ALGORITHMICS FOR DATA MINING
Third Deliverable ADM Project

Predicting Drug Interactions from biomedical texts

Author
Meysam Zamani Forooshani
meysam.zamani@est.fib.upc.edu

May 26, 2019

Abstract

Its a challenge of recognizing drug names, detecting drug-drug interactions and further categorizing them into one of the four classes. I present two machine learning systems. The NER System is resolved with use of CRF model and extensive features' set. The obtained F1 score is 64%. In the DDI system I apply two-stage SVM with data preprocessing. The F1 scores obtained for both stages are: 51%, 18% and the macro-average score: 20%.

1 Introduction

In my project I base on the challenge The DDIExtraction 2013, which is a continuation of the event taking place in 2011. The main goal of the first event was the detection of drug-drug interactions from biomedical texts. The new edition includes also a supporting task: the recognition and classification of pharmacological substances. DDIExtraction 2013 aims to address the extraction of DDIs as a whole, but to allow separate evaluation of the performance for different aspects of the problem was divided into two subtasks. In order to deal with different types of texts and language styles, the organizators decided to add the second part of data set: MedLine abstracts.(11)

2 DDI corpus

Following the paper of Segura-Bedmar, Martinez, Herrero-Zazo, data set could be described as follows. The DDI corpus consists of 1,017 texts (784 DrugBank texts and 233 MedLine abstracts) and was manually annotated with a total of 18 491 pharmacological substances and 5 021 drug-drug interactions.(11). Approximately 77% of the DDI corpus documents were randomly selected for the training data set. The remaining documents (142 DrugBank texts and 91 MedLine abstracts) was used for the test data set. The training data set is the same for both subtasks. The test data set for the NER task was formed by discarding documents which contained DDI annotations. Entity annotations were removed from this data set to be used by participants. The remaining documents (those containing some interaction) were used to create the test dataset for DDI task.(11)

3 Named Entity Recognition task

This task concerns the named entity extraction of pharmacological substances in text, what is a crucial first step for extraction information about drug-drug interactions.(11) In this task there are defined four types of pharmacological substances: drug (any chemical substance used in the treatment, cure, prevention or diagnosis approved for human use), brand (any drug that was first developed by a pharmaceutical company), group (any term in text designating a chemical or pharmacologic relationship among a group of drugs) and drug-n (any chemical substance affecting living organisms, but not approved for human use with the medical purpose). Other types of entities are out of scope of this project. I aim to prepare machine learning system, which recognizes and classifies these substances as belonging to one of these groups. One of the possibilities is basing on handcrafted rules - this approach is commonly used for simple cases. The second approach is based on discriminative model, which learns the rules from training corpus. The most used models are: Conditional Random Fields (CRF), Support Vector Machines (SVM) and Expectation Maximisation (EM). I decided to develop CRF system.

3.1 System description

In this section I focus on specification of designed system: data preprocessing, process of entity identification and classification, metrics used to com-

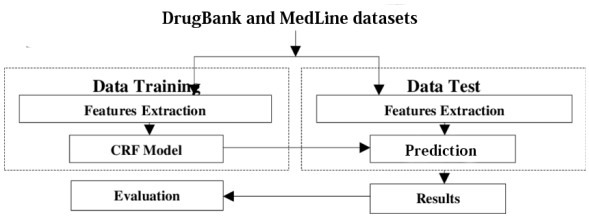

Figure 1: Architecture of proposed model

pare and evaluate the performance of the system. Model approach is presented in the Figure 1.

3.1.1 Data preprocessing

The XML files with which I was provided, do not demand on any modification in structure. In order to get more data to train the model, I decided to combine DrugBank and MedLine data sets. For the purpose of model training, I also decided to add another step of model validation by randomly dividing the training set in the ratio 80:20.

3.1.2 Entity identification process

In the process of entity extraction I obtained 42 features. 21 of them were included in initial corpus of the project. Below I mention all features added to the basic corpus of the project:

- length, lengthPrev, lengthNext: length of a token, a token preceding and a token following it
- pref4, pref4Prev, pref4Next: 3-letter prefix of the token, a token preceding and a token following it
- pref5, pref5Prev, pref5Next: 5-letter prefix of the token, a token preceding and a token following it
- hasDigit, hasDigitPrev, hasDigitNext: boolean indicating, if a token, a token preceding and a token following it consist on a digit
- hasDash, hasDashPrev, hasDashNext: boolean indicating, if a token, a token preceding and a token following it consist on a dash
- isAlpha, isAlphaPrev, isAlphaNext: boolean indicating, if a token, a token preceding and a token following it consist only alphabet letters (without numbers and special signs).

Adding the rules: hasDigit, pref4 and pref5 was inspired by system created by Grego, Pinto and Couto (5).

I decided also to apply part-of-speech tagger: a piece of software, that reads text and assigns parts of speech to each word and other token. In other words, the words of a sentence are translated into their contextually appropriate POS tags (3). The operation of the POS tagger can be described in the following way: The tagger first obtains the set of possible POS tags for each word from a lexicon and then disambiguates between them based on the word context. (10) Due to the long-term calculation of the POS tagging, I decided to apply it only to the top 8 sets of features.

3.1.3 Entity classification process

In order to classify entities, I applied Conditional Random Fields (CRF) model. Conditional Random Fields (CRFs) offer a unique combination of properties: discriminatively trained models for sequence segmentation and labeling; combination of arbitrary, overlapping and agglomerative observation features from both the past and future; efficient training and decoding based on dynamic programming; and parameter estimation guaranteed to find the global optimum. (6)
CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence. Also it assigns a well-defined probability distribution over possible labelings, trained by maximum likelihood. CRFs generalize easily to analogues of stochastic context-free grammars. It is the only model class that has guaranteed of global maximum likelihood convergence. (5) Conditional Random Fields can be trained using the exponential loss objective function used by the AdaBoost algorithm (4).
Typically, boosting is applied to classification problems with a small, fixed number of classes; applications of boosting to sequence labeling have treated each label as a separate classification problem (1). Another attractive aspect of CRFs is that it can implement efficient feature selection and feature induction algorithms for them. In order to evaluate the model performance, I used three metrics: precision, recall and F1. It was needed to use the following scoring categories proposed by MUC (Message Understanding Conference held by Defense Advanced Research Projects Agency DARPA):

- COR: the system's output and the gold-standard annotation agree
- INC: the system's output and the gold-standard annotation disagree
- PAR: the system's output and the gold-standard annotation not identical but containing some overlapping text
- MIS: a gold-standard entity not identified by the system
- SPU: an entity not existing in the gold-standard.

Then two measures should be counted: Possible and Actual. POSSIBLE(POS): the number of annotations in the gold-standard which contribute to the final score

$$POS = COR+INC+PAR+MIS = TP+FN$$

ACTUAL(ACT): the total number of annotations produced by the system

$$ACT = COR+INC+PAR+SPU = TP+FP$$

Precision is the percentage of named entities found by learning system that are correct. Formula for exact match is:

$$PREC = \frac{COR}{ACT} = \frac{TP}{TP+FP}$$

Formula for partial match is:

$$PREC = \frac{COR + 0.5 * PAR}{ACT}$$

Recall is the percentage of named entities present in the corpus that are found by the system. Formula for exact match is:

$$RECALL = \frac{COR}{POS} = \frac{TP}{TP+FN}$$

Formula for partial match is:

$$RECALL = \frac{COR + 0.5 * PAR}{POS}$$

F1 is the harmonic mean of precision and recall calculated in the following way:

$$F1 = \frac{2 * P * R}{P + R}$$

3.2 Results

I trained 88 models, using different set of features and including or not the POS tagging. When choosing the best model, I used the Macro Average F1 criterion. The features used in this iteration are all rules included in the basic code and additional rules: length, lengthPrev, lenghtNext, hasDigit, hasDash, isAlpha and POS taggers. I have trained the model on the entire initial train data set, and then I conducted the test using the test data set. I obtained the following results: Precision: 0.93, Recall: 0.65, F1: 0.72. The full set of obtained scores is presented in the table 1.

The model performs well for train set. Scores are satisfactory for almost all considered measures (I obtain lower score only for drug_n matching). However, during training the model on the entire train data set, I received lower classification results. Nevertheless, the final model still shows good performance in the classification of drugs, brands and groups - respectively, obtaining F1 with the values 0.85, 0.83, 0.80. The main reason for receiving lower average results is the low drug_n score, with recall of 0.03 and F1 0.06. The cause may be the similarity between the words determining drugs and drug_ns - the lack of clear differences between these groups results in a higher probability of improper classification of drug_n entity as a drug. To confirm that the set of rules I selected matches the data structure well, I checked other sets of rules on the main data set. As an example, training a model for a full set of rules, I received the final F1 equal to 0.62 with precision 0.89 and recall 0.55.

my reference point is a model containing a basic set of rules. In its case, the following results were obtained: precision: 0.92, recall: 0.51, F1: 0.58. The total set of results obtained in testing the basic model is presented in Table 2, compared to the final model.

For almost every measure I received better or the same results than the basic model. In particular, strict matching F1 increased from 0.69 to 0.71, exact matching F1 - from 0.77 to 0.81, exact drug matching F1 - from 0.83 to 0.85, exact brand matching from 0.60 to 0.83. Macro-average recall grown from 0.51 to 0.57 and F1 from 0.58 to 0.64. It can therefore be said that my model is more effective than the basic model, especially in the area of brand and drug classification.

Metric	Train score	Test score
Strict match		
Precision	0.91	0.78
Recall	0.83	0.65
F1	0.87	0.71
Exact match		
Precision	0.91	0.88
Recall	0.83	0.73
F1	0.87	0.80
Partial match		
Precision	0.94	0.88
Recall	0.87	0.75
F1	0.90	0.81
Type match		
Precision	0.94	0.81
Recall	0.85	0.67
F1	0.89	0.73
Exact drug match		
Precision	0.96	0.90
Recall	0.89	0.80
F1	0.93	0.85
Exact brand match		
Precision	0.95	1.00
Recall	0.76	0.71
F1	0.84	0.83
Exact group match		
Precision	0.89	0.86
Recall	0.80	0.75
F1	0.84	0.80
Exact drug_n match		
Precision	0.94	0.80
Recall	0.15	0.03
F1	0.25	0.06
Macro-average		
Precision	0.93	0.89
Recall	0.65	0.57
F1	0.72	0.64

Table 1: Scores obtained from final model tested on different data sets.

Metric	Basic	Final
Strict match		
Precision	0.78	0.78
Recall	0.62	0.65
F1	0.69	0.71
Exact match		
Precision	0.87	0.88
Recall	0.69	0.73
F1	0.77	0.8
Partial match		
Precision	0.87	0.88
Recall	0.71	0.75
F1	0.79	0.81
Type match		
Precision	0.80	0.81
Recall	0.64	0.67
F1	0.71	0.73
Exact drug match		
Precision	0.88	0.90
Recall	0.78	0.80
F1	0.83	0.85
Exact brand match		
Precision	0.93	1.00
Recall	0.44	0.71
F1	0.60	0.83
Exact group match		
Precision	0.88	0.86
Recall	0.77	0.75
F1	0.82	0.80
Exact drug_n match		
Precision	1.00	0.80
Recall	0.04	0.03
F1	0.08	0.06
Macro-average		
Precision	0.92	0.89
Recall	0.51	0.57
F1	0.58	0.64

Table 2: Scores obtained for basic and final models.

3.3 Conclusion

The NER model that I have proposed allows to obtain satisfactory classification results of the entity to one of four groups. It performs much better for the train data set, but it allows to obtain correct predictions for many cases of the test set, in particular for brands, drugs and groups. An area that should be improved in my model is the classification of entities belonging to the drug_n group: chemical substances affecting living organisms, but not approved for human use with the medical purpose. The lack of a clear distinction between the words describing the entity belonging to the drug and drug_n clusters results in lower recall scores obtained in the testing.

4 Drug-Drug Interaction

The goal of this subtask is the extraction of drug-drug interactions from biomedical texts.(14) A drug-drug interaction occurs when one drug influences the level or activity of another drug. The re-

sults should not consider the existence of the interaction between each drug-drug pair, but also their classification to one of the four types:

- advice: there is described the recommendation or advice regarding the concomitant use of two drugs
- effect: there is described the effect of drug-drug interaction: a pharmacological effect, a clinical finding, signs or symptoms, an unspecific modification of the effect or action of one of the drugs, an increased of the toxicity or a protective effect or therapeutic failure.
- mechanism: it can be pharmacodynamic (the effects of one drug are changed by the presence of another drug at its site of action) or pharmacokinetic (the process by which drugs are absorbed, distributed, metabolised and excreted are affected)
- int: it occurs when sentence simply states that an interaction occurs and does not provide any information about the interaction.

4.1 System description

my system, the two-stage SVM, is based on the architecture - with some modifications - from the work of Rastegar-Mojarad, Boyce and Prasad (9). Before classification I conduct the pre-processing phase and, afterwards, the extraction of two types of features: per sentence and per drug pair. Then I used two-stage SVM classifier to conduct the binary classification of drug pairs and then multi-class classification of interacting pairs into one of the four interacting types. The all stages of the study, as well as a description of the models, measures and results, are described in more detail in the following sections of the report.

4.1.1 Data preprocessing

As in the previous case, I decided to combine DrugBank and MedLine data sets and divide the training set in the ratio 80:20 for the purpose of adding the model validation step. Before the classification, I decided to submit all sentences in the three received data sets to the following preprocessing:

- all letters were changed to lower case
- all stop words and punctuation were removed
- Part-of-Speech tags were obtained with the Stanford NLP tool (12)
- words were lemmatized with the use of WordNet Lemmatizer
- words were stemmed with the Porter Stemmer. (8)

4.1.2 Feature extraction

Following the authors of original system: 'Since each sentence can have more than two drug mentions, I generated an instance of the sentence for each drug pair'. (9) I extracted the features both per sentence and per instance (each drug-pair). Features per sentence are sentence-level features that have the same values across all instances of a sentence. They include:

- Words: a numeric feature which counts how many words in the particular sentence appears in the whole sentence more than once (considering stemmed and lemmatized words)
- Word bigrams: a numeric feature which counts how many word bigrams in the particular sentence appears in the whole corpus more than once (I resigned from including this feature in the final model because of possible collinearity with the previous one)
- Number of words: total number of words in the sentence
- Number of drug mentions: total number of drug mentions in the sentence.

Features per drug-pair may have different values across the different drug-pair instances. For each sentence, I distinguished all pairs of two main drugs. These features are as follows:

- Number of words: number of words between two main drugs, before the first main drug and after the second main drug
- Number of drugs: number of drugs between two main drugs, before the first main drug and after the second main drug
- Number of POSes: number of POSes (nouns, verbs, adverbs and adjectives) between two main drugs, before first main drug and after the second main drug.

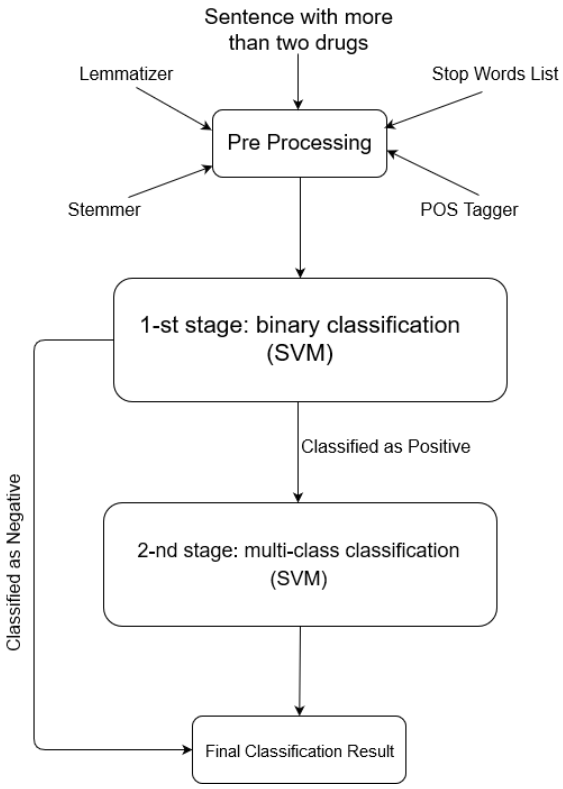

Figure 2: Architecture of proposed model

4.1.3 Entity classification process

I used SVM classifier to conduct the first stage: binary classification of drug pairs to to distinguish pairs interacting and non-interacting. In the next step, I made a multi-class classification of interacting pairs from the first stage into one of the four interacting types. Model approach is presented in the Figure 2. In order to evaluate the model performance, I used three metrics: precision, recall and F1 described in the correspondent section above.

Support Vector Machines are a set of supervised learning methods used for classification, regression and outliers detection. The support-vector network combines three ideas: the solution technique from optimal hyperplanes (that allows for an expansion of the solution vector on support vectors), the idea of convolution of the dot-product (that extends the solution surfaces from linear to non-linear), and the notion of soft margins (to allow for errors on the training set) (2). Following skicit-learn documentation (7) I can mention the following advantages of SVM which makes it proper to my case: can be applied both to binary and multi-class classification, effective in high dimensional spaces, uses a subset of training points in the decision function (called support vectors), so it is also memory efficient, versatile: different Kernel functions can be specified for the decision function. However, I should be aware that SVMs do not directly provide probability estimates - these are calculated using an expensive five-fold cross-validation.

SVMs' implementation in Python scikit-learn library specifies the following parameters:

- kernel - kernel function can be any of the following: linear, polynomial, RBF (Radial Basis Function), sigmoid, custom function defined separately in Python or using the precomputed Gram matrix. Training the model with RBF, two parameters: gamma and C should be considered.
- C - common for all SVM kernels. It trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly.
- gamma - defines how much influence a single training example has. The larger gamma is, the closer other examples must be to be affected.
- class-weights - in problems where it is desired to give more importance to certain classes or certain individual samples, weights can be set according to the will of the user.

Proper choice of C and gamma is critical to the SVMs performance.

4.2 Results

I trained 16 models, using different set of parameters. When choosing the best model, I used the Macro Average F1 criterion, taking into consideration also F1 scores for both stages. The parameters of the best iteration are as follows: C: 1.3, kernel: RBF, gamma: 0.02, weights balanced. I have trained the model on the entire initial train data set, and then I conducted the test using the test data set. I obtained the final results: Precision: 0.2022, Recall: 0.2062, F1: 0.2042. The full set of obtained scores is presented in the table 3.

The model performance on the first stage is medium. For test data set I get F1 equal to 0.51 and its score is better than for train score, and also recall for both data sets is around 0.60. However,

Metric	Train score	Test score
1st stage		
Precision	0.35	0.43
Recall	0.62	0.61
F1	0.45	0.51
2nd stage		
Precision	0.15	0.15
Recall	0.27	0.21
F1	0.19	0.18
Type mechanism		
Precision	0.23	0.23
Recall	0.19	0.14
F1	0.21	0.17
Type effect		
Precision	0.35	0.28
Recall	0.29	0.23
F1	0.32	0.25
Type advice		
Precision	0.19	0.28
Recall	0.34	0.33
F1	0.24	0.30
Type int		
Precision	0.01	0.02
Recall	0.20	0.14
F1	0.01	0.03
Macro-average		
Precision	0.20	0.20
Recall	0.25	0.21
F1	0.22	0.20

Table 3: Scores obtained for final model tested on different data sets.

the relatively low precision (0.35 for train and 0.43 for test set) means that model classifies many instances as FP (false positive). I consider it as a cause of low scores obtained in the second stage. In the multi-classification stage I take into consideration these instances, which in the first stage are classified as interacting. my approach assumes then, that in the second stage both TP and FP instances are classified into one of four groups of interaction. The final score takes into consideration the whole data set, including FN instances from first stage, which should be classified at the second stage, but because of false classification of non-interacting there are not classified to any group. In conclusion, the low score of the final classification is the result of cumulative classification errors from both stages. However, it should be noted that for both data sets the model receives similar (or even better, e.g. for first stage, types advice and int) results, which means that it has good predictive capabilities and I avoided overfitting.

4.3 Conclusion

In conclusion, the final score of the model is lower than I expected. It could be explained by cumulative classification errors from both stages: incorrectly classified at first stage instances repeat their error at the second stage and during final evaluation.

5 Summary and future work

To summarize, I proposed two models - the first classifying instances for the recognition of drugs, the second checking the existence of interactions between pairs of drugs and determining their type. The model in NER task allows us to obtain satisfactory classification results, but it is worth to notice that it performs significantly better for train data set. It could be caused by the choice of excessively specific set of features, which are relevant to the particular data set, not the whole issue. It would be reasonable to check a larger number of models using the test dataset or to apply cross-validation. In the modeling process, you can also include additional steps, such as changing model parameters or changing the model's form. Another area for improvements is the classification of entities belonging to the drug_n group. The distinction between the entities belonging to the drug and drug_n clusters is not clear, what affects the weaker classification score for these groups.
The model in DDI task obtains satisfactionary results for the first stage (binary) classification, but significantly lower results for the second stage. It is caused by duplicating the incorrect classification during calculation of final score. FP and FN obtained at the first stage are also incorrectly classified at the second stage. The way to avoid this problem could be post-processing, which involves correcting the results of both classification by hand-crafted rules. In addition, an interesting idea may be the use of weighted-SVM model after investigating the data in terms of unbalanced frequencies among the groups. I can also consider application of one-stage SVM aiming to classify all instances to one of five classes: the previous four and none interaction. This solution is likely to improve the performance by avoiding the cumulative errors. It is also worth to consider ex-

tending the set of binary variables regarding the occurrence of individual words in the investigated sentence.

6 Reference

Source code is available in the link bellow: https://drive.google.com/drive/folders/1RmxCu7ul9i0HL8PTut9e6f5OVq-8WMyM?usp=sharing

References

[1] Abney S., Schapire R. E., Singer Y. *Boosting applied to tagging and PP attachment*, Proc. EMNLPVLC. New Brunswick, New Jersey: Association for Computational Linguistics, 1999.

[2] Cortes C., Vapnik V., Singer Y. *Support-Vector Networks*, Machine Learning, 20, Kluwer Academic Publishers, Boston, 1995, pp. 273-297.

[3] Daelemans W., van den Bosch A., *Memory-Based Learning*, in: Clark A., Fox Ch., Lappin S., *The Handbook of Computational Linguistics and Natural Language Processing*, Blackwell Publishing, Chichester, 2010, part II, chapter 6.

[4] Freund Y., Schapire R. E. *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences, 55, 119139.(1997).

[5] Grego T., Pinto F., Couto F.M., *LASIGE: using Conditional Random Fields and ChEBI ontology*, University of Lisbon, Portugal, 2013.

[6] Lafferty J., McCallum A., Pereira F.C.N., *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*, Proceedings of the 18th International Conference on Machine Learning 2001, pages 282-289.

[7] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, Volume 12, 2011, pages 2825-2830.

[8] Porter M. F., *An algorithm for suffix stripping*, Program, 14(3), 1980, pages 130-137.

[9] Rastegar-Mojarad M., Boyce R.D., Prasad R., *"UWM-TRIADS: Classifying Drug-Drug Interactions with Two-Stage SVM and Post-Processing"*, Second Joint Conference on Lexical and Computational Semantics, Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, June 14-15, 2013, pages 667674.

[10] Schmid H., *Decision Trees, in:* Clark A., Fox Ch., Lappin S., *The Handbook of Computational Linguistics and Natural Language Processing*, Blackwell Publishing, Chichester, 2010, part II, chapter 7.

[11] Segura-Bedmar I., Martinez P., Herrero-Zazo M., *SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013)*, Second Joint Conference on Lexical and Computational Semantics (*SEM), Vol. 2, pages 341350, Atlanta, Georgia, June 14-15, 2013.

[12] Toutanova K., Klein D., Manning C., Singer Y., *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*, In Proceedings of HLT-NAACL, 2003, pages 173-180.

[13] *Evaluation of the SemEval-2013 Task 9.1: Recognition and Classification of pharmacological substances*.

[14] *Task 9.2: Extraction of drug-drug interactions*.