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

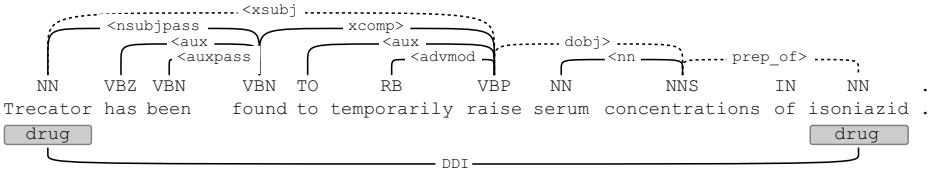

Fig. 1. A potential drug-drug interaction (DDI) can exist for each pair of drug entities in the sentence. This example sentence from the DrugDDI corpus training set has one positive interaction. The automatically generated syntactic deep dependency parse is shown above the sentence. Our system classifies 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

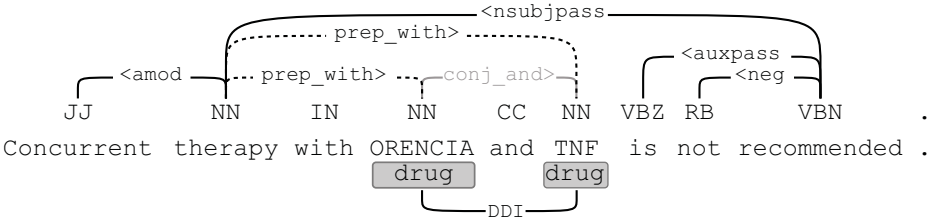

Fig. 2. Skipping the conj_and dependencies when determining the shortest path (dotted line) allows more tokens relevant for the potential interaction to be included in the path.

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*