SCAI: Extracting drug-drug interactions using a rich feature vector

Tamara Bobić[1,2], **Juliane Fluck**[1], **Martin Hofmann-Apitius**[1,2]

[1]Fraunhofer SCAI
Schloss Birlinghoven
53754 Sankt Augustin
Germany

[2]B-IT, Bonn Universität
Dahlmannstraße 2
53113 Bonn
Germany

{tbobic, jfluck, hofmann-apitius}@scai.fraunhofer.de

Abstract

Automatic relation extraction provides great support for scientists and database curators in dealing with the extensive amount of biomedical textual data. The DDIExtraction 2013 challenge poses the task of detecting drug-drug interactions and further categorizing them into one of the four relation classes. We present our machine learning system which utilizes lexical, syntactical and semantic based feature sets. Resampling, balancing and ensemble learning experiments are performed to infer the best configuration. For general drug-drug relation extraction, the system achieves 70.4% in F_1 score.

1 Introduction

Drug-drug interactions (DDI) describe possible interference between pharmacological substances and are of critical importance in drug development and administration (August et al., 1997). A drug may alter the metabolism of another, thus causing an enhanced, reduced or even toxic effect in certain medical treatments. For example: *"Fluvoxamine inhibits the CYP2C9 catalyzed biotransformation of tolbutamide."* Automated extraction of DDI from biomedical literature allows for a more efficient maintenance of the drug knowledge databases and is beneficial for patients, health care professionals and the pharmaceutical industry.

Having in mind their biomedical importance, the objective of the first DDIExtraction challenge[1] in 2011 was to motivate the development and to evaluate the automatic relation extraction (RE) systems for DDI. Given annotated drug entities, the participants addressed the task of identifying undirected binary relations among them. The knowledge extraction was performed on the sentence level and the best system achieved 65.74% F_1 score (Thomas et al., 2011a).

The 2013 DDIExtraction challenge[2] (organized as Task 9 of SemEval 2013 (Segura-Bedmar et al., 2013)) is based on a similar task definition, but additionally includes the disambiguation between four types of interaction: *mechanism*, *effect*, *advise* and *int*. The evaluation of participating systems is twofold, *i. e.* partial and strict. Partial evaluation considers that a prediction is correct when the pair label matches the gold annotation, while strict evaluation requires also a correct relation type to be assigned. The train and test corpora were generated from textual resources of DrugBank (Knox et al., 2011) database and MedLine[3] abstracts, dealing with the topic of DDI.

In the following sections we describe our supervised machine learning based approach for the extraction of DDI, using a rich feature vector (see Section 2.1). The base system employed LibLINEAR classifier, generating the first run submitted to the DDIExtraction challenge. Configurations coming from the two ensemble strategies (Section 2.2) produced the remaining prediction runs. Furthermore, we experimentally investigated the impact of train

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

[2] http://www.cs.york.ac.uk/semeval-2013/task9/

[3] http://www.ncbi.nlm.nih.gov/pubmed/

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 675–683, Atlanta, Georgia, June 14-15, 2013.

corpora imbalance on DDI detection through resampling strategies (Section 2.3). Finally, relation type disambiguation methodology is presented in Section 2.4.

2 Methods

We formulate the task of relation extraction as feature-based classification of co-occurring entities in a sentence. A sentence with n entities contains at most $\binom{n}{2}$ interacting pairs. For entity pairs that the classifier detects as "true", a post-processing step is performed where one of the four relation types is assigned, depending on the identified type-specific trigger words.

2.1 Features

To improve generalization of lexical information Porter stemming algorithm (Porter, 1980) was applied. All entities present in the sentence, which were not a part of the investigated pair, are renamed to a common neutral name (entity blinding).

For the generation of dependency-based features, sentences in the provided corpora were parsed using Charniak-Lease parser (Lease and Charniak, 2005; Thomas et al., 2011b). The resulting constituent parse trees were converted into Stanford dependency graphs (Marneffe et al., 2006). Following the idea of Thomas et al. (2011b), similar relations are treated equally by using their common parent type (unification of dependency types). An example is generalizing relations "subj", "nsubj" and "csubj" to a parent relation "subj".

In the following subsections the three groups of features (lexical, syntactical and semantic) with their corresponding members are described. Table 1 gives a more structured overview of the feature vector, organized by type. It should be noted that the listed features are used for the generation of all three prediction sets submitted to the DDI challenge.

2.1.1 Lexical features

Lexical features capture the token information around the inspected entity pair (EP). The sentence text is divided into three parts: text between the EP, text before the EP (left from the first entity) and text after the EP (right from the second entity). It has been observed that much of the relation information can be extracted by only considering these three contexts (Bunescu and Mooney, 2005b; Giuliano et al., 2006).

The majority of features are n-grams based, with $n \in \{1, 2, 3\}$. They encompass a narrow (window=3) and wide (window=10) surrounding context, along with the area between the entities. Additionally, combinations of the tokens from the three areas is considered, thus forming before-between, between-after and before-after conjunct features (narrow context).

2.1.2 Syntactic/Dependency features

Vertices (*v*) in the dependency graph are analyzed from a lexical (stemmed token text) and syntactical (POS tag) perspective, while the edges (*e*) are included using the grammatical relation they represent.

The majority of dependency-based features are constructed using the properties of edges and vertices along the shortest path (SP) of an entity pair. The shortest path subtree is conceived to encode grammatical relations with highest information content for a specific EP (Bunescu and Mooney, 2005a).

Similarly to lexical features, n-grams of vertices (edges) along the SP are captured. Furthermore, alternating sequences of vertices and edges (*v-walks* and *e-walks*) of length 3 are accounted for, following previous work (Kim et al., 2010; Miwa et al., 2010).

Apart from the SP-related features, incorporating information about the entities' parents and their common ancestor in the dependency graph is also beneficial. The lexical and syntactical properties of these vertices are encoded, along with the grammatical relations on the path from the entities to their common ancestor.

2.1.3 Semantic features

Semantic group of features deals with understanding and meaning of the context in which a particular entity pair appears.

A feature that accounts for hypothetical statements was introduced in order to reduce the number of false positives (phrases that indicate investigation in progress, but not actual facts). Negation (*e. g.* "not") detected close to the entity pair (narrow context) along with a check whether entities in the

pair refer to the same real-word object (abbreviation or a repetition) represent features which also contribute to the reduction of false positive predictions.

Drug entities in the corpora were annotated with one of four classes (drug, drug_n, brand, group), which provided another layer of relation information for the classifier. Prior knowledge about true DDI coming from the train corpora is used as a feature, if a previously known EP is observed in the test data. Presence of other entities (which are not part of the inspected EP) in the sentence text is captured, together with their position relative to the EP.

Finally, mentions of general trigger (interaction) terms are checked in all three context areas. Moreover, interaction phrases specific to a certain DDI type (see Section 2.4) are accounted for.

2.2 Ensemble learning

Combining different machine learning algorithms was proposed as a direction for improvement of the classification accuracy (Bauer and Kohavi, 1999).

A synthesis of predictions using LibLINEAR, Naïve Bayes and Voting Perceptron classifiers is an attempt to approach and learn the relation information from different angles with a goal of increasing the system's performance. The three base models included in the ensemble are employed through their WEKA[4] (Hall et al., 2009) implementation with default parameter values and trained on the full feature vector described in Section 2.1.

LibLINEAR (Fan et al., 2008) is a linear support vector machine classifier, which has shown high performance (in runtime as well as model accuracy) on large and sparse data sets. Support vector machines (SVM, Cortes and Vapnik (1995)) have gained a lot of popularity in the past decade and very often are state-of-the-art approach for text mining challenges.

Naïve Bayes (Domingos and Pazzani, 1996) is a simple form of Bayesian networks which relies on the assumption that every feature is independent from all other features. Despite their naive design and apparently oversimplified assumptions, Naïve Bayes can often outperform more sophisticated classification methods and has worked quite well in many complex real-world situations. Furthermore, it can be robust to noise features and is quite insensitive to stratification (Provost, 2000), which is of high value in class imbalance scenarios.

[4] http://www.cs.waikato.ac.nz/ml/weka/

Corpus	Pos	Neg	Total
MedLine	232 (0.13)	1,555 (0.87)	1,787
DrugBank	3,788 (0.15)	22,217 (0.85)	26,005

Table 2: Ratio of positive and negative instances in the DrugBank and MedLine train corpora.

Voting Perceptron (Freund and Schapire, 1999) combines a series of perceptrons, which are linear classification algorithms that process elements in the train set one at a time ("online"). The system stores the number of iterations the perceptron "survives", *i. e.* when the training set instances are classified correctly. The obtained count represents a weight used for combining the prediction vectors by a weighted majority vote.

In the ensemble learning scenario we consider two strategies that aim at increasing the system's performance by either favoring precision or recall:

1. *"majority"* – a pair represents true relation only if majority of the classifiers support that claim
2. *"union"* – a pair represents true relation if at least one of the classifiers supports that claim

2.3 Train corpora imbalance

Analysis of the basic train corpora statistics reveals an unequal ratio of positive and negative instances, *i. e.* under-representation of true interacting pairs (see Table 2). Class distribution imbalance often causes machine learning algorithms to perform poorly on the minority class (Hulse et al., 2007), thus, in this case, affecting the recall of true relations.

In order to explore the sensitivity of our system to the positive/negative ratio, we performed random undersampling of the data, artificially obtaining a desirable ratio (50-50). All positive instances in the dataset were kept, while the same number of negative instances were randomly chosen. The reverse approach of oversampling was considered, but given the ample train data provided by the organizers, such strategy could pose run-time challenges.

The experimental setting is described as follows. MedLine and DrugBank train corpora were divided further into train (exp-train) and test (exp-test) sets,

	Feature
Lexical	1. n-grams of tokens between the EP
	2. n-grams of tokens before the EP (narrow context, window = 3)
	3. n-grams of tokens after the EP (narrow context, window = 3)
	4. n-grams of tokens before the EP (wide context, window = 10)
	5. n-grams of tokens after the EP (wide context, window = 10)
	6. conjucted positions: before-between, between-after and before-after
Syntactical / Dependency	7. dependency n-grams on the SP
	8. syntactical n-grams on the SP
	9. lexical n-grams on the SP
	10. lexical and syntactical *e-walks*
	11. lexical and syntactical *v-walks*
	12. SP length (number of edges)
	13. lexical and syntactical information of the entities' parents
	14. lexical and syntactical information of the entities' common ancestor
	15. dependency n-grams from both entities to their common ancestor
	16. common ancestor represents a verb or a noun
Semantic	17. hypothetical context
	18. negation close to the EP
	19. entities refer to the same object
	20. type of entities that form the EP
	21. prior knowledge (from the train data)
	22. other entities present close to the EP
	23. DDI trigger words (general)
	24. DDI types trigger words (specific)

Table 1: Overview of features used, stratified into groups. EP denotes an entity pair, SP represent the shortest path.

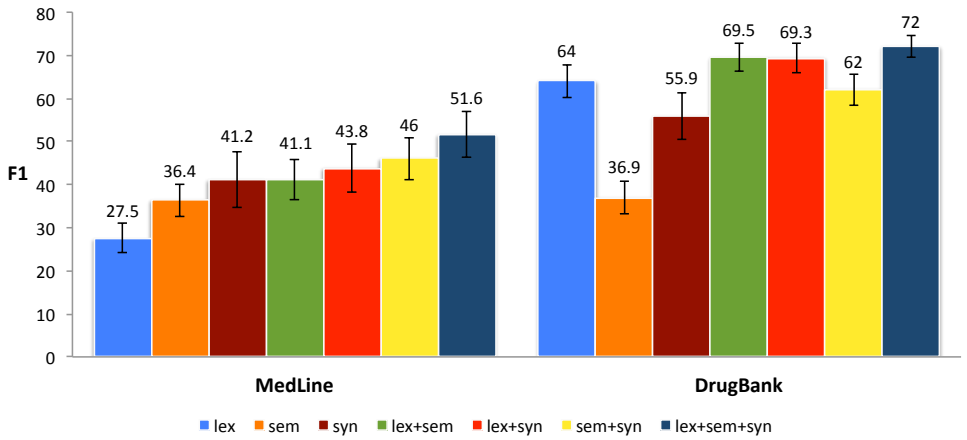

Figure 1: Contribution of individual feature sets and their combinations to the system's performance, evaluated by 10-fold cross-validation on the train corpora. *Lex* is an abbreviation for lexical, *sem* for semantic and *syn* for syntactical features.

Corpus	Exp-train pairs	Exp-test pairs
MedLine	1,259 (70.4%)	528 (29.6%)
DrugBank	18,148 (69.8%)	7,857 (30.2%)

Table 3: Experimental train and test subsets derived from the MedLine and DrugBank train corpora.

Relation	MedLine	DrugBank
mechanism	62 (0.27)	1257 (0.33)
effect	152 (0.66)	1535 (0.41)
advise	8 (0.03)	818 (0.21)
int	10 (0.04)	178 (0.05)

Table 4: The number of positive pairs for different DDI types in the train corpora. Ratios are given in brackets.

with an approximate ratio of 70-30. Instances from a particular document were always sampled to the same subset, in order to avoid information leakage. Table 3 gives an overview of the number of entity pairs each set comprises. The exp-train corpora were used for training the model in an original (full-size) and balanced (subsample) scenario, evaluated on the exp-test sets.

It should be noted that undersampling experiments were performed on the train corpora in order to inspect the impact of data imbalance on our system (results shown in Section 3.4). However, due to the challenge limitation of submitting only three runs, this configuration was ignored in favor of utilizing the complete train corpora.

2.4 Relation type assignment

The DDIExtraction challenge guidelines specify four classes of relations: *advise, mechanism, effect* and *int*. Table 4 illustrates the ratio of positive pairs assigned to each type in MedLine and DrugBank train corpora.

In Section 2.4.1, a brief outlook on the interaction type characteristics is given, along with some of the most common relation (trigger) phrases specific to them. Section 2.4.2 explains the methodology behind the process of relation type assignment.

2.4.1 Relations overview

Advise pertains to recommendations regarding co-administration of two or more drugs. Sentences describing these relations usually contain words such as: should, recommended, advisable, caution, avoid etc., as seen in the following examples:

- ***Barbiturates*** *and* ***glutethimide*** *should not be administered to patients receiving* ***coumarin*** *drugs.*
- *Concurrent therapy with* ***ORENCIA*** *and* ***TNF antagonists*** *is not recommended.*
- *The co-administration of* ***Fluvoxamine*** *Tablets and* ***diazepam*** *is generally not advisable.*

Effect is a relation type describing the signs or symptoms linked to the DDI, including the pharmacodynamic effect, *i. e.* mechanism of interaction. Some of the phrases often found to denote this type of relation are: effect, cause, decrease, increase, inhibit, activate, modulate etc. The following examples present expressions of an *effect* relation:

- *Pretreatment of megakaryocytes with extracellular* ***RR*** *(50 microM) also inhibited* ***InsP(3)****-induced responses.*
- *It is concluded that* ***neurotensin*** *modulates in an opposite way the function of the enkephalinergic neurons and the central action of* ***tuftsin****.*
- ***Diazepam*** *at doses of 0.25 mg/kg and 2.5 mg/kg injected with* ***morphine*** *was found to decrease the antinociceptive effect of* ***morphine****.*

Mechanism illustrates a more detailed description of the observed pharmacokinetic changes that includes biochemical information about metabolism, absorption, biotransformation, excretion etc. *Mechanism* relations often include mentions of *effect*-related interaction phrases, but provide an additional knowledge layer by addressing more complex biological concepts:

- ***Cholestyramine****, an anionic-binding resin, has a considerable effect in lowering the rate and extent of* ***fluvastatin*** *bioavailability.*
- *Additional* ***iron*** *significantly inhibited the absorption of* ***cobalt*** *in both dietary* ***cobalt*** *treatments.*
- ***Macrolide antibiotics*** *inhibit the metabolism of* ***HMG-CoA reductase inhibitors*** *that are metabolized by CYP3A4.*

Int relation implies sentences which only state that an interaction occurs, without providing much additional information about it. Trigger phrases that can be found in such sentences are usually limited to different lexical forms of "interaction":

- ***Rifampin*** *and* ***warfarin****: a drug interaction.*
- *In vitro interaction of* ***prostaglandin F2alpha*** *and* ***oxytocin*** *in placental vessels.*
- *Treatment with* ***antidepressant drugs*** *can directly interfere with blood glucose levels or may interact with* ***hypoglycemic agents****.*

2.4.2 Type disambiguation methodology

We approach the problem of relation type disambiguation as a post-processing step, utilizing identified (sentence level) trigger words as classification determinants. Precompiled relation trigger lists are generated by manual inspection of the train corpora, largely focusing on MedLine. The lists are specific to the four interaction types and non-overlapping.

Cases when a sentence contains trigger phrases from different relation classes are resolved by following a priority list:

1. *advise*
2. *mechanism*
3. *effect*
4. *int*

The rationale behind such priority assignment are the following observed patterns in the train corpora. Regardless of *effect* or *mechanism* connotation, if the sentence contains recommendation-like phrases (*e. g.* "should", "advisable"), it is almost always classified as an *advise*. Likewise, even though a relation might be describing an *effect*, if it contains a more detailed biochemical description, it is most likely representing *mechanism*. Finally, *effect* has advantage over *int* due to the simplicity of the *int* relation, along with the lowest observed frequency.

3 Results and Discussion

3.1 Baseline relation extraction performance

Performances of the submitted prediction runs are shown in Table 5, where the first row (run1) represents a system trained on the original (unbalanced) train corpora, using LibLINEAR classifier and a rich feature vector (see Section 2.1). The table offers results overview on MedLine, DrugBank and joined test corpora ("All"), using partial evaluation (general DDI detection).

The difference in performance on MedLine and Drugbank is apparent, measuring up to almost 25 percentage points (pp) in F_1 score (46.2% for MedLine and 71.1% for DrugBank). Due to a considerably larger size of the DrugBank corpus, overall results are greatly influenced by this corpus ($F_1 = 69.0\%$).

The results imply system's sensitivity towards class imbalance, which manifests in favored precision over recall. However, this discrepancy is much less observed on DrugBank test corpus. Despite the similarity in class ratio, DrugBank is a more compact and homogenous corpus, with a relatively unified writing style. Coming from a manually curated database, it has a rather standardized way of describing interactions, resulting in higher performance of the relation extraction system. MedLine corpora, however, are derived from different journals and research groups which gives rise to extremely diverse writing styles and a more challenging task for information extraction.

3.2 Features contribution

Figure 1 illustrates the performance of the LibLINEAR classifier, when all combinations of the three different feature sets are explored.

It can be observed that the highest performance is always achieved when all the features are included during training (*lex+syn+sem*), resulting in 51.6% and 72.0% F_1 score for 10-fold cross-validation on MedLine and DrugBank train corpora respectively.

Lexical features appear to be most useful for the DrugBank corpus, achieving 88.9% of the maximum performance when used solely. MedLine, on the other hand, benefits the most from syntactic features that reach 79.8% of the best result, compared to 53.3% with lexical features. Semantic group of features exhibits a uniform performance for both corpora, achieving 36.4% and 36.9% of F_1 score. Finally, grouping of two or all three feature sets is always beneficial and results in higher performance than the constituting base configurations.

Classifier	MedLine			DrugBank			All		
	P	R	F_1	P	R	F_1	P	R	F_1
run1: LibLINEAR	**68.8**	34.7	46.2	83.6	61.9	71.1	82.6	59.2	69.0
run2: Majority	68.6	25.3	36.9	**83.7**	61.7	71.0	**82.9**	58.1	68.3
run3: Union	43.1	**52.6**	**47.4**	79.6	**68.1**	**73.4**	74.8	**66.6**	**70.4**

Table 5: Results of the three submitted runs on the test corpora.

Classifier	DrugBank	MedLine
LibLinear	654	48
Naïve Bayes	854	88
V. Perceptron	608	30
Majority	693	35
Union	980	116

Table 6: Number of positive predictions on MedLine and DrugBank test corpora, using different configurations.

3.3 Ensemble experiments

Performance of the majority and union ensemble configurations on the test corpora is presented in Table 5. Table 6 gives an overview of the number of predicted positive pairs by the ensemble, as well as those by the individual base classifiers.

Voting Perceptron behaves similarly to LibLinear, while Naïve Bayes demonstrates insensitivity in terms of class imbalance, predicting the highest number of positive pairs for both MedLine and DrugBank test corpora.

Union voting strategy tends to overcome the limitations of poor recall, resulting in highest performance on all test corpora (47.4% for MedLine, 73.4% for DrugBank and 70.4% for All) among the three runs. The superior result is obtained by diminishing precision in favor or recall, which was shown as beneficial in these use-cases. However, the F_1 score difference is slight (1.2 pp, 2.3 pp and 1.4 pp), as compared to the baseline system (run1).

Predictions using the union ensemble ranked 3^{rd} in the general DDI extraction evaluation, achieving 5.5 pp and 9.6 pp of F_1 score less than the top two participating teams.

Train set	MedLine			DrugBank		
	P	R	F_1	P	R	F_1
original	48.4	39.6	43.6	75.1	62.4	68.2
balanced	37.2	70.4	48.7	60.8	72.7	66.2

Table 7: Comparison of results on the full train set and a balanced subsample, as evaluated on the MedLine and DrugBank train corpora.

3.4 Balanced training corpora

Table 7 presents relation extraction performance for training on a balanced subset, compared to the original unbalanced corpus.

In case of MedLine, an increase of around 5 pp in F_1 score can be observed for the balanced subsample. However, given a relatively high initial performance on DrugBank and the characteristics of that corpus, training on a subsample results in 2 pp reduced F_1 score. The raise of 30.8 pp in recall contributes greatly to the increased performance on MedLine, even though 11.2 pp of precision are lost. However, in case of DrugBank, a 10.3 pp increase in recall is not enough to compensate for the 14.3 pp loss in precision.

It can be observed that although undersampling approach removes information from the model training stage, the class balance plays a more significant role for the final performance.

3.5 Relation type disambiguation

Correct classification of interacting pairs into four defined classes was evaluated using macro and micro average measures.

While micro-averaged F_1 score is calculated by constructing a global contingency table and then calculating precision and recall, macro-averaged F_1 score is obtained by first calculating precision and recall for each relation type and then taking their

	MedLine			DrugBank			All		
	P	R	F_1	P	R	F_1	P	R	F_1
micro avg.	62.5	31.6	42.0	51.3	43.9	47.3	55.1	39.5	46.0
macro avg.	42.0	19.7	26.9	66.5	35.3	46.1	66.6	33.8	44.8
mechanism	70.0	29.2	41.2	58.0	39.2	46.8	53.2	39.1	45.0
effect	64.7	35.5	45.8	52.4	44.6	48.2	48.8	43.9	46.2
advise	18.2	28.6	22.2	50.7	65.0	57.0	50.5	63.3	56.2
int	0	0	0	100	1.1	2.1	100	1.0	2.1

Table 8: Results of DDI extraction when relation class detection is evaluated.

average (Segura-Bedmar et al., 2013). Therefore, macro average takes into consideration the relative frequency of each interaction class, while micro average treats all classes equally.

Table 8 shows an overview of performances for DDI extraction with relation class disambiguation, evaluated for each type separately, as well as cumulatively using micro and macro scores. For MedLine test corpus, the micro average F_1 score of 42% ranked 1^{st} among all participating systems. However, the macro average score is much lower, due to poor performance on *advise* and *int* relation classes and occupies 5^{th} position. Considering that our methodology gives advantage to relations which are observed more frequently, it is more adapted towards the micro measure.

The process of manually generating type-specific trigger lists was largely based on the MedLine train corpus due to its size, with the assumption that the relations in DrugBank are similarly expressed. However, both micro and macro scores for DrugBank ranked 7^{th}, showing that adaptation of trigger word lists needs to be done, depending on the target corpus.

In general, lower performance for relation class assignment is partially due to incompleteness of the trigger lists, but also coming intrinsically from the relation priority hierarchy. Most of classification errors occur when a trigger word belonging to a "higher" priority class is identified in the sentence. In the following example the word "should" implies *advise* relation, although *guanfacine* and *CNS-depressant drug* express an *effect* relation:

*The potential for increased sedation when **guanfacine** is given with other **CNS-depressant drug** should be appreciated.*

Another example is a sentence mentioning "effect", but actually describing a simple *int* relation:

***Chloral hydrate** and **methaqualone** interact pharmacologically with orally administered anticoagulant agents, but the effect is not clinically significant.*

Furthermore, a lot of missclassifications occur in sentences which contain pairs and triggers from different types, resulting in all relations being assigned to the highest identified type.

4 Conclusion

We present a machine learning based system for extraction of drug-drug interactions, using lexical, syntactic and semantic properties of the sentence text. The system achieves competitive performance for the general DDI extraction, albeit demonstrating sensitivity to the train corpora class imbalance. We show that, depending on the use case, resampling, balancing and ensemble strategies are successful in tuning the system to favor recall over precision. The post-processing step of relation type assignment achieves top ranked results for the MedLine corpus, however, needs more adaption in case of DrugBank. Future work includes a comparison with a multi-classifier approach, which circumvents the manual task of trigger list generation, supporting the fully automated scenario of relation extraction.

Acknowledgments

The authors would like to thank Roman Klinger for fruitful discussions. T. Bobić was funded by the Bonn-Aachen International Center for Information Technology (B-IT) Research School.

References

J.T. August, F. Murad, W. Anders, J.T. Coyle, and A.P. Li. 1997. *Drug-Drug Interactions: Scientific and Regulatory Perspectives: Scientific and Regulatory Perspectives*. Advances in pharmacology. Elsevier Science.

E. Bauer and R. Kohavi. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2).

R. C. Bunescu and R. J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *HLT and EMNLP*.

R. C. Bunescu and R. J. Mooney. 2005b. Subsequence Kernels for Relation Extraction. *NIPS*.

C. Cortes and V. Vapnik. 1995. Support vector networks. In *Machine Learning*.

P. Domingos and M. Pazzani. 1996. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *ICML*.

E. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Machine Learning Research*, 9.

Y. Freund and R. E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3).

C. Giuliano, A. Lavelli, and L. Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proc. of the 11st Conf. of the European Chapter of the Association for Computational Linguistics (EACL'06)*.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations*, 11.

J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano. 2007. Experimental perspectives on learning from imbalanced data. In *ICML*.

S. Kim, J. Yoon, J. Yang, and S. Park. 2010. Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, 11.

C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A. Chi Guo, and D.S Wishart. 2011. Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Res*, 39.

M. Lease and E. Charniak. 2005. Parsing biomedical literature. In *Proc. of IJCNLP'05*.

M. C. De Marneffe, B. Maccartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*.

M. Miwa, R. Saetre, J. D. Kim, and J. Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of bioinformatics and computational biology*, 8.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14.

F. Provost. 2000. Machine learning from imbalanced data sets 101 (extended abstract).

I. Segura-Bedmar, P. Martnez, and M. Herrero-Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.

P. Thomas, M. Neves, I. Solt, D. Tikk, and U. Leser. 2011a. Relation extraction for drug-drug interactions using ensemble learning. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*.

P. Thomas, S. Pietschmann, I. Solt, D. Tikk, and U. Leser. 2011b. Not all links are equal: Exploiting dependency types for the extraction of protein-protein interactions from text. In *Proceedings of BioNLP 2011 Workshop*.