# UWM-TRIADS: Classifying Drug-Drug Interactions with Two-Stage SVM and Post-Processing

**Majid Rastegar-Mojarad**
University of Wisconsin-Milwaukee
Milwaukee, WI, USA
Rastega3@uwm.edu

**Richard D. Boyce**
University of Pittsburgh
Pittsburgh, PA, USA
rdb20@pitt.edu

**Rashmi Prasad**
University of Wisconsin-Milwaukee
Milwaukee, WI, USA
prasadr@uwm.edu

## Abstract

We describe our system for the DDIExtraction-2013 shared task of classifying Drug-Drug interactions (DDIs) given labeled drug mentions. The challenge called for a five-way classification of all drug pairs in each sentence: a drug pair is either non-interacting, or interacting as one of four types. Our approach begins with the use of a two-stage weighted SVM classifier to handle the highly unbalanced class distribution: the first stage for a binary classification of drug pairs as interacting or non-interacting, and the second stage for further classification of interacting pairs from the first stage into one of the four interacting types. Our SVM features exploit stemmed words, lemmas, bigrams, part of speech tags, verb lists, and similarity measures, among others. For each stage, we also developed a set of post-processing rules based on observations in the training data. Our best system achieved 0.472 F-measure.

## 1 Introduction

Potential drug-drug interactions (DDIs), defined as the co-prescription of two drugs that are known to interact, are a significant source of preventable drug-related harm (i.e., adverse drug events, or ADEs) (Nebeker et al., 2004). Gurwitz et al, in their cohort study of ADEs among older Americans receiving ambulatory care, found that 13.3% of preventable errors leading to an ADE involved the co-prescription of drugs for which a "...well established, clinically important interaction" was known (Gurwitz et al., 2003). Nearly 7% (23/338) of the ADEs experienced by residents of two academic nursing homes over a nine-month period were attributable to DDIs (Gurwitz et al., 2005). Sixteen cohort and case-control studies reported an elevated risk of hospitalization in patients who were exposed to DDIs (Hines et al., 2011).

Failure to properly manage a DDI is a medical error, and the Institute of Medicine has noted that a lack of drug knowledge is one of the most frequent proximal causes of such errors (Committee on Identifying and Preventing Medication Errors, 2007). Indeed, health care providers often have inadequate knowledge of what drug interactions can occur, of patient specific factors that can increase the risk of harm from an interaction, and how to properly manage an interaction when patient exposure cannot be avoided (Chen et al., 2005; Hines et al., 2012).

Unfortunately, there is no single complete and authoritative source of DDI knowledge (Hines et al., 2012). Rather, there are multiple sources, each tasked with extracting, evaluating, and staying up-to-date with pertinent DDIs reported in the literature, and drug product labeling (Boyce et al., 2012). The dynamic nature of drug knowledge, combined with the enormity of the biomedical literature, makes this task extremely challenging. Hence, natural language processing methods for identifying and extracting DDIs are receiving increased attention.

In 2011, the first shared task challenge for DDI extraction, DDIExtraction-2011 (Segura-Bedmar et al., 2011), invited participants to develop automatic methods to extract DDIs. The task focused on the identification of all possible pairs of interacting drugs, without specifying anything further about the interactions. By contrast, the DDIExtraction-2013 (Segura-Bedmar et al., 2013) shared task emphasized the importance of recognizing *what is being asserted about the interaction*. Accordingly, the challenge called for a



*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 667–674, Atlanta, Georgia, June 14-15, 2013. 

five-way classification of sentences for each drug-pair:

- *Advice*: the sentence notes a recommendation or advice related to the concomitant use of the two drugs (e.g., "… **UROXATRAL** should NOT be used in combination with other **alpha-blockers**.");
- *Effect*: the sentence states the effect of the drug interaction, including pharmacodynamic effect or mechanism of interaction (e.g., "**Quinolones** may enhance the effects of the oral anticoagulant, **warfarin**, …");
- *Mechanism*: the sentence describes a pharmacokinetic mechanism (e.g., "**Grepafloxacin** is a competitive inhibitor of the metabolism of **theophylline**.").
- *Int:* the sentence mentions a drug interaction but doesn't provide any additional information (e.g., "The interaction of **omeprazole** and **ketoconazole** has been established.").
- *None:* the sentence does not show an interaction between the two drugs;

To focus on, and separately evaluate, different aspects of the problem, the 2013 shared task was divided into two subtasks. One task focused on the recognition and classification of drug names, while the other focused on the identification and classification of DDIs, with the drug names provided from the gold standard. In this paper, we describe our approach for handling the second task, namely, DDI identification and classification of all possible pairs of drugs in the provided corpus. Our approach combined machine-learning methods with the use of rules for post-processing. A key feature of our machine-learning approach is that it is specifically designed to handle the highly unbalanced class distribution via the use of a two-stage weighted SVM classifier. In addition to a variety of features exploited for the classifier, we also developed a set of post-processing rules, with a different set of rules applied after each stage of SVM classification. Finally, our approach is also aimed towards exploring the efficacy of methods that do not need to rely on syntactic-parse based features.

The paper is organized as follows. In the next section, we describe the training and test data set used in the challenge. In section 3, we describe our method, the classifiers used at each stage, their features, and post processing. In section 4, we present the evaluation and results. We conclude in Section 5 with discussion and future work.

## 2 Data

The DDIExtraction-2013 challenge provided a DDI corpus for development, containing 142 Medline abstracts on the subject of drug-drug interactions, and 572 documents describing drug-drug interactions from the DrugBank database. The corpus includes 6976 sentences that were annotated with four types of pharmacological entities and four types of DDIs. The DDIs types are: *advice, effect, mechanism, and int.*[1] Table 1 shows the number of instances for each type. Examples can be seen in Section 1. The test set includes 33 Medline abstracts and 158 DrugBank documents containing 1299 sentences and 5519 drug pairs.

| **Type** | | **Number** |
|---|---|---|
| Positive | Advice | 827 |
| | Effect | 1700 |
| | Mechanism | 1322 |
| | Int | 188 |
| Negative | None (non-interacting drugs) | 23772 |
| **Total** | | **27809** |

Table 1: Number of instances in each class

## 3 Methods

Classification of each drug pair in a sentence involved distinguishing between 5 classes, *advice, effect, mechanism, int and none*. As described in Section 2 (see Table 1), a major challenge in this task is posed by the unbalanced distribution of the classes. First, considering just the positive vs. negative classes, just 16.9% (4037/23772) of drug pairs are in the positive class, which include interacting drug pairs (labeled as *advice*, *effect*, *mechanism* and *int*). Furthermore, the four types

[1] http://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/task-9.2-ddi-extraction.pdf

within the positive class are also unbalanced, with the *int* type constituting only 4.6% (188/4037) of the instances. A classifier trained on this data will, therefore, be biased towards the majority class(es). We employed a two-stage classification approach to cope with this problem, as described below.

### 3.1 Two-stage classification

Figure 1 shows the architecture of the system. In the first stage, we trained a binary classifier to classify drug pairs into *positive* and *negative* classes. Then, in the second stage, we considered only instances that were classified as positive by the first classifier, and classified them into *advice, effect, mechanism, and int* classes, using a multi-class classifier. A two-stage classifier offers a distinct advantage over a one-stage classifier for the DDI data set, which is highly skewed towards one class, but particularly because this majority class is also clearly semantically distinct from the other positive classes (see Table 1). By reframing part of this problem as a binary classification task, we can exploit binary classification techniques and allow the classifier to be particularly attentive to features distinguishing positive and negative drug pairs, while at the same time avoiding the bias against each of the non-majority classes. Our experiments with the training set confirm this idea.

Despite the above advantage of a two-stage SVM, however, the unbalanced class problem still remains, especially for training at the first stage, where we have 20854 negative instances and 4026 positives instances. In the second stage, the data is somewhat unbalanced as well, with 20.5% as *advice*, 42.2% as *effect*, 32.6% as *mechanism*, and only 4.7% as *int*. To handle this problem further, we explored different approaches and algorithms, including SMOTE (Chawla et al. 2002) and other resampling algorithms. Our best results over the training data were obtained with Support Vector Machine (SVM) with different class weights. We used LibSVM (Chang and Lin, 2011) and set class weights for each stage using results of cross-validation over the training data (see Table 3 for class weights).

As we wanted to pass the positively classified instances from the first stage to the second stage classifier, we favored the *positive* class in the first stage. This resulted in a relatively high number of false positives for the positive instances, which we attempted to reduce with a set of post-processing rules before sending them to the second stage classifier. A different set of post-processing rules were also developed to apply on the output of the second stage classifier.

### 3.2 Pre-processing

Before classification, all sentence instances in the training and test set were pre-processed for the following:

- All letters were changed to lower case.
- All drug names were normalized by replacing them with one of two strings; one used for drug mentions that were candidates for clas-

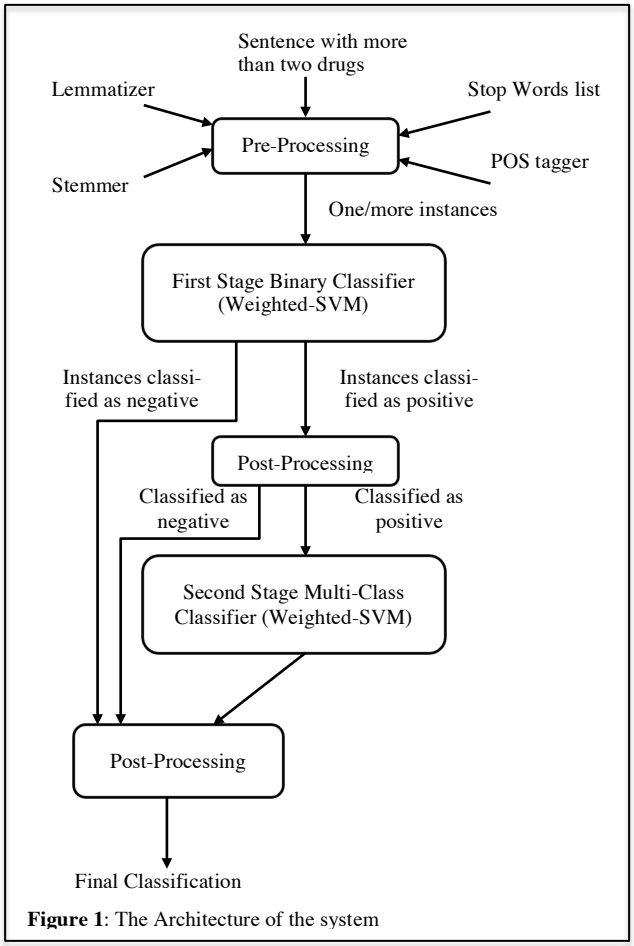


**Figure 1**: The Architecture of the system

sification in the instance, and the other used for all other drug mentions.
- All numbers were normalized by replacing them with the same string.
- Stop words and punctuation were removed. We used different stop word lists for different systems that were submitted to the challenge.
- Part of speech (POS) tags were obtained with the Stanford NLP tool (Toutanova et al, 2003).
- Words were stemmed with the Porter Stemmer (Porter, 1980).
- Words were lemmatized with the dragon tool (Zhou et al, 2007).
- Synsets for words were obtained using WordNet (Fellbaum, 1998).

### 3.3 Features

Since each sentence can have more than two drug mentions, we generated an instance of the sentence for each drug pair. We used different combinations of various features for the three different systems submitted to the challenge (Section 3.4.3). The following describes all the features separated into two categories: features per sentence and features per drug-pair instances.

**Features per sentence:** These are sentence-level features that have the same values across all instances of a sentence.

1- *Words*: This is a binary feature for all words that appeared more than once in the corpus, indicating the presence or absence of each such word in the sentence. We considered stemmed words as well as lemmatized words.
2- *Word bigrams*: This is a binary feature for all word bigrams that appeared more than once in the corpus, indicating the presence or absence of each such bigram in the sentence
3- *Number of words*: This feature represents the total number of words in the sentence
4- *Number of drug mentions*: This feature represents the total number of drug mentions in the sentence.
5- *Cosine similarity between centroid vector of each class and the instance*: Inspired by the vector space Information Retrieval approach, we added new features to represent the cosine similarity between a sentence and the centroid of normalized vectors for sentences assigned the class *X*. Cosine similarity is calculated based on modified tf*idf. We computed modified tf*idf for a word w in class *C*, based on the following formula:

$$(TF * IDF)_{w,C} = \log(count(w,C)+1) * \log(total\#Inst / (\#inst\_contains\_w+1))$$

TF is the logarithm of the number of times the word occurs in all sentences assigned to the class. IDF is 1.0 divided by the logarithm of number of instances in the class divided by the number of times the word occurs across all classes. To calculate the centroid vector for class C, a vector is created for each sentence in class C by giving each word in the sentence a modified TF*IDF weight. The centroid vector for class C is the mean of all vectors of sentences in class C. The Cosine similarity between a given instance and the centroid vector of each class is then used a feature.

**Features per instance (for each drug-pair):** In contrast to sentence-level features, these features may have different values across the different drug-pair instances. In each instance, we distinguished the two *main* drugs of interest for the instance from all other *additional* drugs mentioned in the instance.

1- *Number of words between two main drugs*: This represents the total number of words between the two main drugs.
2- *Number of drugs between two main drugs*: This represents the total number of additional drugs appearing between the two main drugs.
3- *Number of verbs*: We used the number of verbs in the instance as a feature, but relative to their sentential position. In particular, we split each instance into three sections: (i) before the first main drug, (ii) between the two main drugs, and (iii) after the second main drug. Then, we counted the number of verbs in each section, and used them as three different features.
4- *Number of verbs using class-specific verb lists*: For each class, we extracted two lists of verbs. The first list contains verbs that ap-

peared in just that class but not in the others. Thus, the set of verbs extracted for each class are unique and different from the verbs associated with other classes. The second list includes all verbs that appeared in that class and their synonyms, extracted from WordNet. Then, for each of the three sentence sections, as described above, we created two features to represent the number of verbs from each of these lists that appeared in the section. (An alternative way to represent this feature would be to weight the verbs according to their relative frequencies in the different classes.)

5- *POS of words between two main drugs*: This is a binary feature for word POS tags obtained from POS tagging, and indicates the presence or absence of each POS between the two main drugs.

### 3.4 Post processing

As described in Section 3.1, we developed a set of post-processing rules for each stage of the classifier. Here, we describe these rules, developed on the basis of observations in the training data.

#### 3.4.1 Post processing after the first stage

Post-processing rules for the first stage were designed to reduce the number of false positives for the positive class, since the weight assignment in this stage favors this class. We provide examples for each rule:

- The instance is classified as negative if both drug mentions have the same name, since a drug cannot interact with itself.

  "*In controlled clinical trials of AUGMENTIN XR, 22 patients received concomitant allopurinol and AUGMENTIN XR."*

- The instance is classified as negative if one of the drugs is a plural form of the other one, since, as above, they refer to the same drug.

  *"Oral Anticoagulants: Interaction studies with warfarin failed to identify any clinically important effect on the serum concentrations of the anticoagulant or on its anticoagulant effect."*

- The instance is classified as negative if one of the drug mentions refers to a drug class name of the other, since we don't expect a drug to interact with its class. Drug class names were obtained from a classification provided by the FDA.[2] In the example below, "MAOI" is the drug class name for "isocarboxazid".

  *"You cannot take mazindol if you have taken a monoamine oxidase inhibitor (MAOI) such as isocarboxazid (Marplan), tranylcypromine (Parnate), or phenelzine (Nardil) in the last 14 days."*

- The instance is classified as negative if "," or ", and" appears between the two main drug mentions, and is accompanied by an additional drug mention. This rules identifies contexts where drugs are mentioned as a set, in interaction with a different drug. The following sentences show "glyburide", "tolbutamide" and "glipzide" as part of a set of drugs in interaction with the additional drug "DIFLUCAN".

  *"DIFLUCAN reduces the metabolism of tolbutamide, glyburide, and glipizide and increases the plasma concentration of these agents."*

  "*DIFLUCAN reduces the metabolism of tolbutamide, glyburide**, and** glipizide and increases the plasma concentration of these agents.*"

- The sentence is classified as negative if "," and additional drugs appear between the main drug mentions. Like the previous rule, this again recognizes drugs mentioned as a set but identifies non-adjacent mentions. For example, the following sentence doesn't express any interaction between "tolbutamide" and "glipizide", and the rule recognizes them as part of a set mention even though they are non-adjacent.

  *"DIFLUCAN reduces the metabolism of tolbutamide, glyburide, and glipizide and increases the plasma concentration of these agents."*

[2] http://www.fda.gov/ForIndustry/DataStandards/StructuredProductLabeling/ucm162549.htm

- The instance is classified as negative if "or" appears between the two main drug mentions and the sentence contains additional drug mentions. The presence of additional drug mentions in the sentence is required here since such conjoined pairs can interact with each other when they occur alone.

  *"Concurrent ingestion of antacid (20 mL of antacid containing aluminum hydroxide, magnesium hydroxide, and simethicone) did not significantly affect the exposure of oxybutynin **or** desethyloxybutynin."*

### 3.4.2 Post processing after the second stage

Post-processing after the second classifier identifies sentences like the following:

*"Coadministration of **alosetron** and strong CYP3A4 inhibitors, such as **clarithromycin**, **teli thromycin**, **protease inhibitors**, **voriconazole**, and **itraconazole** has not been evaluated but should be undertaken with caution because of similar potential drug interactions."*

Examples like these illustrate that if drugs are mentioned as a set, then all drugs in the set must have the same interaction type with a drug mentioned outside the set. Thus, in the example, the interaction of each of "clarithromycin", "telithromycin", "protease inhibitors", "voriconazole", and "itraconazole" with "alosetron" should be classified in the same way. We used several syntactic and lexical cues to identify set mentions of drugs. Then, since the SVM classifier can make different decisions for each such pair (e.g., it may assign one label to the interaction of "clarithromycin" with "alosetron" and another label to the interaction of "telithromycin" with "alosetron"), we applied uniform labeling for the interaction of all such pairs. The majority label was used as the common label. Ties were not encountered in this data, although a solution would have to be devised otherwise.

An important consideration for this rule is that it uses both positively and negatively labeled instances. The former are taken from the result of the second stage classifier, and the latter from the negative instances of the first stage classifier and the negative instances of the first post-processor. These varied inputs to the rule are illustrated by the three ingoing arrows into the second post-processor in Figure 1.

### 3.4.3 Submitted Systems

We used the Weka (Hall et al. 2009) tool for all experiments and submitted three systems (System1, System 2, and System 3 in Table 2) to the challenge. All systems used the same two-stage approach and SVM classification (LibSVM), but differed in the use of some of the features (Section 3.3) and in the weights assignment (Table 3). We used linear kernel and the cost (C) was 1.2 and gamma was 0.5. In System 1, we used stemmed words (instead of lemmatized words) and a stop word list of 165 words. In System 2, we used stemmed words again, but a different

| System | Metric | Drug-Bank | Medline | All |
|---|---|---|---|---|
| System 1 | Prec | 0.44 | 0.21 | 0.43 |
| | Rec | 0.49 | 0.23 | 0.47 |
| | F | 0.46 | 0.22 | 0.45 |
| System 2 | Prec | 0.49 | 0.30 | 0.47 |
| | Rec | 0.49 | 0.41 | 0.47 |
| | F | 0.49 | 0.35 | 0.47 |
| System 3 | Prec | 0.42 | 0.26 | 0.40 |
| | Rec | 0.51 | 0.47 | 0.50 |
| | F | 0.46 | 0.33 | 0.44 |

Table 2. Results of each system. The three systems are described in Section 3.4.3.

| System | Stage | Class | Weight |
|---|---|---|---|
| System 1 | First Stage | Positive | 6.5 |
| | | Negative | 1.0 |
| | Second Stage | Advice | 800.0 |
| | | Effect | 600.0 |
| | | Int | 3200.0 |
| | | Mechanism | 500.0 |
| System 2 | First Stage | Positive | 2.5 |
| | | Negative | 1.0 |
| | Second Stage | Advice | 800.0 |
| | | Effect | 600.0 |
| | | Int | 3200.0 |
| | | Mechanism | 500.0 |
| System 3 | First Stage | Positive | 6.5 |
| | | Negative | 1.0 |
| | Second Stage | Advice | 80.0 |
| | | Effect | 60.0 |
| | | Int | 320.0 |
| | | Mechanism | 50.0 |

Table 3: Class weight assignments in different systems

stop word list of 263 words. Finally, in System 3, we used lemmatized words and the same stop word list of 263 words as in System 2. Weights assignment was different across all systems, as shown in Table 3.

## 4 Results

Table 2 shows the evaluation results of our system over the test set. Our best results are achieved with System 2, in which we used stemmed words and our 263 stop word list, in addition to the other features described in Section 3.3. Both the stop word list and the use of stemmed vs. lemmatized words can be seen to affect the performance. Clearly, a larger stop word list is more useful, since both System 2 and System 3 show an improvement over System 1. On the other hand, the use of lemmas (used in System 3) seems to be detrimental, compared with stemmed words.

## 5 Conclusion and future work

To the best of our knowledge, this is the first study to explore the value of a two-stage SVM classification process for performing the complex task of identifying sentences describing DDIs, and making the important distinction between statements providing advice, mechanism and effect, or declaring a pharmacokinetic and pharmacodynamic DDI: critical distinctions in the fields of pharmacology and pharmacy. We find that the use of a two-stage classifier to handle the problem of an unbalanced class distribution for the task of identifying and classifying DDIs is feasible but requires further development.

It's valuable to consider these results within the context of previous efforts for extracting DDIs. Ten research papers were presented at the 2011 SemEval Conference (Segura-Bedmar et al, 2011) which used a smaller DDI corpus (Medline abstracts were not included) and a simpler classification task (Segura-Bedmar et al, 2010). The best performing system in this challenge utilized an ensemble learning approach (Thomas et al, 2011) and produced an F-measure of 0.657. The second best performing method utilized composite kernels, a method that combines feature-based and kernel-based methods, and was found to perform with an F-measure of 0.64 (Chowdhury et al, 2011). Other NLP research has focused exclusively on extracting pharmacokinetic DDIs from either Medline (e.g., Airola et al, 2008) or drug product labeling (e.g., Boyce et al, 2012).

Due to time constraints, we couldn't test other classifiers such as Naïve Bayes, JRip and Randomforest in our approach. Future work will test if SVM is the best choice for the first stage binary classifier. It is possible that libShortText (Yu et al, 2013) works better than LibSVM because this task is for sentence classification. We also plan to explore if Naïve Bayes, JRip, or Randomforest could work better than SVM for the second stage multi-class classifier.

Since only three systems were permitted to the challenge, and since the labeled test data was not available until the time of writing, we did not have the opportunity to test the impact of all the features that we considered, or of the post-processing rules. This will be explored in future work.

We also plan to explore some variations to our approach. For example, we will try to incorporate some of the rules, especially those in the first post-processor, as features in our system. Finally, although we did utilize some semantic information from WordNet for this work, we would like to explore additional rich features, drawing on syntax, semantics and discourse.

## References


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

Boyce R. D., C. Collins, M. Clayton, J. Kloke and J. R. Horn. 2012. Inhibitory metabolic drug interactions with newer psychotropic drugs: inclusion in package inserts and influences of concurrence in drug interaction screening software. *The Annals of Pharmacotherapy* 46.10 (2012): 1287-1298

Boyce R. D., G. Gardner and H. Harkema. 2012. Using Natural Language Processing to Extract Drug-Drug Interaction Information from Package Inserts. *Proceedings of the 2012 Workshop on BioNLP*.

Chang C. and C. Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2(3): 27.

Chawla N. V., K. W. Boyer, L. O. Hall and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16: 321-357.

Chen Y. F., A. J. Avery, K. E. Neil, C. Johnson, M. E. Dewey and I. H. Stockley. 2005. Incidence and possible causes of prescribing potentially hazardous/contraindicated drug combinations in general practice. *Drug Safety*, 28(1): 67-80.

Chowdhury F. M., A. B. Abacha, A. Lavelli and P. Zweigenbaum. 2011. Two Different Machine Learning Techniques for Drug-Drug Interaction Extraction. *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction (DDIExtraction-2011)*, 19–26.

Committee on Identifying and Preventing Medication Errors, Aspden P, Wolcott J, Bootman JL, and Cronenwett LR. 2007. Preventing Medication Errors: Quality Chasm Series. Washington, D.C. *The National Academies Press*.

Fellbaum C. 1998. *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press.

Gurwitz J. H., T. S. Field, L. R. Harrold, J. Rothschild, K. Debellis, A. C. Seger, C. Cadoret, L. S. Fish, L. Garber, M. Kelleher and D. W. Bates. 2003. Incidence and preventability of adverse drug events among older persons in the ambulatory setting. *Journal of the American Medical Association*, 289(9): 1107-1116.

Gurwitz J. H., T. S. Field, J. Judge, P. Rochon, L. R. Harrold, C. Cadoret, M. Lee, K. White, J. LaPrino, J. Erramuspe-Mainard, M. DeFlorio, L. Gavendo, J. Auger and D. W. Bates. 2005. The incidence of adverse drug events in two large academic long-term care facilities. *The American Journal of Medicine*, 118(3): 251-258.

Hall M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, Volume 11, Issue 1.

Hines L. E., and J. E. Murphy. 2011. Potentially harmful drug-drug interactions in the elderly: a review. *The American Journal of Geriatric Pharmacotherapy*, 9(6): 364-377.

Hines L. E., D. C. Malone and J. E. Murphy. 2012. Recommendations for Generating, Evaluating, and Implementing Drug-Drug Interaction Evidence. *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy*, 32(4): 304-313.

Nebeker J. R., P. Barach and M. H. Samore. 2004. Clarifying Adverse Drug Events: A Clinician's Guide to Terminology, Documentation, and Reporting. *Annals of Internal Medicine*, 140(10): 795-801.

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

Segura-Bedmar I., P. Martinez and C. Pablo-Sanchez. 2010. Extracting drug-drug interactions from biomedical texts. *BMC Bioinformatics* 11, Suppl 5, P9.

Segura-Bedmar I., P. Martinez and D. Sánchez-Cisneros. 2011. The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction (DDIExtraction-2011)*.

Segura-Bedmar I., P. Martínez and M. Herrero-Zazo. 2013. SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts. *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.

Thomas P., M. Neves, I. Solt, D. Tikk and U. Leser. 2011. Relation Extraction for Drug-Drug Interactions using Ensemble Learning. *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction (DDIExtraction-2011)*.

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

Yu H., C. Ho, Y. Juan and C. Lin. 2013. *LibShortText: A Library for Short-text Classification and Analysis*. Technical Report. http://www.csie.ntu.edu.tw/~cjlin/ papers/libshorttext.pdf.

Zhou X., X. Zhang and X. Hu. 2007. Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI),* 197-201.