UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge

Jari Björne, Suwisa Kaewphan and Tapio Salakoski
Turku Centre for Computer Science (TUCS)
Department of Information Technology
University of Turku
Joukahaisenkatu 3-5B, 20520 Turku, Finland
firstname.lastname@utu.fi

Abstract

The DDIExtraction 2013 task in the SemEval conference concerns the detection of drug names and statements of drug-drug interactions (DDI) from text. Extraction of DDIs is important for providing up-to-date knowledge on adverse interactions between co-administered drugs. We apply the machine learning based Turku Event Extraction System to both tasks. We evaluate three feature sets, syntactic features derived from deep parsing, enhanced optionally with features derived from DrugBank or from both DrugBank and MetaMap. TEES achieves F-scores of 60% for the drug name recognition task and 59% for the DDI extraction task.

1 Introduction

Drug-drug interactions (DDI) refer to one drug affecting the function of another when they are co-administered. These interactions are often adverse, frequently not well known and a source of potentially life-threatening unintended consequences for the patients. Databases such as DrugBank and Micromedex have been developed to store information about known DDIs, but at present their coverage remains limited and there can be inconsistencies in supplementary information (Knox et al., 2011; Wong et al., 2008). Text mining has been proposed as a solution for providing not only lists of DDIs but also a connection to the scientific evidence and supplementary information in the literature (Tari et al., 2010). Several groups of researchers are developing text-mining techniques to extract DDIs from literature and pharmaceutical documents (Tari et al., 2010; Segura-Bedmar et al., 2011a).

The DDIExtraction 2013 shared task concerns the detection of drug mentions and statements of DDIs from unannotated text (Segura-Bedmar et al., 2013). The first version of the DDIExtraction shared task was organized in 2011, with 10 teams participating from various universities (Segura-Bedmar et al., 2011b). The best result of 65.74% was achieved by team WBI of Humboldt University of Berlin (Thomas et al., 2011). University of Turku participated also in this task, placing 4th with an F-score of 62.99%, using the Turku Event Extraction System (Björne et al., 2011).

The Turku Event Extraction System (TEES)[1] is an open source program for extracting events and relations from biomedical texts. It was originally developed for extracting events in the BioNLP Shared Task scheme, and it models event extraction as a graph generation task, where keywords are nodes and the event arguments connecting them are edges. The system can be directly applied to pairwise relation extraction, representing relations as edges and the words they connect as nodes. The node detection system is somewhat similar to named entity recognition (NER) tools, and while quite flexible, can in many tasks exhibit lower performance and higher processing requirements than dedicated NER systems.

In the DDIExtraction 2013 task we apply the Turku Event Extraction system to detecting both drug name entities (task 9.1) as well as drug-drug interactions (task 9.2). We evaluate three different

[1] http://jbjorne.github.com/TEES/

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

feature sets for both tasks. As a baseline system deep syntactic parsing is used to generate large graph-based feature sets. For additional features, we test the impact of labeling examples with information from external sources. We test both the DrugBank Open Data Drug & Drug Target database (Knox et al., 2011) as well as the MetaMap tool to enrich the features derived from the corpus text.

MetaMap is a publicly available program developed at NLM for automatic mapping of texts to UMLS Metathesaurus concepts (Aronson, 2001). The UMLS Metathesaurus is an extensive repository of biomedical vocabularies that is derived from NLM databases and other external sources that contain information about biomedical concepts, synonyms and the relationship among them (Bodenreider, 2004).

The version of TEES used in the 2011 DDIExtraction task had been publicly available as an open source project since July 2012, but as small modifications were required for compatibility with the 2013 task, we published an updated 2.1 version that task participants could use. To simplify utilization of the numerous analyses TEES produces we also provided our drug-drug interaction predictions freely available for all DDIExtraction 2013 task participants in the hope of encouraging further participation in this interesting shared task.

We demonstrate that TEES has good performance for both drug name detection as well as drug-drug interaction detection, achieving an F-score of 60% in the drug name detection task 9.1 and an F-score of 59% in the drug-drug interaction detection task 9.2. We show that external information from DrugBank and MetaMap can considerably improve extraction performance, but observe that the use of such information must always depend on the exact requirements of each text mining task.

2 Methods

We present a unified approach to drug name and DDI extraction, utilizing largely the same machine learning approaches in both tasks. We develop three variants for tasks 9.1 & 9.2 each, testing the baseline performance of TEES for these tasks, as well as the impact of using external databases as additional training data.

2.1 Turku Event Extraction System

The Turku Event Extraction System is described in detail in Björne et al. (2012). Here we give a general overview about applying the system for the current task. TEES processes text in a pipeline of components, starting from preprocessing tasks such as NER and parsing and proceeding to the multiple, consecutive steps of event extraction. As tasks 9.1 and 9.2 are independent of each other the entity and interaction detection components of TEES are used independently, and for preprocessing, only the parsing is done (See Figure 1).

2.2 Training data preparation

TEES is a machine learning system based on support vector machines (SVM) (Tsochantaridis et al., 2005). To train the system for a new task, two datasets are required: a *training* set on which the SVM model is trained, and a *development* set on which the newly trained model is tested to determine parameter settings for optimal performance (See Figure 2). The optimal model can then be used to detect what it was trained for on unannotated datasets, such as the hidden shared task *test* set.

The DDIExtraction 2013 corpus consists of two parts: A training corpus used for system development and a test corpus for evaluating the participating systems. The annotation of the test corpus is not revealed to task participants. To develop the system, we estimate performance on the training corpus using 10-fold cross validation. To provide the datasets TEES requires, the training corpus is randomly divided (on the document level) into ten parts. For predicting drug names or DDIs for each part, seven of the remaining nine parts are used as a training set and two as a development set for parameter optimization. When producing the final models for classifying the test corpus, five parts of the training corpus are used for training and the other five for parameter optimization. In both cases, the parameter optimization set is merged with the training set when producing the final model for classifying the test set.

The DDIExtraction 2013 corpus is provided in an XML format originally introduced as a unified format for several pairwise protein-protein interaction (PPI) corpora (Pyysalo et al., 2008). TEES uses a variant of this format as its internal data representa-

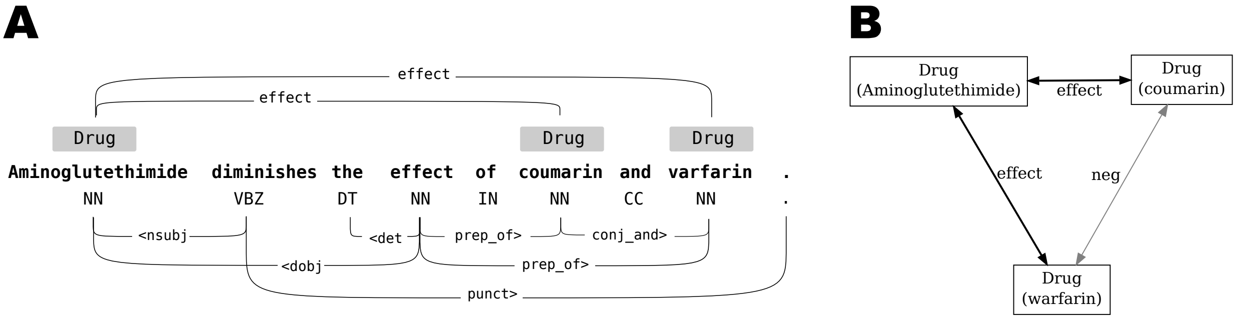

Figure 1: TEES graph representation for drug name and interaction extraction, with example sentence *DDI-DrugBank.d372.s2* from the DDIExtraction 2013 training corpus. A) Both the annotation (above the sentence) and the syntactic parse (below the sentence) are represented as graphs. Tokens form the nodes and dependencies the edges of the syntactic parse graph. Drug names form the nodes and DDIs the edges of the annotation graph. Drug name entities are linked to their syntactic head tokens, connecting the two graphs and allowing the parse to be used as a source of features. For DDI edges, most features are derived from the *shortest path of dependencies* connecting the two drug entities. B) For DDI extraction, one example is generated for each interaction type for each undirected pair of drug entities. The gray *neg* class edge is a negative example.

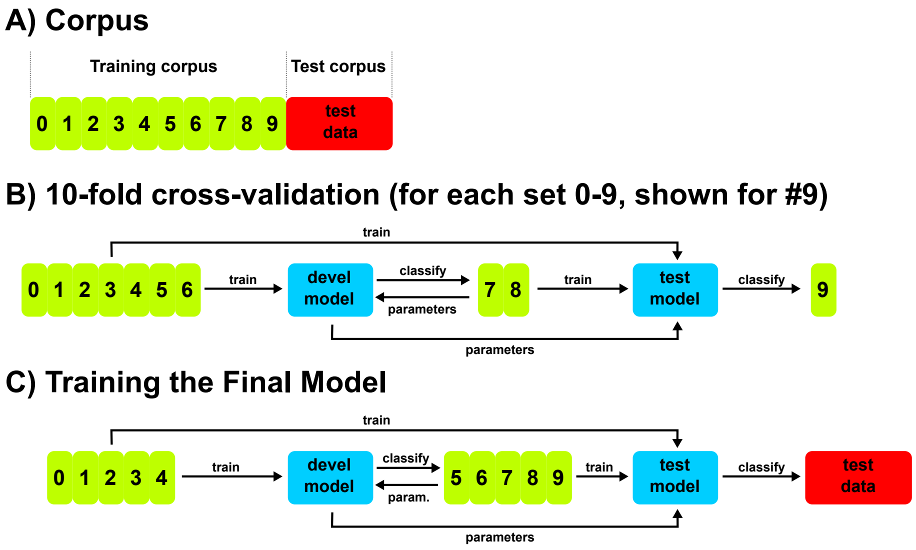

Figure 2: DDIExtraction 2013 corpus. A) To evaluate performance, and to provide analyses for the full training corpus, the training corpus is divided for 10-fold cross validation. B) Each of the ten parts is classified using seven of the remaining parts for training the model and the last two for optimizing parameters. After parameter optimization, all nine parts are used to train the model (with the optimal parameters) for classifying the test set. C) To classify the hidden DDIExtraction 2013 corpus half of the training corpus is used for training and the other half for determining optimal parameters. The test corpus is finally classified with a model trained using the full training corpus.

tion. While close to the DDIExtraction 2013 format, some differences exist, so we preprocess the corpora for compatibility with TEES. Namely, *ddi* elements are renamed as *interaction* elements, *entity* elements in task 9.2 are tagged with the *given* attribute to mark them as pre-annotated data for TEES and all character offsets are converted to the TEES format by increasing the end offset by one, resulting in spans denoted with the beginning character and end character plus one, a common convention in programming languages such as Java and Python.

Before use, all DDIExtraction 2013 corpora are parsed with the TEES preprocessing pipeline, using the BLLIP parser with David McClosky's biomodel to produce a Penn-tree style parse which is converted with the Stanford parser tools to the collapsed CC processed Stanford dependency scheme (Charniak and Johnson, 2005; McClosky, 2010; de Marneffe et al., 2006).

2.3 Drug name recognition with TEES

For drug name recognition the TEES entity detector module is used. Baseline syntactic features (model 1) are generated from the parse, using both information on the tokens and their linear context, as well as dependency chains starting from the entity head token. External data is added to the head token features, from where it is combined into more complex features. One example is generated for each token in the sentence, and these are classified into negatives or one of the positive classes.

As a new feature we generate all substrings starting from the first and last characters of the drug name, with the intention of detecting common prefixes and suffixes among the drug names.

2.4 Drug-drug interaction detection with TEES

For DDI extraction we use the TEES edge detector module. DDIs are typed, undirected edges, so one example is generated for each undirected pair of drug name entities present in the sentence (See Figure 1). The baseline syntactic features (model 1) are generated mostly from the *shortest path of dependencies* connecting the pair of drug name entities' head tokens. From this shortest path several feature groups are generated, including *N*-grams of various lengths, governor–dependent information for dependencies etc. External data is added into the two drug name entities, and combined into the path features.

We also use the TEES modification from DDIExtraction 2011 task where *conj_and* dependencies are ignored when calculating the shortest path, with the aim of including more of the relevant interaction words in the path.

2.5 Using DrugBank for Domain Knowledge

DrugBank[2] is a public database of information on drugs and drug targets. We use the downloadable XML version of the database.

For drug name recognition, for each candidate token, we add as features its presence as a known drug name in DrugBank and the *synonym*, *brand*, *group* and *category* annotations this drug may have. We also mark whether the candidate token exactly equals an annotation of one of these types, indicating cases where the token is e.g. a known brand name.

For DDI extraction, we mark as a feature whether the drug name pair is listed in DrugBank as having interactions or not. We also mark if one of the drug names is not listed in DrugBank.

2.6 Using MetaMap for Domain Knowledge

The MetaMap program has been used extensively for a wide array of BioNLP studies, such as automatic indexing of biomedical literature and concept-based text summarization (Reeve et al., 2007; Quanzhi and Yi-Fang Brook, 2006). For drug-related information extraction, two recent applications demonstrated that integrating the MetaMap program to their existing systems produces high overall performance in i.) identification and classification of the pharmaceutical substances and ii.) extraction of drug indication information (Segura-Bedmar et al., 2008; Fung et al., 2013).

MetaMap finds Metathesaurus concepts by performing a shallow syntactic analysis of the input text, producing a set of noun phrases. The noun phrases are then used to generate sets of variants which are consequently looked up from the Metathesaurus concepts. Matching concepts are evaluated against the original text and the strength of the mappings are calculated. The candidates are finally combined and the final scores are computed, where the highest score of a complete mapping represents

[2] `http://www.drugbank.ca/`

MetaMap's interpretation of the text.

The MetaMap program can be run both locally and remotely[3]. We ran the current version, MetaMap2012, remotely via the batch mode facility by converting the sentences of the DDIExtraction corpora into the MetaMap input format. Many of the applications that integrate MetaMap into their systems use the default settings that are claimed to be suitable for general purposes. However, we applied different options with the aim of increasing the coverage of Metathesaurus concepts found by MetaMap. The parameter set that influences the performance of MetaMap included; using a relaxed model, selecting the NLM2012AB Metathesaurus version, including all derivational variants, enabling unique acronym/abbreviation variants only, allowing candidates from one or two character words, preferring multiple concepts and using word sense disambiguation.

The Relaxed Model is provided by MetaMap in addition to the strict model which is offered as a default setting in which all types of filterings are applied. However, we chose the relaxed model in which only manual and lexical filterings are used. While the strict model is most appropriate for experiments that require the highest accuracy, it covers only 53% of the Metathesaurus strings. As we consider high coverage of concepts an important factor, we applied the relaxed model which consists of up to 83% of Metathesaurus strings.

The versions of Metathesaurus, Base, USAbase and NLM, provided with MetaMap are different in their Metathesaurus coverage and the license type required for using vocabulary sources. The NLM2012AB version which is offered at no cost for research purposes and covers all of the provided Metathesaurus was used in our work.

Variants, such as inflectional and derivational variants, are computed by MetaMap to account for the textual variation in the text. With this setting, many types of variants are generated recursively, and only acronyms and abbreviations are restricted to the unique ones. In addition, the candidates also include words that can be prepositions, conjunctions or determiners if they occur often enough in Metathesaurus.

Prefer multiple concepts causes MetaMap to score the mappings with more concepts higher than those with fewer concepts. This option is useful for discovering higher-order relationships among concepts found in the text and as such is assumed to be helpful for discovering the DDIs.

Word sense disambiguation attempts to solve lexical ambiguities by identifying the correct meaning of a word based on its context. By using this option in MetaMap, the program attempts to solve the ambiguities among equally scoring concepts by choosing the concept(s) based on semantic type.

We use the XML version of the MetaMap output which is post-processed by TEES to extract relevant features; candidate concepts, preferred concepts, CUI (Concepts Unique Identifier), score, semantic types and sources.

For drug name recognition, these are added as binary features for the candidate token, with the exception of the score, the value of which is normalized into the [0, 1] range. For DDI extraction, the binary features are added for the two drug names, and combined into the shortest path features.

2.7 Public analyses

The TEES 2.0 system used in DDIExtraction 2011 Shared Task has been public since summer 2012. While only small modifications are needed to make the DDIExtraction 2013 corpus usable with the TEES system, these can be complicated for new users. Therefore, to make sure our public DDIExtraction 2011 system is usable not only in theory, but easy enough to use in practice, we updated the system into the 2.1 version capable of automatically converting the DDIExtraction 2013 corpus and provided with precalculated models for DDI prediction.

To improve usability, we provided fully precalculated analysis files for the DDIExtraction 2013 corpus, produced using TEES 2.1. These analyses contain the TEES drug-drug interaction predictions, BLLIP Penn tree-bank style parses (using the McClosky biomodel), Stanford dependency parses (in the collapsed CC processed format) and syntactic head offsets for drug entities.

The analyses were calculated with the baseline TEES 2.1 system, without using the external datasets which were tested only later. The analyses were provided for task 9.2, which is the direct

[3] http://metamap.nlm.nih.gov/

continuation of the 2011 task for which the public TEES system was already available.

The analyses for the DDIExtraction 2013 corpus were made available on February 25th 2013. Despite being published quite late in the training period there was interest in this supporting data, and before the task result submission deadline the analyses were downloaded 14 times. The test set analyses were provided for registered DDIExtraction 2013 participants during the test period.

3 Results and Discussion

Three feature sets were used to produce the results. The baseline set (model 1) consisted of the TEES entity and edge detectors which build a large feature set from syntactic parses. Model 2 adds DrugBank features to this baseline and model 3 further extends model 2 with MetaMap information.

Three runs using these models were submitted for both tasks 9.1 and 9.2. The results indicate the system was capable of detecting both drug names and drug-drug interactions with reasonable performance. The best F-scores were 60% for task 9.1 drug name detection and 59% for task 9.2 DDI extraction.

As task 9.1 is completely new, and task 9.2 was extended from the 2011 DDI extraction task with typed interactions and MEDLINE abstracts, the current results are not directly comparable with the 2011 ones. The evaluation metric closest to the 2011 task is task 9.2 DDI detection regardless of type, using only the DrugBank subset of the corpus. With this metric, our system achieved an F-score of 72% in 2013 vs. 62.99% in 2011, which may indicate higher baseline performance, potentially influenced by a larger training dataset.

3.1 Drug name recognition

The decision to not attempt detection of more than one token per drug entity proved to be not too detrimental to the final performance. In the training corpus, there are 14,765 drug name entities of which only 2,768 (18.7%) consist of more than one token, and of these only 38 are disjoint (not forming a continuous span). For our best performing drug name detection model (number 3) typed, partial span matching was at 78% F-score vs. typed, strict span matching at 65%. Therefore, detecting only a single token per entity resulted in a maximum loss of 13 percentage points (pp), but considering that a scheme designed to detect multi-token entities would be inherently more complex, potentially having lower performance, and that not all of the spans would be correctly detected, we feel this tradeoff in performance is worth it for the considerably more simple system design it allows.

Adding the external datasets to the classifier models proved to have a considerable impact on the task performance (See Table 1). The baseline system reached an F-score of 47% which was increased by 9 percentage points when including DrugBank information and a further 4 percentage points when also MetaMap information was included.

As seen from the type-specific F-scores (on the training corpus), *brand* class entity detection was improved by 30 pp when DrugBank information was added, and increased slightly further with MetaMap information (See Table 2). DrugBank lists brand names for many drugs, and when this information is added as a feature for each detected drug, determining the type of the drug is greatly improved.

The official primary metric in both tasks 9.1 and 9.2 is a macro-averaged F-score, which gives equal weight to performance in each class, emphasizing the importance of detecting also the difficult, small classes. In particular, the class *drug_n* (active substances not approved for use in humans for medical purposes) was very difficult to detect for our system. While performance remained low for all three models, including the MetaMap information gave a large relative increase in *drug_n* detection performance, increasing it from 2% F-score to 8% (See Table 2). With the macro-averaged overall performance, this resulted in model three with the MetaMap information having notably higher performance.

We hypothesized that the *drug_n* category might be hard to detect as it could contain entities similar to the *drug* category, which may differ only by approval for use in humans, information that is not likely present in the corpus. Analysis of classification errors (See Table 3) confirms this hypothesis, showing that *drug_n* entities are by far the most commonly misclassified ones. Addition of DrugBank and MetaMap information considerably reduces *drug_n* misclassifications into the *drug* category.

M	task	P	R	F
1	9.1	0.48 (0.70)	0.46 (0.51)	0.47 (0.59)
2	9.1	0.6 (0.77)	0.52 (0.59)	0.56 (0.67)
3	9.1	0.69 (0.76)	0.54 (0.59)	0.6 (0.66)
1	9.2	0.73 (0.69)	0.47 (0.44)	0.57 (0.54)
2	9.2	0.76 (0.69)	0.48 (0.45)	0.59 (0.55)
3	9.2	0.73 (0.68)	0.48 (0.44)	0.58 (0.53)

Table 1: Official results for TEES in the DDIExtraction 2013 task and in parentheses corresponding 10-fold cross-validation results on the training corpus. The three models (M) used are 1) baseline syntactic features, 2) baseline with DrugBank features and 3) baseline with both DrugBank and MetaMap features.

model	drug	brand	group	drug_n
1	0.72	0.6	0.48	0.02
2	0.78	0.9	0.49	0.02
3	0.78	0.91	0.48	0.08

Table 2: Per-class micro-average scores for the drug name recognition task 9.1.

Task rules allowed using the test corpus of task 9.2 (with annotated entities) as additional training data for task 9.1. Due to time constraints we did not use it for training, but it is likely that performance could be further enhanced by using it.

3.2 Drug-drug interaction extraction

Performance of the three feature sets in the 9.2 DDI extraction task are much closer than in the 9.1 drug name recognition task. Still, additional information from DrugBank and MetaMap slightly increase performance, but DrugBank alone outperforms using both MetaMap and DrugBank. With the performance difference range between the models being only 2 pp, we think the results remain inconclusive.

That external data did not provide a further increase might indicate that drug-drug interaction detection is mostly a matter of interpreting the syntactic parse, whereas drug-name recognition benefits more from dictionary matching methods.

As with task 9.1, we analyse the classification errors on the 10-fold classification performed on the training dataset for which annotations are publicly available (See Table 4). None of the DDI classes are as hard to detect as the drug name class *drug_n*, but the *int* class has much lower performance than the other classes, with most examples classified incorrectly as negatives.

4 Conclusions

We applied the Turku Event Extraction System 2.1 to detection of both drug names and drug-drug interactions in the DDIExtraction 2013 task. The system showed good performance for both tasks, but we must consider that name and interaction detection were evaluated in isolation. In real world text mining tasks, these steps will be consecutive and as such result in lower overall performance. TEES achieves good performance using deep syntactic parsing, but this is a computationally expensive processing step. When drug names are detected with TEES, all input sentences need to be parsed, but if some other method is used for drug name recognition, TEES can parse just the sentences with drug names, as only they can potentially contain DDIs, enabling much faster DDI extraction.

We showed that adding external data from the DrugBank database and from MetaMap preprocessing can considerably increase extraction performance. However, we assume this makes the system more dependent on such data being available for candidate drug names and DDIs in the text being processed, potentially making it harder to detect completely new names and interactions. Therefore, using external data is likely to introduce a tradeoff of higher performance vs. wider detection. Use of such data should be chosen according to the task, as in some cases the goal is to retrieve documents with known drugs and interactions, in others to maximize detection of information not yet in the databases.

As with previous TEES versions, we will provide our source code freely available under an open source license at the TEES project repository[4]. We will also include a wrapper for using the MetaMap tool via the TEES preprocessing pipeline, allowing it to be easily integrated into event and relation extraction tasks.

Acknowledgments

We thank CSC — IT Center for Science Ltd, Espoo, Finland for providing computational resources.

[4] http://jbjorne.github.com/TEES/

	neg	**brand**	**drug_n**	**group**	**drug**
neg	**99.57** **99.60** **99.60**	0.04 0.03 0.03	0.00 0.00 0.01	0.15 0.14 0.14	0.24 0.22 0.22
brand	21.43 8.91 8.63	**67.92** **89.70** **89.98**	0.07 0.07 0.07	0.63 0.21 0.28	9.95 1.11 1.04
drug_n	49.70 63.27 65.27	2.79 0.00 0.00	**12.18** **15.37** **15.37**	0.40 1.00 1.20	34.93 20.36 18.16
group	13.80 14.13 14.04	0.12 0.00 0.06	0.03 0.03 0.06	**85.15** **84.97** **85.00**	0.90 0.87 0.84
drug	6.71 5.60 6.20	0.69 0.27 0.32	0.10 0.08 0.08	0.75 0.79 0.69	**91.75** **93.27** **92.72**

Table 3: Task 9.1 drug name classification errors for the training corpus. Each cell in the table lists from top to bottom results for models one to three (baseline, baseline+DrugBank, baseline+DrugBank+MetaMap). The results are percentage of SVM examples of each class (vertical) classified into each potential class (horizontal).

	neg	**int**	**advise**	**effect**	**mechanism**
neg	**97.27** **97.32** **97.40**	0.02 0.03 0.03	0.52 0.49 0.47	1.09 1.06 1.04	1.09 1.09 1.05
int	61.70 61.70 70.74	**22.87** **23.40** **19.15**	0.53 0.00 0.00	9.57 8.51 7.45	5.32 6.38 2.66
advise	34.50 34.02 33.54	0.12 0.24 0.24	**60.17** **60.05** **60.77**	4.24 4.36 4.36	0.97 1.33 1.09
effect	38.59 38.41 39.18	0.41 0.41 0.41	3.85 3.73 3.68	**54.06** **54.30** **53.59**	3.08 3.14 3.14
mechanism	50.34 48.75 52.16	0.15 0.15 0.23	2.05 1.82 1.29	5.08 5.08 5.00	**42.38** **44.20** **41.32**

Table 4: Task 9.2 drug-drug interaction classification errors for the training corpus. Each cell in the table lists from top to bottom results for models one to three (baseline, baseline+DrugBank, baseline+DrugBank+MetaMap). The results are percentage of SVM examples of each class (vertical) classified into each potential class (horizontal).

References

Alan R Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Jari Björne, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2011. Drug-drug interaction extraction from biomedical texts with SVM and RLS classifiers. In *Proc. of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) at SEPLN 2011*, volume 761, pages 35–42, Sept 5.

Jari Björne, Filip Ginter, and Tapio Salakoski. 2012. University of Turku in the BioNLP'11 Shared Task. *BMC Bioinformatics*, 13(Suppl 11):S4.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

Eugene Charniak and Mark Johnson. 2005. 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)*, pages 173–180. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454.

Kin Wah Fung, Chiang S Jao, and Dina Demner-Fushman. 2013. Extracting drug indication information from structured product labels using natural language processing. *Journal of the American Medical Informatics Association*.

Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, Yannick Djoumbou, Roman Eisner, Anchi Guo, and David S. Wishart. 2011. Drugbank 3.0: a comprehensive resource for omics research on drugs. *Nucleic Acids Research*, 39(Database-Issue):1035–1041.

David McClosky. 2010. *Any domain parsing: automatic domain adaptation for natural language parsing*. Ph.D. thesis, Department of Computer Science, Brown University.

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC bioinformatics*, 9(Suppl 3):S6.

Li Quanzhi and Wu Yi-Fang Brook. 2006. Identifying important concepts from medical documents. *Journal of Biomedical Informatics*, 39(6):668 – 679.

Lawrence H Reeve, Hyoil Han, and Ari D Brooks. 2007. The use of domain-specific concepts in biomedical text summarization. *Information Processing & Management*, 43(6):1765–1776.

Isabel Segura-Bedmar, Paloma Martínez, and María Segura-Bedmar. 2008. Drug name recognition and classification in biomedical texts: a case study outlining approaches underpinning automated systems. *Drug discovery today*, 13(17):816–823.

Isabel Segura-Bedmar, Paloma Martínez, and César de Pablo-Sánchez. 2011a. A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. *BMC bioinformatics*, 12(Suppl 2):S1.

Isabel Segura-Bedmar, Paloma Martínez, and Daniel Sánchez-Cisneros. 2011b. The 1st DDIExtraction-2011 challenge task: extraction of drug-drug interactions from biomedical texts. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011: 7 Sep 2011; Huelva, Spain*, pages 1–9.

Isabel Segura-Bedmar, Paloma Martínez, and Maria 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)*.

Luis Tari, Saadat Anwar, Shanshan Liang, James Cai, and Chitta Baral. 2010. Discovering drug–drug interactions: a text-mining and reasoning approach based on properties of drug metabolism. *Bioinformatics*, 26(18):i547–i553.

Philippe Thomas, Mariana Neves, Illés Solt, Domonkos Tikk, and Ulf Leser. 2011. Relation extraction for drug-drug interactions using ensemble learning. In *Proc. of the 1st Challenge task on Drug-Drug Interaction Extraction (DDIExtraction 2011) at SEPLN 2011*, page 11–18, Huelva, Spain, Sept 5.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453–1484.

Chen-May Wong, Yu Ko, and Alexandre Chan. 2008. Clinically significant drug-drug interactions between oral anticancer agents and nonanticancer agents: profiling and comparison of two drug compendia. *The Annals of pharmacotherapy*, 42(12):1737–1748.