11,12,14,15,18], and its subsequent application in summarization, case finding, decision-support, or statistical analysis tasks.
In this context, we accepted the challenge presented within the "First Challenge Task: Drug-Drug Interaction Extraction" competition and developed a system for the automatic extraction of DDIs from a corpus [13] of documents, collected from the DrugBank database [8], describing, for each drug, the relating DDIs.

2 Methods

On the following section we present the proposed system and its components.

2.1 System Outline

We exploit three different approaches, which rely upon different methods for the extraction of such information. The first approach (henceforth referred as hybrid approach) is twofold: it combines a supervised learning technique based on Conditional Random Fields (CRFs) [16] with a rule-based method. We modeled the problem as follows: in a first step we employed the CRFs classifier in order to assign the correct semantic category to each word, or segment of sentence, of the text. We considered the following three semantic categories:

1. *DrugNotInteracting*: describes a drug entity, which is not involved in an interaction;
2. *DrugInteracting*: describes a drug entity, which is involved in an interaction;
3. *None*: indicates elements that are not relevant for this task.

Once every potential interacting entity has been identified by the CRFs classifier, we defined a set of rules for the construction of the actual pairs of interacting entities, and match them with the sentences.
The second (henceforth referred as pair-centered CRFs approach) and third (henceforth referred as pair-centered SVMs approach) approaches are very similar: they are both based on supervised learning methods, CRFs and Support

Vector Machines (SVMs) [2,17], respectively. In this case we focused on the single pair of drug entities: for any given pair in a sentence, such techniques predict the presence or absence of interaction relation, relying on a set of hundreds of engineered features, which take into account the properties of the text, by learning the correspondence between semantic categories and features. We considered only two semantic categories:

1. *Interaction*: describes a pair of drug entities which interact;
2. *NotInteraction*: describes a pair of drug entities which don't interact;

All these three methodologies have been developed through different steps. We began with a pre-processing pass over the corpus in order to prepare the dataset for the use by the extraction module. Then, we defined a set of binary features that express some descriptive characteristics of the data, and we converted the data in a set of corresponding features. Finally, we processed the data through the three methodologies described above.

2.2 Supervised Learning Methods: CRFs and SVMs

Supervised learning approaches have been widely applied to the domain of IE from free text. A typical application of supervised learning works to classify a novel instance x as belonging to a particular category y. Given a predefined set of categories, such methods use a set of training examples to take decision in front of new examples. They automatically tune their own parameters to maximize their performance on the training set and then generalize from the new samples. We processed the data through the two linear classifiers, CRFs and SVMs: both algorithms iterate the tokens in the sentence, and label proper tokens with semantic categories. These classifiers discriminate between semantically interesting and uninteresting content through the automatic adaptation of a large number of interdependent descriptive characteristics (features) taking into account the properties of the input text. Each token is represented by a set of features, then the classifiers learn a correspondence between semantic categories and features, and assign real-valued weight to such features.

2.3 Pre-processing

The first step of our DDIs detection system has been a pre-processing over the data provided within the challenge contest.
We designed two different pre-processing strategies, one for the hybrid approach, the other one for the pair-centered CRFs and the pair-centered SVMs approach. The first pre-processing strategy analyzes sentence-by-sentence the training corpus, using a quite classical NLP system developed using Gate [3], an open source framework for language processing. This system includes:

– Tokenizer: splits the atomic parts of the sentence (tokens) according to a specific language (English in our case);

- Part of Speech (POS) Tagger [6]: assigns to the tokens their grammatical class (e.g. noun, verb, adjective ...);
- Morphological Analyzer: assigns the lexical roots to the tokens;
- UMLS concept finder: a module we developed, in order to discover concepts referable to the Unified Medical Language System (UMLS) [10] within the text.

The pre-processing system returns as output a line for each token; such line contains the token itself together with additional information necessary for the features generation task. In particular:

- the semantic category of the token itself;
- the "entity tag" that is the entity's code (e.g. DrugDDI.d385.s4.e0) when the token is an entity and null otherwise;
- the "main drug tag" that is *true* if the token matches the standard name of the referential drug[1] and *false* otherwise;
- the "brand name tag" that is *true* if the token matches one of the brand names of the referential drug and *false* otherwise. Brand names come from the DrugBank;
- the "POS tag" that is the grammatical class provided by the POS Tagger (entities are automatically tagged as proper nouns - NNP);
- the "root tag" which is the root of the token provided by the Morphological Analyzer (the entity itself for the entities);
- the "semantic group tag" that, when the token belongs to a UMLS concept, is the semantic group of the concept itself (e.g. "DISO" for concepts belonging to the "Disorders" group); it is "ENT" when the token is an entity and null otherwise.

As an example, given the input sentence:

```
<sentence id="DrugDDI.d368.s0" origId="s0" text="Itraconazole
decreases busulfan clearance by up to 25%, and may produce AUCs >
1500 muMolmin in some patients.">
    <entity id="DrugDDI.d368.s0.e0" origId="s0.p0" charOffset="0-12"
type="drug" text="Itraconazole" />
    <entity id="DrugDDI.d368.s0.e1" origId="s0.p2" charOffset="23-31"
type="drug" text="busulfan" />
    <pair id="DrugDDI.d368.s0.p0" e1="DrugDDI.d368.s0.e0"
e2="DrugDDI.d368.s0.e1" interaction="true" />
</sentence>
```

the first pre-processing strategy will generate the following lines:

```
itraconazole-DrugInteracting-DrugDDI.d368.s0.e0-false-false-NNP-
    itraconazole-ENT
decreases-None-null-false-false-NNS-decrease-CONC
busulfan-DrugInteracting-DrugDDI.d368.s0.e1-true-false-NNP-busulfan-
```

[1] We indicate by "referential drug" the drug described in the specific document under examination.

```
    ENT
clearance-None-null-false-false-NN-clearance-PHEN
...
```

and so on.
The second pre-processing strategy evaluates separately all the pairs within a sentence; it uses the same NLP system described for the first strategy, but it formats the output in a different way. For each pair, the output consists of a header line, containing the codes of the involved entities and the semantic category of the pair. The header line is followed by a line for each token standing between the two entities involved in the pair; for each line the elements describing the token are exactly the same as those described for the first strategy (*token, interaction tag, entity tag, etc.*).
Given the input sentence from the previous example, the second pre-processing strategy will generate the following lines:

```
DrugDDI.d368.s0.e0 DrugDDI.d368.s0.e1-Interaction
decreases-None-null-false-false-NNS-decrease-CONC
```

2.4 Feature Definition and Data Conversion

The feature construction process aims at capturing the salient characteristics of each token in order to help the system to predict its semantic label. Feature definition is a critical stage regarding the success of feature-based statistical models such as CRFs and SVMs. A careful inspection of the corpus has resulted in the identification of a set of informative binary features that capture salient aspects of the data with respect to the tagging task. Subsequently, the stream of tokens has been converted to features. In particular, in the pair-centered CRFs and pair-centered SVMs approaches we considered only the tokens between the two entities which form each pair. This means that features for drug entities pair E_1-E_2 contain predicates about the n tokens between E_1 and E_2.
In the following we report on the set of features used in our experiments.

Orthographical Features As a good starting point, this class of features consists of the simplest and most obvious feature set: word identity feature, that is the vocabulary derived from the training data.

Part Of Speech (POS) Features We supposed lexical information might be quite useful for identifying named entities. Thus, we included features that indicate the lexical function of each token.

Punctuation Features Also notable are punctuation features, which contain some special punctuation in sentences. After browsing our corpus we found that colon might prove helpful. Given a medication in fact, colon is usually preceded by the interacting substance and followed by the explanation of the specific interaction effects.

Semantic Features In order to have these models benefit from domain specific knowledge we added semantic features which use external semantic resources. This class of features includes:

1. *root feature*: takes account of the root associated to each word;
2. *UMLS feature*: relies on the UMLS Metathesaurus and for each word returns the corresponding semantic group;
3. *brand name feature*: it recognizes the corresponding brand names occurring in the text. DrugBank database drug entries are provided with the field "Brand Names", which contains a complete list of brand names from different manufacturers. We create a binary feature, which, every time a text token coincides with one of such names, is active, indicating that the token corresponds to a brand name of the specific referential drug;
4. *standard drug name feature*: identifies the standard name of the source drug. For each token this feature tests if it matches such standard name;
5. *drug entity feature*: allows the models to recognize the drug entities annotated by the MetaMap tool: it is active for the tokens which have been annotated as drug entity by the MetaMap tool.

Context Features Finally, we extended all the classes of feature we described above to a token window of [-k,k]. The descriptive characteristics of tokens preceding or following a target token may be useful for modeling the local context. It is clear that the more context words analyzed, the better and more precise the results could become. However, widening the context window quickly leads to an explosion of the computational and statistical complexity. For our experiments, we estimated a suitable window size of [-3,3].

2.5 Rule-based Method

As we have already stated, while both pair-centered CRFs and pair-centered SVMs approaches focus on entities pairs and predict directly the presence or absence of interaction, the first one considers a token at a time, then the semantic category prediction is on a token-by-token basis. Therefore, a further processing pass was necessary in order to build up the interaction pairs, starting from each single entity. For this purpose, we employed a rule-based method which relies upon a set of rules, manually-constructed from the training data analysis. In particular, the rules that we built to find out the interacting pairs are the following:

- if a sentence contains less than two tokens labeled as *DrugInteracting*, then no interacting pair is generated;
- an interacting pair must contain two tokens labeled as *DrugInteracting*;
- one and only one of the token involved in the interacting pair, must be the referential drug or one of its brand names.

3 Experiments

We used the Unified format of the DrugDDI corpus [1] provided by the competition organizers.
For the linear SVMs, we found the regularization parameter $\lambda = 1$ to work well. SVMs results have been produced using 10 passes through the entire training set. For the variance of the Gaussian regularizer of the CRFs we used the value 0.1.
We submitted a total of three runs: the first run includes the predictions generated by the hybrid approach; the second run includes the predictions generated by the pair-centered CRFs approach; the third run includes the predictions generated by the pair-centered SVMs approach.

4 Results and Discussion

The evaluation process was performed by the challenge organizers.
The overall results of the three approaches can be found in Table 1. In general, the hybrid approach outperforms the other two. This performance gain can be attributed to the additional contribute of rule-based method, that played an important role in building the interacting pairs. In particular it makes the system benefit from additional knowledge that facilitates the pairs disambiguation process. It specifies, for example, that a pair has to include the referential drug or one of its brand names together with another drug entity different from them.
There is room for improvement, especially for the pair-centered CRFs and pair-centered SVMs approaches. In such approaches we mainly relied on tokens occurring between the two entities which form each pair, however tokens preceding and following the pairs could also be taken into account.

Table 1. Overall experimental results of the different runs

Approach	**Hybrid**	**Pair-centered CRFs**	**Pair-centered SVMs**
True Positive	369	196	317
False Positive	545	110	456
False Negative	386	559	438
True Negative	5726	6161	5815
Precision (%)	40.37	64.05	41.01
Recall (%)	48.87	25.96	41.99
F_1 Score (%)	44.22	36.95	41.49

5 Conclusion and Future Work

In this paper we presented three different approaches for the extraction of DDIs that we have developed within the "First Challenge Task: Drug-Drug Interaction Extraction" competition. We employed three different methodologies: two machine learning-based (CRFs and SVMs) and one which combines a machine learning-based (CRFs) with a rule-based technique. The latter achieved better results with an overall F_1 score of about 44%. This figure doesn't seem encouraging: the comparison with the other systems that face the same problem with the same corpus within this competition probably will allow to understand this result and realize the weakness of our approaches.

References

1. http://labda.inf.uc3m.es/ddiextraction2011/dataset.html
2. Bordes, A., Usunier, N., Bottou, L.: Sequence labelling SVMs trained in one pass. In: ECML PKDD 2008. pp. 146–161. Springer (2008)

3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02) (2002)
4. Evans, D.A., Brownlowt, N.D., Hersh, W.R., Campbell, E.M.: Automating concept identification in the electronic medical record: An experiment in extracting dosage information. In: Proc. AMIA Annu Fall Symp. pp. 388–392 (1996)
5. Gold, S., Elhadad, N., Zhu, X., Cimino, J.J., Hripcsak, G.: Extracting structured medication event information from discharge summaries. In: Proc. AMIA Annu Symp. pp. 237–241 (2008)
6. HeppleIn, M.: Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000) (2000)
7. Institute of Medicine (ed.): Preventing Medication Errors. The National Academics Press, Washington (2007)
8. Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A., Wishart, D.: Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. Nucleic Acids Res (2011)
9. Levin, M.A., Krol, M., Doshi, A.M., Reich, D.L.: Extraction and mapping of drug names from free text to a standardized nomenclature. In: Proc AMIA Annu Symp. pp. 438–442 (2007)
10. Lindberg, D., Humphreys, B., McCray, A.: The unified medical language system. Methods Inf Med (1993)
11. Pereira, S., Plaisantin, B., Korchia, M., Rozanes, N., Serrot, E., Joubert, M., Darmoni, S.J.: Automatic construction of dictionaries, application to product characteristics indexing. In: Proc Workshop on Advances in Bio Text Mining (2010)
12. Segura-Bedmar, I., Martínez, P., de Pablo-Sánchez, C.: Extracting drug-drug interactions from biomedical texts. In: Workshop on Advances in Bio Text Mining (2010)
13. Segura-Bedmar, I., Martinez, P., de Pablo-Sanchez, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics, In Press (2011)
14. Shah, A.D., Martinez, C.: An algorithm to derive a numerical daily dose from unstructured text dosage instructions. Pharmacoepidemiology and Drug Safety 15, 161–166 (2006)
15. Sirohi, E., Peissig, P.: Study of effect of drug lexicons on medication extraction from electronic medical records. In: Proc. Pacific Symposium on Biocomputing. vol. 10, pp. 308–318 (2005)
16. Sutton, C.: Grmm: Graphical models in mallet. http://mallet.cs.umass.edu/grmm/ (2006)
17. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
18. Xu, H., Stenner, S.P., Doan, S., Johnson, K.B., Waitman, L.R., Denny, J.C.: Medex: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association 17, 19–24 (2010)

AN EXPERIMENTAL EXPLORATION OF DRUG-DRUG INTERACTION EXTRACTION FROM BIOMEDICAL TEXTS

Man Lan, Jiang Zhao, Kezun Zhang, Honglei Shi, and Jingli Cai

East China Normal University, Shanghai, P.R.China

Abstract. The First Challenge of Drug-Drug Interaction Extraction (DDIExtraction 2011) involves doing a binary DDI detection to determine whether a drug pair in a given sentence (with annotated drug names) has interaction information. This may be the first attempt at extraction of drug interaction information in wide community. In this paper we compare and evaluate the effectiveness of different strategies of example generation from texts and different feature types for drug relation extraction. The comparative results show that (1) drug interaction classification at drug entity pair level performs better than that at sentence level; (2) simple NLP output does not improve performance and more advanced way of incorporating NLP output need to be explored.

1 Introduction

In pharmacology domain, one drug may influence the level or activity of another drug if there is a drug-drug interaction (DDI) between them. Typically, the detection of DDIs between drug pair is an important research area for health care professionals to find dangerous drug interactions and possible side effects, which helps to decrease health care costs.

Like other entity (e.g., gene or protein) relation extraction tasks (i.e., BioCreAtIvE) from biomedical literature, information extraction (IE) techniques can provide an interesting way of reducing the time spent by health care professionals on reviewing the literature. Recently, DDIExtraction Challenge 2011 has played a key role in comparing various IE techniques applied to the pharmacological domain by providing a common benchmark for evaluating these techniques. Specifically, they create the first annotated Drug DDI corpus that studies the phenomena of interactions among drugs. Meanwhile, the organizers have devoted to several comparative experimental assessments of different exploration strategies on this corpus, e.g., Segura-Bedmar et al. (2010a), (2010b), (2011a) and (2011b). For example, they manually created linguistic rules (i.e. pattern) using shallow parsing and syntactic and lexical information with the aid of domain expert in Segura-Bedmar et al. (2010a) and (2011b). Moreover, they adopted shallow linguistic kernel-based supervised machine learning (SVM) method to build relation classifier for DDI extraction. Their experimental results showed

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 75–82
Huelva, Spain, September 2011.

that the sequence kernel-based method performs significantly better than the construction of linguistic rules.

The basic idea of our system is to make use of feature-based supervised machine learning approach for DDI extraction. Our work consists of two explorations, i.e., comparison of different strategies of example generation from texts and comparison of different feature types. The purpose of this work is twofold: (1) compares the performance of different strategies of example generation, different feature types for drug interaction extraction; (2) provides an overview of our practical and effective process for this challenge.

The rest of the paper is structured as follows. Section 2 describes the overview of DDIExtraction Challenge 2011. Section 3 presents the methods adopted in our participation. Section 4 describes the system configurations and results on the test data. Finally, Section 5 summarizes the concluding remarks and suggests the future work.

2 Overview of DDIExtraction Challenge 2011

In recent years, most biomedical relation extraction study and corpora have focused on describing genetic or protein entity interactions, e.g., BioInfer (2007), BioCreative II (2008) and II.5 (2009), or AIMed (2005), rather than drug-drug interaction. The First Challenge of Drug-Drug Interaction Extraction (i.e., DDIExtraction Challenge 2011) provides a new standard benchmark and creates the first annotated corpus for drug interaction extraction to a wider community. The DDI corpus is created by Segura-Bedmar et al.(2011a). The Drug DDI corpus consists of 579 documents describing DDI, which are randomly selected from the DrugBank database (2008). In DDIExtraction Challenge 2011, this corpus is split into 435 training documents (4267 sentences) and 144 test documents (1539 sentences) for evaluation. Table 1 lists the detailed various statistical information of training and test data set. From this table, we can see that the data distribution in training data set is quite close to that in test data set.

This corpus is provided in two different formats: (1) the unified XML format as the PPI Extraction format proposed in Pyysalo et al. (2008) and (2) a Metamap format based on the information provided by the UMLS MetaMap Transfer (MMTx) tool (2001). In MMTx format, the documents were analyzed by the MMTx tool that performs sentence splitting, tokenization, POS-tagging, shallow syntactic parsing, and linking of phrases with Unified Medical Language System (UMLS) Metathesaurus concepts. Besides, the MMTx format documents annotate a variety of biomedical entities occurring in texts according to the UMLS semantic types. An experienced pharmacist recommended the inclusion of the following UMLS semantic types as possible types of interacting drugs: (1) Clinical Drug (clnd), (2) Pharmacological Substance (phsu), (3) Antibiotic (antb), (4) Biologically Active Substance (bacs), (5) Chemical Viewed Structurally (chvs) and (6) Amino Acid, Peptide, or Protein (aapp).

Clearly, the MMTx format contains not only shallow NLP information but also domain-specific annotations. Therefore it is expected to provide more useful

Table 1. Statistical information of training and test data set.

Category	Training set	Test set
#-Documents	435	144
#-Sentences	4267	1539
#-Drug entities	11260	3689
#-Drug pairs	23827	7026
#-DDIs	2402	755
#-Documents containing, at least, one drug pair	399	134
#-Sentences with, at least, one drug pair	2812	965
#-Sentences with, at least, one DDI	1530	503
#-Total entities that participate in a pair	10374	3398
Avg drug per doc (considering only docs with drug pairs)	26.02	25.36
Avg drug per sentence (considering only sentences with drug pairs)	3.69	3.52
Avg DDI per doc (considering only docs with drug pairs)	6.02	5.63
Avg DDI per sentence (considering only sentences with drug pairs)	0.85	0.80

information than unified XML format for DDI extraction. Consequently, participants are required to indicate the document format their methods involved. Another thing need to note is that this challenge only considers the interactions between drugs within the same sentence.

Participants are allowed to submit a maximum of 5 runs. For each drug pair within one sentence, the participated algorithm is expected to generate label "0" for non-authentic DDI and label "1" for predicted DDI. For performance evaluation, this challenge adopted the most widely-used text classification evaluation measures, i.e., precision (P), recall (R) and their combination F_1 score.

3 Methods

In our work we cast drug relation extraction as a classification problem, in which each example is generated from texts and formed as a feature vector for classification. Specifically, we generate examples from all sentences containing at least two drug entities. That is, the sentences which have none or only one drug should be removed first before they come into the pipeline of text processing.

Here we need to take into account the following special considerations. One is the issue of example generation from texts. Another is the issue of feature types extracted from texts. Next we will discuss these two special considerations.

3.1 Example Generation

The training and test examples from texts can be generated at different levels, e.g., sentence level or drug pair level.

At sentence level, each example corresponds to one sentence. That is, each sentence is represented as a feature vector, no matter how many DDIs this sentence has. Typically, a sentence having n drugs ($n \geq 2$) generates C_n^2 drug

pairs but not all drug pairs are DDIs. Thus, in order to assign the DDI label to each sentence, we have the following two assumptions and they serve as baselines in our work.

Assumption 1: In training step, if there is at least one DDI annotated in the sentence, we assign the DDI label of this sentence 1. That is, this sentence is assumed to be a DDI sentence. In test step, if one sentence is predicted by classifier to be a DDI sentence, then all drug pairs within this sentence are predicted to be DDIs as well.

Assumption 2: In training step, if the number of DDIs is equal to or larger than the number of non-DDIs in the sentence, we label this sentence as DDI sentence. That is, for a sentence having n drugs, if it has at least $C_n^2/2$ DDIs, it is regarded as DDI sentence. In test step, if one sentence is predicted by classifier to be a DDI sentence, all drug pairs within this sentence are predicted to be DDIs as well.

Clearly, the built-in flaw of the above two assumptions is that they consider all drug-pairs in one sentence have one common taxonomy label. This is not true in real world case. We use the two assumptions as baseline systems in our work.

At drug pair level, each example corresponds to each drug pair in a sentence. That is, the number of examples generated for each sentence is given by the combinations of distinct drug entities (n) selected two at a time, i.e. C_n^2. For example, if one sentence contains three drug entities, the total number of examples generated from this sentence is $C_3^2 = 3$. In training step, for each example, we use its annotated DDI label as the label of this example. If a DDI relation holds between a drug pair, the example is labeled 1; otherwise 0. In test step, for each drug pair, the classification system predicts its DDI label based on the classifier constructed on training examples.

3.2 Features Extraction

No matter which level examples are generated from texts, the examples are represented as feature vectors for classifier construction and prediction. Here we describe the feature sets adopted by above two example generation approaches.

As for sentence level feature representation, we adopt a feature set consisting of all words in texts. Specifically, we remove stop words (504 stop words), punctuation, special characters and numbers from sentences.

As for drug pair level feature representation, instead of using all words in texts, we explore different feature types, i.e., lexical, morpho-syntactic, semantic and heuristic features (from annotated biomedical information), with the purpose of capturing information between drug pairs. The features consist of the following 6 types. The first two feature types are generated from unified XML text format. The following four feature types are obtained from MMTx text format.

F1: Token between drug pair. This feature includes the tokens (words) between two target drug entities. Given two annotated target drug entities, first all the words between them are extracted and then the Porter's stemming (1980) is performed to reduce words to their base forms.

F2: Lemma of target entities. This feature consists of the lemma of the target drug entities annotated in the given sentence. That is, this feature records the words of the target drug names.

F3: UMLS semantic types of target entities. This feature is to record the six UMLS semantic types of the drug entities annotated in the given sentence.

F4: Information of other drug entities. This feature is to indicate whether there is other drugs between the current target drug pair and the number of other drug entities.

F5: Relative position between verbs and target drug entities. This feature is to record if there is verb before, between or after the target drug pair.

Except for the above two approaches, we also explore experiment using only the position information of verbs and target drug entities as follows.

F6: Position of verbs and target drug entities. This feature is different from above 5 feature types, which only records the position information of verbs and drug entities. To do so, for the first drug entity, we record the relative positions of three closest verbs before it and after it. For example, if the position of the two verbs offset is 10 and 11, and the position of the first drug is 15, the first three feature values is 5, 4 (relative position) and 0 (since no third verb before the first drug). For the second drug entity, we record the relative positions of three closest verbs after it. In addition, we also assign one label for each verb to record if there is a negation before it, yes for 1 and no for 0. We manually created list of 16 negation words including: *little*, *few*, *hardly*, *never*, *none*, *neither*, *seldom*, *scarcely*, *rarely*, *cannot*, *can't*, *isn't*, *hasn't*, *couldn't*, *unlike*, *without*.

3.3 Learning Algorithms

Generally, according to the different kernel functions from computational learning theory, SVMs are classified into two categories, i.e., linear and nonlinear (such as polynomial, radial-based function (RBF), etc). Specifically, in this study, we adopt the radial-based nonlinear SVM because in our preliminary study the nonlinear SVM performs better than linear SVM models. The SVM software we used in all experiments is LIBSVM-2.9 (2001).

4 Results And Discussion

4.1 Text Preprocessing

In text processing step, the stop words (504 stop words), punctuation and numbers were removed. The Porter's stemming (1980) was performed to reduce words to their base forms. The resulting vocabulary has 3715 words (terms).

4.2 System configuration and Results

In this work, we config five different classification systems with different example generation strategies and different feature types. The classifiers for all systems

were optimized independently in a number of 5-fold cross-validation (CV) experiments on the provided training sets. First we consider two baseline systems at sentence level described in section 3.1. We create a global feature set consisting of all words in texts. The resulting vocabulary of the two systems has 3715 and 3224 words (terms) respectively. Table 2 shows the results of the first two systems at sentence level.

Table 2. Two system configurations at sentence level with two assumptions and results on the test data.

System	Description (sentence level)	P (%)	R (%)	F_1(%)
1	assumption 1, all words in texts	14.37	76.82	24.21
2	assumption 2, all words in texts	39.63	16.95	23.75

In the third system, we conducted several comparative experiments at drug pair level using different combination of features described in section 3.2. In addition, in the fourth system, we evaluated the system with only relative position information between drugs and verbs in one sentence. Finally, in the fifth system, we performed majority voting to combine the best results of the first four systems to further improve performance. Table 3 shows the results of these three systems at drug pair level using different feature sets.

Table 3. System configurations at drug pair level with different feature types and results on the test data.

System	Description (drug pair level)	P (%)	R (%)	F_1(%)
3	F1	31.49	68.48	43.14
	F1, F2	28.08	42.91	33.94
	F1, F2, F3	32.70	31.92	32.31
	F1, F2, F3, F4	37.96	31.13	34.21
	F1, F2, F3, F4, F5	41.71	35.63	38.43
4	F6	32.70	27.28	29.75
5	Majority voting	29.57	46.49	36.15

4.3 Discussion

Based on the above series of experiments and results shown in Table 2 and Table 3, some interesting observations can be found as follows.

Specifically, the first two baseline systems at sentence level yield quite similar F-measures of 24.21 and 23.75 but different recall and precision. The first system has high recall but low precision. Conversely, the second system has high

precision but quite low recall. This difference comes from the different principle of the two assumptions. This F-measure is similar to the result reported in Segura-Bedmar et al. (2011b) using only linguistical patterns with the aid of domain expert.

Generally, the systems at drug pair level (Table 3) perform better than those at sentence level (Table 2). This result is consistent with our preliminary surmise that it is too rough for example generation at sentence level and it did not take the relation between drug pair into consideration. Certainly many previous work on entity relation extraction generated example using this representation.

Moreover, the comparative result of the third serial of systems, i.e., the systems at drug pair level with different feature sets, is beyond our preliminary expectation. Surprisingly, the system with only words between two drug entities performs the best among the serial of the third systems. Although we extracted and constructed more features which are supposed to hold more useful information, such as drug names, drug types and the position information between drug and verb, these features did not improve the performance. One possible explanation is that the number of F1 feature is much larger than other features, and thus F1 feature dominates the performance of classifier. Another possible reason is that these manually constructed or NLP features may not be appropriate for representation and thus more advanced NLP techniques and advanced ways of incorporating NLP output is necessary for future exploration.

Another surprise is that the fourth system performs better than the two baseline systems at sentence level but still worse than the third system. Since the fourth system only considers relative position information rather than words and other features, this result is quite interesting. However, we do not expect more improvement on this simple feature set and we have no further explorations.

As an ensemble system, the fifth system combines the best results of the previous four systems. However, this majority voting strategy has not shown significant improvements. The possible reason may be that these classifiers come from a family of SVM classifiers and thus the random errors are not significantly different.

5 Summary

Based on the comparative experimental results, we summarized that, first, example generated at drug pair level performs better than sentence level; second, using only words between drug pair entities performs better than adding more constructed NLP and domain-specific features. It indicates that NLP output has not yet succeeded in improving classification performance over the simple bag-of-words approach and more advanced way of incorporating NLP output need to be explored.

We have to mention that although the best performance on the test set yields a final score of no more than 45% (F-measure), which is quite lower than the best performance 60.01% reported in Segura-Bedmar et al. (2011a), it is still quite promising since we do not involve domain expert, domain knowledge and

complicated NLP outputs neither. In other words, this suggests that there may be ample room for improving the performance.

ACKNOWLEDGMENTS

This research is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Pujiang Talent Program (No.09PJ1404500) and Doctoral Fund of Ministry of Education of China (No.20090076120029).

References

Pyysalo, F., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., and Salakoski, T. Comparative analysis of protein-protein interaction corpora. BMC Bioinformatics 9 (Suppl 3) : S6 (2008).

Isabel Segura-Bedmar, Paloma Martinez, Cesar de Pablo-Sanchez. Extracting drug-drug interactions from biomedical texts. BMC Bioinformatics 2010, 11(Suppl 5):P9.

Isabel Segura-Bedmar, Paloma Martinez, Cesar de Pablo-Sanchez. A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. BMC Bioinformatics 2011, 12(Suppl 2):S1.

Giuliano C, Lavelli A, Romano L. Exploiting shallow linguistic information for relation extraction from biomedical literature. Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006) 2006, 5-7.

Isabel Segura-Bedmar, Paloma Martinez, Cesar de Pablo-Sanchez. Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics, April 2011.

Isabel Segura-Bedmar, Paloma Martinez, Cesar de Pablo-Sanchez. Combining syntactic information and domain-specific lexical patterns to extract drugCdrug interactions from biomedical texts. In: Proceedings of the ACM fourth international workshop on data and text mining in biomedical informatics (DTMBIO10); 2010. p. 49-56.

Sampo Pyysalo , Filip Ginter , Juho Heimonen , Jari Bjorne , Jorma Boberg , Jouni Jarvinen and Tapio Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. BMC Bioinformatics 2007, 8:50

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

Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, Wong YW. Comparative experiments on learning information extractors for proteins and their interactions. Artificial Intelligence in Medicine 2005, 33(2):139-155.

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

Aronson A: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proceedings of the AMIA Symposium 2001, 17-22.

M. Porter. An algorithm for suffix stripping. Program, vol. 14, no. 3, pp.130-137, 1980.

C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.

EXTRACTION OF DRUG-DRUG INTERACTIONS USING ALL PATHS GRAPH KERNEL.

Shreyas Karnik[1,2], Abhinita Subhadarshini[1,2], Zhiping Wang[2], Luis M. Rocha[3,4], and Lang Li*[2,5]

[1] School of Informatics, Indiana University, Indianapolis, IN, USA, 46202
[2] Center for Computational Biology & Bioinformatics, School of Medicine, Indiana University , Indianapolis, IN, USA, 46202
[3] School of Informatics & Computing, Indiana University, Bloomington, IN, USA, 47408
[4] Instituto Gulbenkian de Ciencia, Oeiras, Portugal
[5] Department of Medical & Molecular Genetics, School of Medicine, Indiana University , Indianapolis, IN, USA, 46202
Corresponding Author: lali@iupui.edu

Abstract. Drug-drug interactions (DDIs) cause nearly 3% of all hospital admissions. Regulatory authorities such as the Food and Drug Administration (FDA) and the pharmaceutical companies keep a rigorous tab on the DDIs. The major source of DDI information is the biomedical literature. In this paper we present a DDI extraction approach based on all paths graph kernel [1] from the DrugDDI corpus [2]. We also evaluate the method on an in-house developed clinical *in vivo* pharmacokinetic DDI corpus. When the DDI extraction model was evaluated on the test dataset from both corpora we recorded a F-score of 0.658 on the clinical *in vivo* pharmacokinetic DDI corpus and 0.16 on the DrugDDI corpus.

1 Introduction

Polypharmacy has been a general clinical practice. More than 70% of old population (age >65) take more than 3 medications at the same time in US and some European countries. Since more than 80% of the drugs on the market are metabolized by the Cytochrome P450 enzyme system, and many of these drugs are inhibitors and inducers of CYP450 enzyme system, drug interactions have been extensively investigated *in vitro* and *in vivo* [3,4,5]. These DDIs in many ways affect the overall effectiveness of the drug or at some times pose a risk of serious side effects to the patients [6]. Thus, it becomes very challenging to for the successful drug development and clinical patient care. Regulatory authorities such as the Food and Drug Administration (FDA) and the pharmaceutical companies keep a rigorous tab on the DDIs. Major source of DDI information is the biomedical literature. Due to the unstructured nature of the free text in the biomedical literature it is difficult and laborious process to extract and analyze the DDIs from biomedical literature. With the exponential growth of the

Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011), pages 83–88, Huelva, Spain, September 2011.

biomedical literature, there is a need for automatic information extraction (IE) systems that aim at extracting DDIs from biomedical literature. The use of IE systems to extract relationship among biological entities from biomedical literature has experienced success to a great scope [7] for example protein-protein interaction extraction. Researchers have now started to investigate DDI IE from biomedical literature. Some early attempts include retrieval of DDI relevant articles from MEDLINE [8] ;DDI extraction based on reasoning approach [9]; DDI extraction based on shallow parsing and linguistic rules[10];and DDI extraction based on shallow linguistic kernel [2].

BioCreAtIvE has established the standard of evaluation methods and datasets in the area of information extraction [7,11,12,13] which has been a asset for the community. To encourage the involvement of the community in the DDI extraction Segura-Bedmar *et al.*[2] released an annotated corpus of DDIs (DrugDDI corpus) from the biomedical literature and organized the DDIExtraction2011 challenge.

In this article, we implement the all paths graph kernel [1] to extract DDIs from the DrugDDI corpus. We also test the all paths graph kernel approach on in-house corpus that has annotations of pharmacokinetic DDIs from MEDLINE abstracts.

The paper is organized as follows, section 2.1 and 2.2 describe the datasets, section 2.3 describes the all paths graph kernel approach and section 3 describes the results.

2 Methodology

DrugDDI Corpus We used the unified format [14] of the DrugDDI corpus [2] of the DrugDDI corpus. Detailed description of the corpus can be found at DrugDDI Corpus.

2.1 Clinical *in-vivo* pharmacokinetic DDI corpus

Our research group has been studying clinical DDIs reported in biomedical literature (MEDLINE abstracts) and extraction of numerical pharmacokinetic (PK) data from them[15]. During this process, we have collected MEDLINE abstracts that contain clinical PK DDIs, and further develop them into a PK DDI corpus. We decided that the ultimate goal of this task is extraction of DDIs from biomedical literature and it will be interesting to use this corpus as an additional resource. This corpus comprises of 219 MEDLINE abstracts which contains one or more of PK DDIs in same sentences. Here we call it PK-DDI corpus. Please note that a PK DDI means that one drugs exposure is changed by the co-administration of the other drug. As DrugDDI corpus focuses mainly on DDIs that change drug effects, our PK-DDI corpus is a good complementary source. In order to identify drugs in our PK-DDI corpus, we developed a dictionary based tagging approach using all the drug name entries in DrugBank [16]. The corpus was converted into the unified format as proposed in [14]. The DDI instances

were annotated based on guidelines from in-house experts. We split the corpus into training (80%) and testing (20%) fractions. This corpus will also be made public on the lines of the DrugDDI corpus. There are 825 true DDI pairs present in our corpus.

2.2 All paths graph kernel

We implemented the approach described by Airola *et al.* [1] for DDI extraction. This approach centers around the drugs, where a graph representation of the sentence is generated. Sentences are described as dependency graphs with interacting components (drugs). The dependency graph is composed of two unconnected sub-graphs: i) One sub-graph explores the dependency structure of the sentence; ii) the other explores the linear order of the words in the sentence. We used the Stanford parser [17]to generate the dependency graphs for both corpora. IIn the dependency graph, the shortest path between two entities was given higher weight as compared to other edges, this is because the shortest path contains important keywords which are indicative of interaction between two entities. In the linear sub-graph, all the edges have the same weight and the order in which words occur before, in the middle, or after drug mentions was considered. The all paths graph kernel algorithm [18] was subsequently implemented to compute the similarity between the graphical representations of the sentences. In particular, all paths graph kernels will be generated for tagger positive DDI sentences and negative DDI sentences. We then used Support Vector Machines (SVM) for classification. More details about the all paths graph kernel algorithm can be found in [1]. A pictorial representation of the approach is presented in figure 1.

3 Results

In this study we used an in-house corpus in addition to the DrugDDI corpus; both corpora contain training and testing subsets. We generated DDI extraction models based on both the training datasets individually and combined, and evaluated the performance of the DDI extraction models on the respective testing datasets.

Table 1 illustrates the summary of training and testing data in two corpuses. For the purpose of evaluation we used precision, recall and the balanced F-Score measure. We also performed 10-fold cross-validation during the training phase.

Table 2 displays the DDI extraction performance on DDI-PK corpus testing data. It suggests that using the DDI-PK corpus training data either with or without the DrugDDI corpus training data, led to the precision above 0.78 and recall above 0.64. On the other hand, if only the DrugDDI corpus was used, both precision and recall were around 0.41.

Table 3 displays the DDI extraction performance on DrugDDI corpus testing data. It suggests that all there models had similar perform in F-score, which was between 0.13 and 0.16, although using DDI PK corpus generated slightly better F-score than the other two approaches.

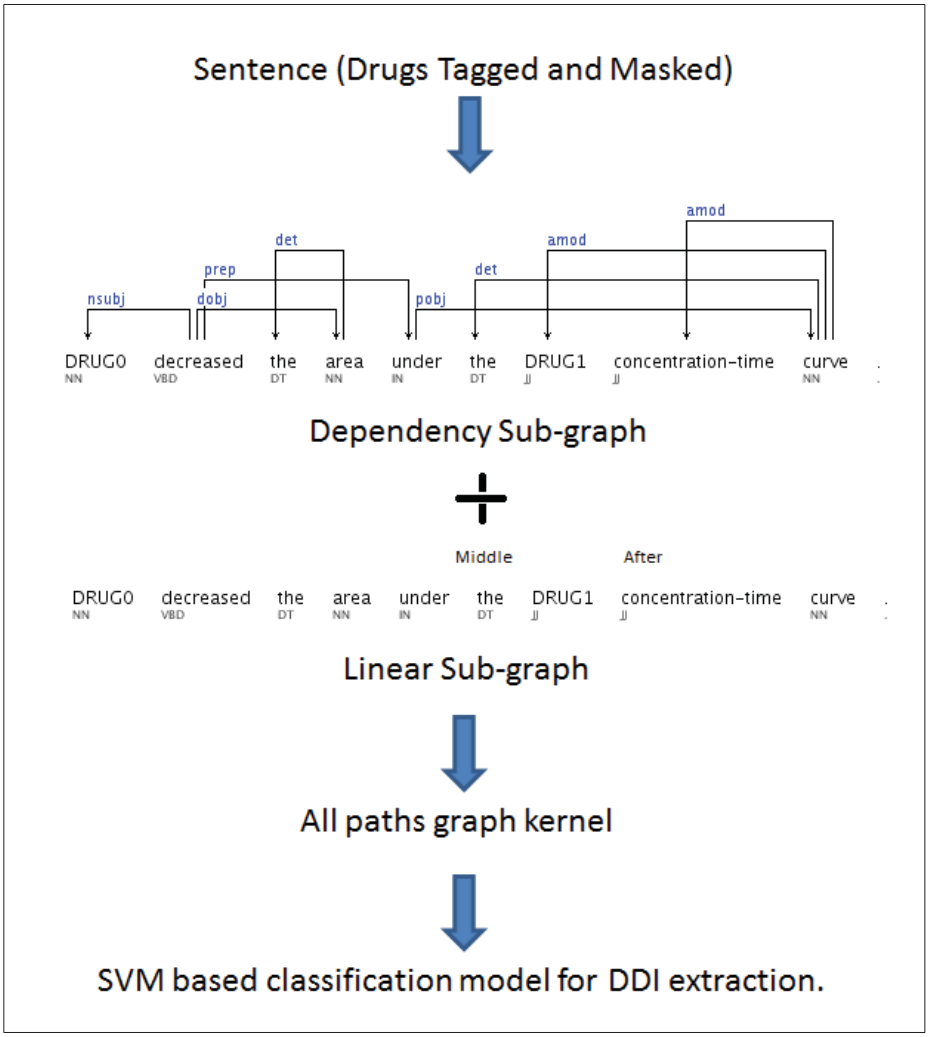


Fig. 1. Description of the methodology

Dataset	Number of sentences	Number of DDI Pairs
PK DDI Corpus (Train)	1939	2411
PK DDI Corpus (Test)	498	606
DDI Corpus (Train)	3627	20888
DDI Corpus (Test)	1539	7026

Table 1. Summary of the corpora used in this study

Dataset	F-Score	Precision	Recall
PK DDI Corpus (Train) + DDI Corpus (Train)	0.64	0.53	0.8
PK DDI Corpus (Train)	0.658	0.567	0.7857
DDI Corpus (Train)	0.415	0.417	0.414

Table 2. Performance of the different models on PK DDI Corpus (Testing dataset)

Dataset	F-Score	Precision	Recall
PK DDI Corpus (Full) + DDI Corpus (Train)	0.1346	0.1250	0.1457
PK DDI Corpus (Full)	0.1605	0.1170	0.2556
DDI Corpus (Train)	0.1392	0.1187	0.1682

Table 3. Performance of the different models on DrugDDI corpus test data

4 Discussion and Conclusion

There is large room for improvement in the DDI extraction from the biomedical literature. We also learned that the in-house DDI PK corpus and Drug DDI corpus have different DDI structures. It seems the all paths graph kernel method performed better in DDI PK corpus than the Drug DDI corpus.

The apparent low precision and recall in the Drug DDI corpus may result from the fact that the number of real DDIs is much less than the number of false DDIs in both corpus, but a comparison with the results of other teams is forthcoming once those get released. It is also possible that the weights on the sub-graph need to be further adjusted to get a better performance. We noticed a marked performance difference between the training corpora. The sentences in the DrugDDI corpus were long and complex. On the other hand, our DDI PK corpus has a simply sentence structure, and there is an average of one to two DDI pairs per abstract. Even with the same algorithm, these major differences between two corpora resulted in different DDI extraction performances.

DrugDDI corpus focuses on DDIs that affect the clinical outcomes (i.e. pharmacodynamics DDI); while PK DDI corpus focuses on DDIs that change the drug exposure. They are complementary to each other. Therefore, our work enriches the set of resources and analysis available to this community.

References

1. Airola, A., Pyysalo, S., Bjorne, J., Pahikkala, T., Ginter, F., Salakoski, T.: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics **9**(Suppl 11) (2008) S2
2. Segura-Bedmar, I., Martnez, P., de Pablo-Snchez, C.: Using a shallow linguistic kernel for drug-drug interaction extraction. Journal of Biomedical Informatics **In Press, Corrected Proof** (2011) –

3. Zhou, S., Yung Chan, S., Cher Goh, B., Chan, E., Duan, W., Huang, M., McLeod, H.L.: Mechanism-based inhibition of cytochrome P450 3A4 by therapeutic drugs. Clinical Pharmacokinetics **44**(3) (2005) 279–304
4. Cupp, M., Tracy, T.: Cytochrome P450: New nomenclature and clinical implications. American Family Physician **57**(1) (1998) 107
5. Lin, J., Lu, A.: Inhibition and induction of cytochrome P450 and the clinical implications. Clinical pharmacokinetics **35**(5) (1998) 361–390
6. Sabers, A., Gram, L.: Newer anticonvulsants: Comparative review of drug interactions and adverse effects. Drugs **60**(1) (2000) 23–33
7. Hirschman, L., Yeh, A., Blaschke, C., Valencia, A.: Overview of biocreative: critical assessment of information extraction for biology. BMC Bioinformatics **6**(Suppl 1) (2005) S1
8. 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: American Medical Informatics Association (AMIA) Annual Symposium proceedings, American Medical Informatics Association (2005) 216–220
9. Tari, L., Anwar, S., Liang, S., Cai, J., Baral, C.: Discovering drug-drug interactions: A text-mining and reasoning approach based on properties of drug metabolism. Bioinformatics **26**(18) (2010) i547–i553
10. Segura-Bedmar, I., Martinez, P., de Pablo-Sanchez, C.: A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. BMC Bioinformatics **12**(Suppl 2) (2011) S1
11. Yeh, A., Morgan, A., Colosimo, M., Hirschman, L.: Biocreative task 1a: gene mention finding evaluation. BMC Bioinformatics **6**(Suppl 1) (2005) S2
12. Hirschman, L., Colosimo, M., Morgan, A., Yeh, A.: Overview of biocreative task 1b: normalized gene lists. BMC Bioinformatics **6**(Suppl 1) (2005) S11
13. Blaschke, C., Leon, E., Krallinger, M., Valencia, A.: Evaluation of biocreative assessment of task 2. BMC Bioinformatics **6**(Suppl 1) (2005) S16
14. Pyysalo, S., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics **9**(Suppl 3) (2008) S6
15. Wang, Z., Kim, S., Quinney, S.K., Guo, Y., Hall, S.D., Rocha, L.M., Li, L.: Literature mining on pharmacokinetics numerical data: A feasibility study. Journal of Biomedical Informatics **42** (2009) 726–735
16. Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A.C., Wishart, D.S.: DrugBank 3.0: A comprehensive resource for 'Omics' research on drugs. Nucleic Acids Research (2010)
17. De Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. Volume 6., Citeseer (2006) 449–454
18. Gartner, T., Flach, P., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Learning theory and Kernel machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003: proceedings, Springer Verlag (2003) 129