EXTRACTION AND CLASSIFICATION OF DRUG-DRUG INTERACTION FROM BIOMEDICAL TEXT USING A TWO-STAGE CLASSIFIER

by

Majid Rastegar-Mojarad

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Engineering

at

The University of Wisconsin-Milwaukee

December 2013

ABSTRACT
EXTRACTION AND CLASSIFICATION OF DRUG-DRUG INTERACTION FROM BIOMEDICAL TEXT USING TWO-STAGE CLASSIFIER

by

Majid Rastegar-Mojarad

The University of Wisconsin-Milwaukee, 2013
Under the Supervision of Professor Rashmi Prasad

One of the critical causes of medical errors is Drug-Drug interaction (DDI), which occurs when one drug increases or decreases the effect of another drug. We propose a machine learning system to extract and classify drug-drug interactions from the biomedical literature, using the annotated corpus from the DDIExtraction-2013 shared task challenge. Our approach applies a two-stage classifier to handle the highly unbalanced class distribution in the corpus. The first stage is designed for binary classification of drug pairs as interacting or non-interacting, and the second stage for further classification of interacting pairs into one of four interacting types: *advise, effect, mechanism,* and *int*. To find the set of best features for classification, we explored many features, including stemmed words, bigrams, part of speech tags, verb lists, parse tree information, mutual information, and similarity measures, among others. As the system faced two different classification tasks, binary and multi-class, we also explored various classifiers in each stage. Our results show that the best performing classifier in both stages was Support Vector Machines, and the best performing features were 1000 top informative words and part of speech tags between two main drugs. We obtained an F-Measure of 0.64, showing a 12% improvement over our submitted system to the DDIExtraction 2013 competition.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisor, Rashmi Prasad, who patiently guided me through the project. Without her supervision and constant help, this project would not have been possible. I could not have imagined having a better advisor and mentor for this study. My interest in information extraction related to drug-drug interactions developed while collaborating with Richard Boyce. I want to thank him for introducing me to this interesting topic and also for his support, guidance, and helpful suggestions. I would like to thank to my committee members, Susan McRoy and Rohit J. Kate. Also, I would like to thank Soheil Moosavinasab, who as a good friend was willing to help and give his best suggestions. It would have been a lonely lab without him.

I am dedicating this thesis to my parents who were always supporting me and encouraging me with their best wishes.

Chapter One:

Introduction

1- 1 Motivation

One of the important causes of medical errors is Drug-Drug Interactions (DDI). Some studies estimate that medical errors result in around 44,000 and 98,000 deaths per year in the United States[1], and that 7,000 of those deaths are due to medication-related errors [2]. Other reports have noted similar patterns around the globe [1], [3]–[5] so that this problem is not unique to the United Sates. The good news, however, is that most of these medication-related errors are preventable. A report in Australia [6] mentioned that 75% of hospital admissions related to medication errors are preventable.

Potential drug-drug interactions, defined as the co-prescription of two or more drugs that are known to interact, are a significant source of preventable drug-related harm (i.e., *adverse drug events* (*ADEs*)) [7]. A recent review estimates that clinically important events that are attributable to potential drug-drug interaction exposure occur in 5.3% - 14.3% of inpatients, and are responsible for 0.02% to 0.17% of the 129 million emergency department visits that occur each year [8]. Gurwitz et al, in a cohort study of adverse drug events, reported that 13.3% of preventable errors leading to an ADE involved the co-prescription of drugs for which interactions are well known [9]. Nearly 7% (23/338) of the ADEs experienced by residents of two academic nursing homes over a nine-month period were attributable to DDIs [10]. Sixteen cohort and case-control studies reported an elevated risk of hospitalization in patients exposed to DDIs [11].

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 [12]. 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 of how to properly manage an interaction when patient exposure cannot be avoided [11], [13].

Unfortunately, there is no single complete and authoritative source of DDI knowledge [14]. 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 [15]. The following is a list of the resources that currently provide this information:

1- **DrugBank** [15]: is a database containing bioinformatics and cheminformatics information. It combines detailed information about drugs and their targets (such as sequence, structure).

2- **DailyMed**[1]: Through the DailyMed website, the U.S. National Library of Medicine provides downloadable information about marketed drugs, including FDA labels (package inserts). Package inserts are authoritative resources which healthcare professionals can rely on for warnings and precautions about the potential dangers of prescribing a drug, and the potential interactions and side-effects that a drug may have.

3- **National Drug File - Reference Terminology (NDF-RT)** [16], [17]: is produced by the U.S. Department of Veterans Affairs, Veterans Health Administration (VHA). It provides information about drug characteristics, including ingredients, chemical structure, dose form, physiologic effect, mechanism of action, drug interaction, and related diseases.

[1] http://dailymed.nlm.nih.gov/dailymed/about.cfm

4- **Micromedex**[2] **(Drug-Reax)**: is a US drug compendium. It is commonly used by health professionals to obtain information on drug interactions. It further classifies the drug-drug interactions into different categories, such as major, moderate, and minor, based on their severity.

5- **Express Scripts DRUG DIGEST Check Interactions**[3]: is a database including more than 5,000 drugs and herbals and 11,500 potential interactions.

6- **Medscape for WebMD**[4]: provides information on interactions resulting from combination of drugs, herbals and supplements.

Despite the availability of several such resources providing information on drug interactions, there are bottlenecks in fully and effectively utilizing them. Some such as DrugBank, are simply not practical or useful for prescribers to review at the point–of-care, while others, such as product labels, are often complete and updated, even though they are often reviewed at the point of care.

The bigger issue, however, is the inconsistency of information found in these resources, and sometimes, between them. Our own study [18] demonstrates this for package inserts. Manual review by a clinical pharmacologist of 100 randomly selected potential interactions out of more than 300,000 automatically extracted interactions revealed that 40% were genuinely inconsistent. Vitry [19] studied the consistency and grading of major drug interactions for 50 drugs in four leading international drug interaction compendia: the drug interactions appendix of the British National Formulary, the interaction

[2] http://www.micromedex.com/

[3] http://www.drugdigest.org/

[4] http://reference.medscape.com/drug-interactionchecker

supplement in the French Drug Compendium Vidal, and two US drug interaction compendia, Drug Interaction Facts and Micromedex. They analyzed major interactions for a list of 50 drugs in all four compendia and found that 14-44% of interactions are not mentioned in at least one resource. The authors also noted inconsistencies in the severity grading system between the compendia. Shah et al. [20] found that the drug-drug interaction knowledge databases used for clinical decision support systems are not consistent with official package labels, causing spurious warnings and inaccurate information. They highlighted the importance of the accuracy and completeness of drug databases. Li et al. [21] showed that the top commercial drug interaction databases take approximately one month to update with new information.

The dynamic nature of drug knowledge, combined with the enormity of the biomedical literature, makes the task of collecting and maintaining up-to-date information on drug interactions extremely challenging and time-consuming. Therefore, there is a strong need to approach this task with automated methods, supplemented with human effort. Natural language processing (NLP) and information extraction methods for identifying and extracting DDIs have been receiving increased attention in the last few years, and several attempts have already been made to develop methods for this task, showing good potential for success, although there remains much room for improvement.

The goal of this thesis is to design, implement, and evaluate a method for extracting and classifying drug-drug interactions from biomedical text using machine learning algorithms and NLP methods. For this purpose, we participated in a shared task challenge, the DDIExtraction 2013 Shared Task Challenge, that provided a benchmark dataset for drugs and drug-drug interactions. Participating in the shared task allowed us to

directly compare our system with those of several others participating in the challenge, and also, since it took place in 2013, assured us that this task poses an ongoing challenge to the current research community. The DDIExtraction 2013 challenge was a follow up to the first challenge organized for this shared task in 2011. In what follows, we describe the design of both of these challenges. Methods developed by the participants of the challenges, along with their results, are discussed in Chapter 2.

1-2 Benchmark datasets for Information Extraction of Drug-Drug Interactions

Two shared tasks, DDIExtraction-2011 and DDIExtraction-2013, provided two benchmark datasets for drug-drug interactions.

1-2-1 The 1st Drug-Drug Interaction Extraction 2011 Shared Task Challenge (DDIExtraction-2011)

In 2011, the first shared task challenge for DDI extraction, DDIExtraction-2011 [22], invited participants to develop automatic methods to extract DDIs. Ten teams participated in the challenge. The task focused on the identification of all possible pairs of interacting drugs in sentences, without specifying anything further about the interactions. The organizers provided an annotated corpus for training and testing and evaluated the participants' systems on the test portion of the corpus. The corpus was drawn from 579 documents about DDIs from DrugBank. After choosing the documents, the UMLS MetaMap Transfer [23] was applied to automatically identify drug names in the sentences, and sentences containing two or more drugs were then manually annotated by a researcher with a pharmaceutical background. Of the total of 5,806 sentences across the 579 documents, 2,044 sentences were found to contain at least one DDI, and there were a

total of 3,160 DDIs in the corpus. Figure 1-1 illustrates one sentence of this corpus annotated with DDIs, presented in the unified XML format [24]: <sentence>, <entity>, and <pair> element tags mark up the raw sentence text, drug entities found in the sentence, and DDIs found in the sentence, respectively. Word-based offsets are used to link drug entities to the raw text. The "interaction" attribute in the <pair> element encodes the classification of a drug pair as "true" (i.e., interacting) or "false" (i.e., non-interacting).

```
−<sentence id="DrugDDI.d346.s0" origId="s0" text="Uricosuric Agents: Aspirin may decrease the effects of probenecid,
  sulfinpyrazone, and phenylbutazone.">
    <entity id="DrugDDI.d346.s0.e0" origId="s0.p0" charOffset="0-17" type="drug" text="Uricosuric Agents"/>
    <entity id="DrugDDI.d346.s0.e1" origId="s0.p2" charOffset="19-26" type="drug" text="Aspirin"/>
    <entity id="DrugDDI.d346.s0.e2" origId="s0.p6" charOffset="55-65" type="drug" text="probenecid"/>
    <entity id="DrugDDI.d346.s0.e3" origId="s0.p7" charOffset="67-81" type="drug" text="sulfinpyrazone"/>
    <entity id="DrugDDI.d346.s0.e4" origId="s0.p9" charOffset="87-101" type="drug" text="phenylbutazone"/>
    <pair id="DrugDDI.d346.s0.p0" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e1" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p1" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e2" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p2" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e3" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p3" e1="DrugDDI.d346.s0.e0" e2="DrugDDI.d346.s0.e4" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p4" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e2" interaction="true"/>
    <pair id="DrugDDI.d346.s0.p5" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e3" interaction="true"/>
    <pair id="DrugDDI.d346.s0.p6" e1="DrugDDI.d346.s0.e1" e2="DrugDDI.d346.s0.e4" interaction="true"/>
    <pair id="DrugDDI.d346.s0.p7" e1="DrugDDI.d346.s0.e2" e2="DrugDDI.d346.s0.e3" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p8" e1="DrugDDI.d346.s0.e2" e2="DrugDDI.d346.s0.e4" interaction="false"/>
    <pair id="DrugDDI.d346.s0.p9" e1="DrugDDI.d346.s0.e3" e2="DrugDDI.d346.s0.e4" interaction="false"/>
  </sentence>
```

Figure 1-1: Annotations of one sentence from the DDIExtraction-2011 corpus.

1-2-2 The 2nd Drug-Drug Interaction Extraction 2013 Shared Task Challenge (DDIExtraction-2013)

The second shared task challenge on drug-drug interactions, DDIExtraction-2013 [25], was offered as part of the 2013 International Workshop on Semantic Evaluation (SemEval-2013).[5] There were 14 participating teams for the task. In contrast to the 2011 challenge, the 2013 challenge also included the task of drug name recognition and classification, although it was set up as distinct from the task of DDI extraction, which was itself an extension of the 2011 task. Thus, the following two subtasks were offered:

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

1. Recognition and classification of drug names
2. Extraction and classification of drug-drug interactions

In addition to extracting drug-drug interactions in the second task, the task called for their further classification into one of five different classes: *advise*, *effect*, *mechanism*, *int*, and *none*. The task emphasized the importance of recognizing what is being asserted about a drug interaction instead of just identifying an interaction. The five classes are defined below, with examples provided for illustration:

1. *Advise*: 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**.");
2. *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**, …");
3. *Mechanism*: the sentence describes a pharmacokinetic mechanism (e.g., "**Grepafloxacin** is a competitive inhibitor of the metabolism of **theophylline**.").
4. *Int:* the sentence mentions a drug interaction but doesn't provide any additional information about its nature (e.g., "The interaction of **omeprazole** and **ketoconazole** has been established.").
5. *None:* the sentence does not show an interaction between the two drugs;

As the goal of our work, and this thesis, is to focus on the extraction and classification of drug-drug interactions, we participated in the second sub-task of the challenge.

The training corpus provided in the challenge contains 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 annotated with four types of pharmacological entities and five types of DDIs, as described above. Figure 1-2 illustrates annotations of drug entities and drug interactions for three sentences in a document. All possible drug-pairs in a sentence are annotated as either "true", for an interacting drug-pair, or "false", for a non-interacting drug pair. Each annotated drug pair is treated as an instance for training, and from the perspective of training, drug-pairs annotated as "true" belong to the *positive* class (or the set of positive instances), while those annotated as "false" belong to the *negative* class (or the set of negative instances). Positive instances are further annotated as one of the four interacting types described above, namely, *advise*, *effect*, *mechanism*, *int*.

```
−<document id="DDI-DrugBank.d372">
  −<sentence id="DDI-DrugBank.d372.s0" text="Cytadren accelerates the metabolism of dexamethasone;">
      <entity id="DDI-DrugBank.d372.s0.e0" charOffset="0-7" type="brand" text="Cytadren"/>
      <entity id="DDI-DrugBank.d372.s0.e1" charOffset="39-51" type="drug" text="dexamethasone"/>
      <pair id="DDI-DrugBank.d372.s0.p0" e1="DDI-DrugBank.d372.s0.e0" e2="DDI-DrugBank.d372.s0.e1" ddi="true" type="mechanism"/>
    </sentence>
  −<sentence id="DDI-DrugBank.d372.s1" text="therefore, if glucocorticoid replacement is needed, hydrocortisone should be prescribed.">
      <entity id="DDI-DrugBank.d372.s1.e0" charOffset="14-27" type="group" text="glucocorticoid"/>
      <entity id="DDI-DrugBank.d372.s1.e1" charOffset="52-65" type="drug" text="hydrocortisone"/>
      <pair id="DDI-DrugBank.d372.s1.p0" e1="DDI-DrugBank.d372.s1.e0" e2="DDI-DrugBank.d372.s1.e1" ddi="false"/>
    </sentence>
  −<sentence id="DDI-DrugBank.d372.s2" text="Aminoglutethimide diminishes the effect of coumarin and warfarin.">
      <entity id="DDI-DrugBank.d372.s2.e0" charOffset="0-16" type="drug" text="Aminoglutethimide"/>
      <entity id="DDI-DrugBank.d372.s2.e1" charOffset="43-50" type="group" text="coumarin"/>
      <entity id="DDI-DrugBank.d372.s2.e2" charOffset="56-63" type="drug" text="warfarin"/>
      <pair id="DDI-DrugBank.d372.s2.p0" e1="DDI-DrugBank.d372.s2.e0" e2="DDI-DrugBank.d372.s2.e1" ddi="true" type="effect"/>
      <pair id="DDI-DrugBank.d372.s2.p1" e1="DDI-DrugBank.d372.s2.e0" e2="DDI-DrugBank.d372.s2.e2" ddi="true" type="effect"/>
      <pair id="DDI-DrugBank.d372.s2.p2" e1="DDI-DrugBank.d372.s2.e1" e2="DDI-DrugBank.d372.s2.e2" ddi="false"/>
    </sentence>
</document>
```

Figure 1-2: Annotations of three sentences from a document in the DDIExtraction-2013 corpus.

Table 1-1 shows the number of instances for each of the five DDI types in the training set, grouped further into positive and negative classes. We note again that each instance is associated with a single pair of drugs. For example, a sentence with 4 drugs contains 6

instances, corresponding to 6 distinct drug pairs with a potential for interaction. Thus, while the entire training corpus contains 6976 sentences, the number of training instances as shown in the Table is much higher, i.e., 24891 instances.

The test set for the task, used only during the evaluation period in the challenge, includes 33 Medline abstracts and 158 DrugBank documents, containing 1299 sentences and 5519 drug pairs (instances). As the test set had not been made available until the time of writing of this thesis, the experiments we conducted after the challenge, and report on here, are done using only the training data, in particular by splitting the training data into a training set (90%) and a test set (10%).

Table 1-1: Distribution of drug-pair instances for each class in the DDIExtraction-2013 corpus. Classes are categorized in two super classes: *positive* and *negative*, to indicate presence and absence of interaction, respectively.

Type		**DrugBank**	**Medline**	**Total**
Positive	*Advise*	819	8	827
	Effect	1548	152	1700
	Mechanism	1260	62	1322
	Int	178	10	188
Negative	*None* (non-interacting drug-pairs)	19479	1375	20854
Total		**23,284**	**1607**	**24891**

In this thesis, we describe our system for extracting and classifying drug-drug interactions from biomedical text, utilizing the training corpus provided for the DDIExtraction-2013 shared task challenge. Our approach combines machine-learning methods with 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

observed in the data, via the use of a two-stage 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 classification. Although we applied weighted SVM as the classifier for the DDI-2013 competition, here we report additional experiments with several other classifiers to assess if a classifier other than SVM may be better suited to the task. Our experiments indicate that SVM is the best fit for both stages. We also describe our experiments with exploring additional features for the classifier, specifically those exploiting syntactic information obtained from sentence parse trees. Finding effective features and utilizing them in the system resulted in improving the F-measure by 12%, when compared to the results obtained in the competition.

The thesis is organized as follows. In Chapter 2, we describe the related work on drug named entity recognition as well as DDI extraction and classification, particularly discussing all systems that participated in the DDIExtraction 2011 and 2013 challenges. In Chapter 3, we describe our method, the classifiers used in each stage, their features, and post processing. In Chapter 4, we present the evaluation and results. Error analysis, discussion, and future work are presented in Chapter 4.

Chapter Two:

Related Work

2-1 Introduction

In this chapter, we review the studies that have been conducted to extract and classify drug interactions from text. Most studies on this problem have been carried out as part of the DDIExtraction-2011 and DDIExtraction-2013 challenges, so we focus on these here. First, however, we review recent Drug Named Entity Recognition methods, since their outputs are vital for DDIExtraction systems, although our own work uses gold standard annotations of drug names in the corpus, provided as part of the challenge.

2-2 Drug Named Entity Recognition

The first step for extracting Drug-Drug interactions from text involves detecting drug names. Needless to say, performance of Drug Named Entity Recognition (NER) system has an impact on the performance of DDI extraction systems. Three common approaches for Drug NER are dictionary-based, rule-based and machine learning methods.[6] For creating a dictionary that contains a list of drug names and their property, DrugBank [15] is a useful source. It is an open access, web-enabled database that contains structural, physicochemical, pharmacological and target information of approximately 4300 substances, of which 1177 are approved drugs. Another useful resource for drugs is Daily Med, which presents all drug labels (Package Inserts). It is created by the U.S. National Library of Medicine. In fact, the main goal of package inserts is to provide useful information about drugs to physicians and help them to prescribe drugs appropriately.

[6] When comparing the performance these methods it is worth keeping in mind that some of these methods were designed for detecting general chemical names, which is a harder task rather than drug NER and therefore a possible reason for the poorer performance.

Several tools, based on machine learning techniques, have been developed to identify drug names in text. One of them is cTAKES [26], an open source system to extract medical information from clinical text. It has several components, including named entity recognition, which covers NER for drugs, in addition to other entity types.

One of the recent studies for identifying and classifying drug names is done by Segura et al [27]. Their rule-based system combines information from several resources such as UMLS MetaMap Transfer, World Health Organization, and International Nonproprietary Names Program. Besides identifying pharmaceutical substances, the system is able to detect drug names.

Hettne et al. [28] have developed a dictionary that detects small molecules and drugs. They combined information from UMLS, MeSH, ChEBI, DrugBank, KEGG, HDMB and ChemlDplus. They also used rule-based term filtering. They report a precision of 0.67 and recall of 0.40.

ChemSpot [29] is a Named Entity Recognition tool for identifying mentions of chemicals in text. It detects trivial names, drugs, abbreviations, molecular formulas and IUPAC entities. It uses CRF (Conditional Random Fields) and a dictionary-based approach. It obtained 68.1% F-measure on the SCAI corpus. There were five systems [25] participating in the Drug NER task in DDIExtraction-2013, variously using dictionary based and machine learning techniques.

2-3 Drug-Drug Interactions

As we noted earlier, most DDI-Extraction studies were conducted as part of the DDI-Extraction 2011 and 2013 challenges. In this section, we review these studies.

2-3-1 DDIExtraction 2011

Segura et al [30] report one of the first attempts to extract drug-drug interactions from the biomedical literature. They used a hybrid method that combines shallow parsing and syntactic simplification with pattern matching. The UMLS MetaMap tool (MMTx) is used to provide shallow syntactic parsing and a set of domain-specific lexical patterns were developed to extract DDIs. Separately, in later work, they utilized a supervised machine learning approach to identify DDIs [31], while also creating a DrugDDI corpus for evaluating their approach. Their SVM classifier achieved 0.51 precision, 0.72 recall and F-measure of 0.60.

Mata et al [32] developed a Machine Learning system for DDI extraction that achieved an F-measure of 0.4702. For developing the system, they used around 600 features, such as keyword before first drug, keyword after second drug, keyword between drugs, and number of words and phrases between drugs. They explored four classification algorithms: RandomForest, NaïveBayes, SMO, and multiBoosting. Their best result comes from RandomForest.

Garcia et al [33] built a Machine Learning system based on bag of words and pattern extraction. 1,010 words with a high gain ratio were collected and used as a "bag of words" feature, in addition to word categories to reflect the structure of the sentence, including subordinators, independent markers, appositions, coordinators, absolute, quantifiers, negations, etc. They also used Maximal Frequent Sequences (MFS) as a feature. A sequence is defined as an ordered list of elements, in this case, words. A sequence is maximal if it is not a subsequence of any other; that is, if it does not appear in any other sequence in the same order. All MFS from the training corpus were extracted,

with length between 2 and 7, and appearing in at least 10 sentences. Several classifiers were explored in this work, including Support Vector Machines, Decision Trees and multiple ensemble classifiers such as Bagging, MetaCost and Random Forests. Their best choice was Random Forest with 100 iterations and 100 attributes per iteration, with an F-Measure of 0.5829.

Thomas et al [34] have used Ensemble learning for DDI extraction. Their single best single classifier achieved an F-Measure of 0.63 and the best ensemble achieved 0.65. They used three kernel based approaches (APG, kBSPS, and SL) and case-based reasoning (Moara).

Bjorne et al [35] presented a DDI system that explored both SVM and regularized least-squares classifiers. They obtained 0.62 F-measure on DrugDDI. Minard et al [36] also presented a system based on SVM by using LIBSVM and SVMPerf tools. They reported a 0.5965 f-measure on DrugDDI.

Chowdhury et al [33] participated in the DDIExtraction 2011 challenge and evaluated a range of new composite kernels for DDI. These kernels combine different combinations of mildly extended dependency tree (MEDT) kernel, phrase structure tree (PST) kernel, local context (LC) kernel, global context (GC) kernel and shallow linguistic (SL) kernel. The best result is an F-Measure of 0.6370 by combining MEDT, PST and GC kernels. They used the UMLS SPECIALIST lexicon tool to normalize tokens to avoid spelling variations and to provide lemmas. They also used dependency parse trees for corresponding sentences.

Karnik et al [37] presented a DDI extraction system that used all paths graph kernel. The system didn't work well on DrugDDI corpus and it obtained a 0.16 F-measure. But F-measure for a clinical pharmacokinetic DDI corpus was 0.658.

2-3-2 DDIExtraction 2013

In the 2013 challenge, the system with the highest F-Measure is proposed by the FBK-irst team [38]. Their system is a multi-phase relation extraction system. They used two separate phases for DDI extraction and classification. For DDI extraction, they removed less informative sentences and instances, and then trained a system on the remaining instances. A hybrid kernel classifier that contains a feature based kernel, a shallow linguistic kernel, and a Path-Enclosed Tree kernel is used in the first step. For classification of DDI, they trained 4 separate models for each class (one vs. all the other classes).

The innovative part of this system is detecting "less informative sentences", where a sentence is considered less informative if all drugs in a sentence fall under the scope of a negation cue (such as *not*). A negation detector system (focused on a limited set of negation cues, such as *no*, *n't* and *not*) is used to identify and filter the less informative sentences. The remaining sentences are classified with the SVM Light-TK toolkit (Moschitti, 2006)[39], utilizing the Charniak-Johnson reranking parser [40], a self-trained biomedical parsing model [41], and the Stanford parser [42]. On the DDI-DrugBank test dataset, they obtained 0.68 F-Measure and on the DDI-Medline test dataset, 0.40 F-Measure.

The WBI-DDI team [43] presented a two-step system, like the first system in this competition, that splits the step for extracting DDIs step from that of classifying DDIs.

For extracting DDIs, an ensemble approach is applied, which combines the output of five different classifiers via majority voting. The framework for this ensemble approach is provided in [44]. All-Paths Graph [45], shallow linguistic [46], subtree [47], subset tree [48], and spectrum tree [49] method are the classifiers used in the ensemble method. Each classifier uses different sets of features, but most of them used part-of-speech tags, constituent parse tree, and dependency parse tree information. In the second step, the subtype prediction of Turku Event Extraction System [35] is applied.

For pre-processing, this system uses the Charniak-Johnson PCFG parser [40] with a self-trained re-ranking model augmented for biomedical texts [41]. Like most teams in the competition, the drug entity names are replaced with a generic string to ensure the generality of the approach [50].

This approach achieved the second rank in the competition, with 0.61 F-Measure on the DDI-DrugBank test dataset and 0.35 F-Measure on the DDI-Medline test dataset.

The UTurku team [51] developed a machine learning system based on the Turku Event Extraction System (TEES) [35]. TEES is an NLP tool for event and relation extraction based on SVM. It considers part-of-speech tags, dependency chains, dependency path N-grams, entities, and external resources such as hypernyms in WordNet. For this task, Bjorne et al. used deep syntactic parsing to generate large graph-based feature sets. They parsed the corpus with TEES and extracted most of their syntactic features from the shortest path of dependencies between two main drugs, such as N-grams and governor–dependent information for dependencies.

The significant difference between this system and the others is in using external resources. This system derived some features from external resources such as DrugBank and MetaMap. They trained three systems with different sets of features:

1. Features extracted from the text as baseline
2. Adding extracted features from DrugBank to the baseline
3. Adding extracted features from MetaMap to the baseline.

Their results showed that the external features, especially from DrugBank, increased the performance, because they extracted DDIs from DrugBank and used them as a feature in the system. However, MetaMap didn't improve the performance, although their results show that MetaMap is useful for the Drug NER task.

They obtained 0.61 F-Measure on the DDI-DrugBank test dataset and 0.23 F-Measure on the DDI-Medline test dataset.

The NIL-UCM team [52] presented a SVM classifier with a linear kernel and a rich set of lexical, morphosyntactic and semantic features. They experimented with two approaches. In the first approach, they extracted and classified all DDIs in one step, as a 5-class classification problem. But in the second approach, they extracted DDIs in one step, and then classified them into 4 DDI classes in the next step. Most of the teams in the competition applied the second approach, separating the extraction step from the classification step.

Features in this system included word features (such as words between drugs, three words before first drug, and so on), morphosyntactic features (such as POS), constituency parse tree features (such as shortest path between drugs, shortest path between first token in the

sentence and first drug, etc.), conjunction features, verb features, and negation features. They applied feature selection approaches and information gain ranker for selecting the best features.

Only this team separated the DrugBank data from the Medline data and trained two separate SVM systems for each. However, this approach didn't obtain a good overall result compared to the other approaches.

A better result was obtained with the second approach, which separated the extraction phase from the classification phase. Like the other teams, they obtained a better result on the DrugBank data rather than the Medline data. The authors attribute the reason for the poorer performance to the fact that the Medline corpus has fewer words as compared to the DrugBank corpus. This is also suggested by Chowdhury et al. [38].

In this system, Paice/Husk Stemmer [53], Stanford parser [42], NegEx[7] and Weka [54] are used. Their F-Measure on the DDI-DrugBank test dataset is 0.56 and on the DDI-Medline test dataset is 0.12.

The system presented by the UC3M team [25] is based on shallow linguistic (SL) kernel methods. The system contains three steps: pre-processing, DDI extraction, and DDI classification. They submitted two runs to the competition. The first run was based on linguistic information and the second one on semantic information. For the pre-processing step, GATE analyzer[8] and Stanford parser [42] are applied to obtain POS and lemmatization. Also, multiword entities are pre-processed to keep words related to same

[7] http://code.google.com/p/negex/

[8] http://gate.ac.uk.

concept together. For example, they unified “beta-adrenergic receptor blocker” into a singleton word “beta-adrengic_receptor_blocker” as type NNP, whereas the Stanford parser would have processed the phrase with three different tags and phrase labels. For the third step, they trained four systems for each class. The only semantic information used is the ATC code value. They obtained a higher result with the system that used linguistic information. However, because they just explored one semantic feature, we can’t conclude anything about the (non-)importance of using semantic information for this task. They obtained 0.56 F-Measure on the DDI-DrugBank test dataset and 0.26 F-Measure on the DDI-Medline test dataset.

Our team, UWM-TRIADS [55], presented a system based on SVM and rule-based post-processing. We explored two approaches, one separating DDI extraction from DDI classification, and the other doing both in one step. We obtained a better result from the first approach, with a two-stage classifier. We used SVM as the classifier in both stages. Because of the unbalanced distribution of the classes, we assigned different weights to each class. Our SVM features exploited stemmed words, lemmas, bigrams, part of speech tags, verb lists, and similarity measures, among others.

Also, we developed a set of post-processing rules after each stage. The post-processing rules improved our results.

In this system, we used LibSVM [56], Weka [54], Stanford NLP tool [42], [57], Dragon toolkit [58] and WordNet [59].

We obtained 0.48 F-Measure on the DDI-DrugBank test dataset and 0.34 F-Measure on the DDI-Medline test dataset.

The SCAI team [60] presented a machine learning system which utilizes lexical, syntactic and semantic feature sets. Like the other teams, this system contained two steps, extracting DDIs and classifying DDIs. This system used an ensemble classifier in the first step, but for the second step, it just applied some post-processing rules.

The set of features for the classifier contained lexical, syntactic dependency, and semantic features. Their feature set contained most of the features that are used by the other teams, also considering negation words in sentences. LibLINEAR, Naïve Bayes and Voting Perceptron classifiers are used in the ensemble method. After extracting DDIs, they applied a post-processing step to classify DDIs into 4 classes. For this step, they generated 4 lists of relation trigger words, manually. Different priorities are assigned to each class, for cases when a sentence contained trigger phrases from different classes.

They achieved 0.46 F-Measure on the DDI-DrugBank test dataset and 0.26 F-Measure on the DDI-Medline test dataset. They used a rich set of features in the first step; this poor result shows that using only post-processing rules for classifying DDIs is not a good approach.

The UColorado-SOM team [61] presented a machine learning system based on SVMs. Morphosyntactic, lexical and semantic features were used to train the system. They approached the task as a binary classification task by applying one-vs-all multi-class classification techniques. In essence, the system extracted and classified DDIs at the same time, which appears to be the reason for the poor result.

LIBSVM [56], GENIA, TEES [35] and OpenDMAP [62] are used in this system. They obtained 0.42 F-measure on the DDI-DrugBank test dataset and 0.27 F-measure on the DDI-Medline test dataset.

2-3-3 Other Approaches

In contrast to the classification of DDIs in the shared task competitions, there are several studies that classify DDIs in terms of their "mechanism of interaction", distinguishing between Pharmacodynamic (PD) interactions and Pharmacokinetic (PK) interactions. Pharmacodynamic interactions include the concurrent administration of drugs having the same (or opposing) pharmacologic actions, and alteration of the sensitivity or the responsiveness of the tissues to one drug by another. Many of these interactions can be predicted from knowledge of the pharmacology of each drug. The change in an organism's response on administration of a drug is an important factor in pharmacodynamics interactions. Pharmacokinetics refers to the study of the absorption, distribution, metabolism and excretion (ADME) of bioactive compounds in a higher organism. In a Pharmacokinetics interaction, modifications in the effect of a drug are caused by differences in the absorption, transport, distribution, metabolization or excretion of one or both of the drugs compared with the expected behavior of each drug when taken individually.

Tari et al. [63] evaluated a rule-based algorithm for extracting pharmacokinetic DDIs from papers and abstracts in the scientific literature. In this study, the authors distinguished between explicit DDIs (statements indicating a direct observation of a PK effect from a given drug combination) and implicit DDIs (DDIs that can be inferred based on claims about drug metabolic properties extracted from scientific texts). The algorithm was run over more than 17 million Medline abstracts and the output DDIs were compared with DrugBank drug interactions. The recall of the algorithm was very low, but

their study showed that 78% of the DDIs extracted were valid. These results illustrated that DDIs in DrugBank aren't complete.

Boyce et al. [64] presented a tool to extract PK DDI. They manually created a corpus of Federal Drug Administration approved drug package insert statements, containing 592 PK DDI. Then they implemented and evaluated three different classifiers using machine-learning algorithms. Besides classifying PK DDI in the corpus, their system classified statements by their modality. They evaluated SVM, Jrip, and J48, and their best result was 0.859 F-measure with SVM.

Chapter Three:

Methods

3-1 Introduction

In this chapter, we describe our approach to extract and classify drug interactions from biomedical text. Our system classifies each drug pair into 5 classes – *advise*, *effect*, *mechanism*, *int* and *none*. A major challenge in this task is posed by the unbalanced distribution of the classes. First, considering just the positive vs. negative classes, just 19.3% (4037/20854) of drug pairs are in the positive class. Furthermore, the four types 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). To handle this problem, we propose a two-stage classification approach.

In the following sections, we provide details about our approach and discuss its advantages, including pre-processing steps, the set of features explored in our machine learning method, and the post-processing rules developed for further manipulation of the result of machine learning.

3-2 Pre-processing Steps

Before classification, all sentence instances in the corpus were pre-processed in order to clean and normalize the corpus as well as to extract features for machine learning. We utilized existing NLP tools for several steps in the pre-processing. The following steps describe the pre-processing:

- 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 classification in the instance (main drugs), and the other used for all other drug mentions (additional drugs).
- All numbers were normalized by replacing them with the same string.
- Sentences with less than two drug names were removed, since the system is tasked with detecting and classifying drug interactions between two drugs.
- Stop words and punctuation were removed. We used different stop word lists to compare how the number of stop words affect the system. However, as stop words between two main drugs can contain useful information as an indicator for interaction, stop words in this context were retained.
- Part of speech (POS) tags were obtained with the Stanford NLP tool [57].
- Words were stemmed with the Porter Stemmer [65].
- Words were lemmatized with Dragon tool [58].
- Synsets for words were obtained using WordNet [59].
- We developed and implemented a tool to detect negations in sentences. The tool highlights negated sentences and also identifies negation indicators such as *not*. The negation tool will be described later below.
- Phrase structure parse tree of sentences were obtained with the Stanford NLP tool [42]. We explored multiple types of information from parse trees as features in the classifier, including syntactic path between the main drugs and whether or not both main drugs appear in the same clause.

3-2-1 Negation detector

The Negation Detector tool mentioned above was developed by us. It utilizes the machine-learning approach of Hidden Markov models (HMMs). Hidden Markov Model is the stochastic analog of finite state automata. An HMM is defined by a set of states and a set of transitions between them. Each state has an associated emission distribution, which defines the likelihood of a state to emit various tokens. The transitions from a given state have an associated transition distribution, which defines the likelihood of the next state, given the current state.

We generated a HMM and trained the negation detection model with the BioScope[9] corpus [66]. We used a java implementation of HMM, called Jahmm[10]. The BioScope corpus consists of medical and biological texts annotated for negation and speculation, with the annotation encoding negation and speculation keywords and their scopes. It contains more than 20,000 manually annotated sentences from clinical notes and published biological articles. For our negation detector tool, we only considered the negation annotations of the corpus.

A HMM model is trained via sequences of observations. In our model, we considered POS tags as observations. In particular, we replaced all non-negated words with POS tags and generated the sequences of observations. Figure 3-1 shows our HMM model. It has two states, positive and negative and includes 72 observation, 35 POS tags and 37 negated words. The accuracy of the system is 96.44% and F-measure is 92.03% in negation sentence detection.

---

[9] http://www.inf.u-szeged.hu/rgai/bioscope

[10] https://code.google.com/p/jahmm/

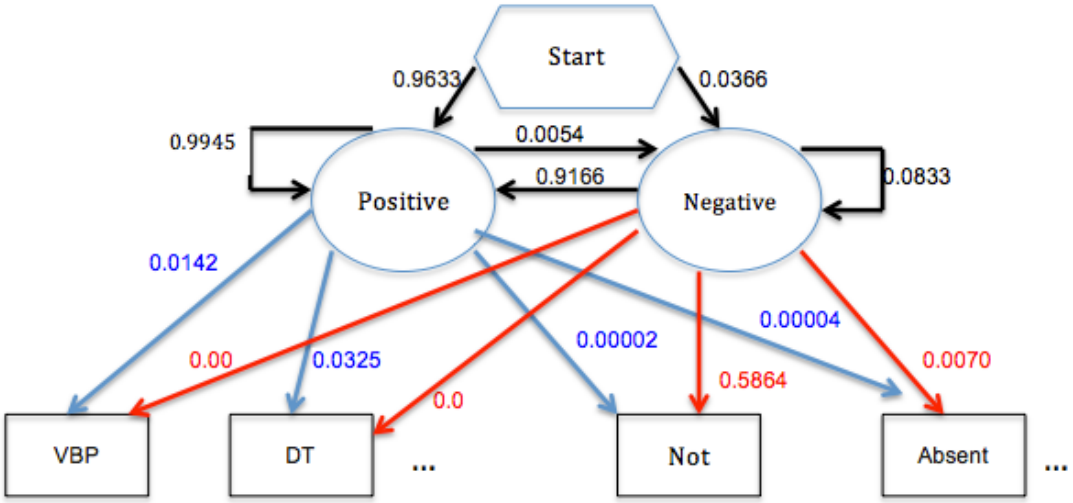

Figure 3-1: This Hidden Markov Model is used in the negation detector system. This HMM consists two states, positive and negative. Each rectangle shows one of observations that can be a POS tag or a word that appeared in the negated parts of the sentences.

3-3 Two-stage classification

The architecture of our system is illustrated in Figure 3-2. The system comprises two classifiers in separate stages. In the first stage, we train a binary classifier to classify drug pairs into positive and negative classes. Then, in the second stage, we consider only those instances that are classified as positive by the first classifier, and classify them into one of four types within the positive class – *advise*, *effect*, *mechanism*, and *int* – using a multi-class classifier.

A two-stage classifier offers a distinct advantage over a one-stage classifier for the DDI data set, not just because it is highly skewed towards one class – the negative class – but also because this majority class is clearly semantically distinct from the other positive classes. Therefore, 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. Using a two-stage classification approach also allows us to explore different classifiers for each stage and find the best fit for each of them separately, by pursuing advantageous approaches for binary classification on the one hand and multi-class classification on the other hand.

After pre-processing, the remaining sentences contain two or more drugs. In the first stage, we need a binary classifier to classify each drug pair as positive or negative. The following is an example of a sentence with drug names highlighted.

- ***"Catecholamine-depleting drugs****, such as* ***reserpine****, may have an additive effect when given with* ***beta-blocking agents****."*

As the sentence has three drug names, the system needs to consider DDI between the following three drug pairs:

1- Catecholamine-depleting drugs and ***reserpine***

2- Catecholamine-depleting drugs and ***beta-blocking agents***

3- reserpine and ***beta-blocking agents***

At this point, the DDI extraction task is carried out via a binary classifier. If the classifier predicts a DDI between a pair, then it classifies the pair as positive, and otherwise as negative. As we want to pass only the positively classified instances from the first stage to the second stage classifier, we favor the positive class in the first stage. For this purpose, if the classifier allows us to assign weights to each class (e.g., SVM) we assign a high weight to the positive class. This results in a relatively high number of false

positives for the positive instances, which we attempt to reduce with a set of post-processing rules before sending them to the second stage classifier.

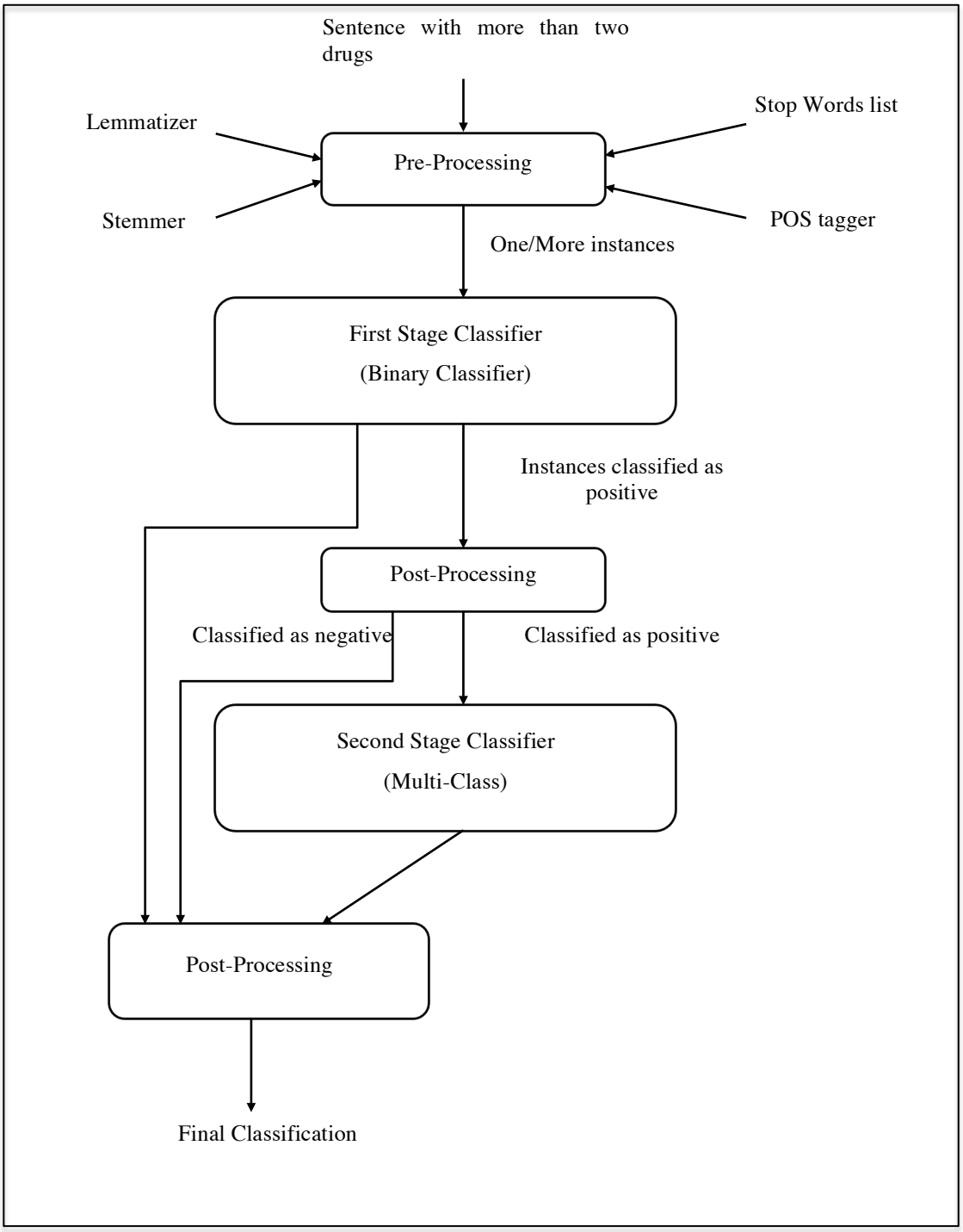

Figure 3-2: DDI Extraction and Classification System architecture.

We should add that for training the first classifier we used all the instances in the corpus but for training the second classifier, we just used the positive instances that classified into Advice, Effect, Mechanism, and Int.

3-3-1 Features

To train the classifiers, we utilized a variety of features to investigate their effectiveness and choose the best combination of features for classification. In this section, we explain these features by separating them into two categories: *features per sentence* and *features per drug-pair instances*. Recall that since one sentence can have more than two drug names, an instance of the sentence is generated for each drug pair.

Features per sentence

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

- **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.
- **Mutual Information [67]:** Instead of using all words that appeared in the corpus, we apply mutual information as a feature selection approach to choose informative words. Mutual information for term t and class c is calculated with the following formula:

$$P(t,c) = \left(\frac{N_{t,c}}{N} * \log_2^{\left(\frac{N_{t,c}*N}{N_t*N_c}\right)}\right) + \left(\frac{N_{t,\bar{c}}}{N} * \log_2^{\left(\frac{N_{t,\bar{c}}*N}{N_t*N_{\bar{c}}}\right)}\right)$$

$$+\left(\frac{N_{\bar{t},c}}{N} * \log_2^{\left(\frac{N_{\bar{t},c}*N}{N_{\bar{t}}*N_c}\right)}\right)+\left(\frac{N_{\bar{t},\bar{c}}}{N} * \log_2^{\left(\frac{N_{\bar{t},\bar{c}}*N}{N_{\bar{t}}*N_{\bar{c}}}\right)}\right)$$

Where, N: Number of all instances, $N_{t,c}$: Number of instances in c that contain t, $N_{t,\bar{c}}$: Number of instances that contain t but not in the class c, N_t : Number of instances that contain t, … We rank the words based on the Mutual Information score and use 100/200/500/1000 top words.

- **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
- **Number of words:** This feature represents the total number of words in the sentence
- **Number of drug mentions:** This feature represents the total number of drug mentions in the sentence.
- **Cosine similarity between centroid vector of each class and the instance:** Inspired by the vector space [67] Information Retrieval approach, we add new features to represent the cosine similarity between a sentence and the centroid of normalized vectors for sentences assigned the class C. Cosine similarity is calculated based on modified tf*idf. We compute modified tf*idf for a word w, based on the following formula:

$$\text{Tf} * \text{idf} = \text{Count}(\text{w in instance}) * \log_2^{\left(\frac{\text{number of all instances}}{\text{number of instances contain (w)}+1}\right)}$$

TF is the number of times the word occurs in the instance. IDF is the logarithm of number of all instances divided by the number of instances that contain the word. To calculate the centroid vector for class C, a vector is created for each instance in class C by giving each word in the instance 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 as a feature.

Features per instance (each 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.

- **Number of words between two main drugs:** This represents the total number of words between the two main drugs.
- **Number of drugs between two main drugs:** This represents the total number of additional drugs appearing between the two main drugs.
- **Number of verbs:** We use 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 count the number of verbs in each section, and use them as three different features.
- **Number of verbs using class-specific verb lists:** For each class, we extract two lists of verbs. The first list contains verbs that appeared in just that class but not in the others. Thus, the set of verbs extract 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 create two features to represent the number of verbs from each of these lists that appeared in the section.

- **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.
- **Path between two main drugs:** Path between two main drugs in the parse tree is another feature in our system. Because syntactic paths are in general a sparse feature, we reduced the sparsity by collapsing identical adjacent non-terminal labels. E.g., NP-S-VP-VP-NP is converted to NP-S-VP-NP. This technique decreased the number of paths by 24.8%.
- **Negation:** To consider negation in the instance, three features are defined that respectively indicate negation before the first main drug, between two main drugs, and after the two main drugs.
- **Clause Boundary:** This feature shows that both main drugs are in the same clause or not. The system utilizes the parse tree to set the binary value for this feature.

3-3-3 Post processing

As described before, we have two sets of post-processing rules for each stage of the system. Here, we describe these rules, developed on the basis of observations in the 90% of the competition training data (that was used as the training set in our experiments). In the next chapter, we evaluate the effectiveness of these rules.

3-3-3-1 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. The following describes the rules, with examples:

1. An instance is classified as negative if both drug mentions have the same name, since a drug cannot interact with itself. In the following instance, one drug is appeared twice. So, the system considers them as a drug pair. If the classifier in the first stage classifies them as positive, the post-processing step updates its label to negative.
    - *"In controlled clinical trials of **AUGMENTIN XR**, 22 patients received concomitant allopurinol and **AUGMENTIN XR**.*
2. An 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."*
3. An 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 own class as a whole. Drug class names were obtained from a table provided by the FDA.[11] In the example below, "MAOI" is the drug class name for "isocarboxazid".

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

- *"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."*

4. An instance is classified as negative if "," or ", and" appears between the two main drug mentions, and is accompanied by an additional drug mention. The rule 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."*
5. An instance 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 in particular, 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."*
6. An 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**."*

3-3-3-2 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 use several syntactic and lexical cues to identify set mentions of drugs. Then, since the classifiers 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 apply 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 3-2.

Chapter Four:

Results and Conclusion

4-1 Introduction

In this chapter, we first present our implementation of the system, including the libraries and tools that we utilized. Metrics used to evaluate system performance are then presented, followed by the results, where we also discuss the impact of various features on the system using SVM. Then, the performance of different classifiers in each of the two system stages is evaluated and the best system introduced. We also present results of applying the post-processing rules to the output of each stage. We finish with the error analysis, conclusions, and discussion of future work.

4-2 Implementation

As there are many Natural Language Processing and Machine Learning tools and libraries in Java, we used this programming language to implement our method. Some of these tools such as Weka [54] are applied in classification tasks, but Weka is not designed to directly handle the steps in our system architecture, which consists of two classifiers with the output of the first classifier serving as input to the second one, and a set of post-processing rules applied after each stage. Therefore, we implemented a tool that, besides using available NLP libraries, handles our two-stage approach for classification.

The following lists and describes the existing libraries and tools utilized in our system. (These were also mentioned briefly in Chapter 3.) We have used these tools for various tasks in our system, including machine learning, pre-processing, and feature extraction.

- **Weka:** Weka [54] is a collection of machine learning algorithms available as a java-based software package, with graphical user interfaces providing easy access to its functionalities. We used Weka to apply some of its classifiers in our method.

The input to Weka has to be in a specific format, called the Attribute-Relation File Format (Arff). An Arff file is a text file created by declaring all attributes (i.e., the feature names) in the file, followed by a list of the data instances. Each instance is represented as a vector of values for the attributes, with one value provided for each attribute. The last value of the instance vector indicates the class label of the instance. To utilize Weka for training our classifiers, we created an Arff file using the instances in the data and the features described in Chapter 3.

- LibSVM [56]: LibSVM is software library for Support Vector Machines, implemented in Java, that allows users to apply SVM for their applications. We used this library to train the SVM classifier.
- Stanford NLP tool: This library is used to obtain POS tags [57] and parse trees [42] for sentences. The tool contains several NLP algorithms such as part-of-speech (POS) tagger, named entity recognizer (NER), parser, and the coreference resolution system. It is provided in different languages such as Java, Perl, Python, Ruby, Sacala, and Clojure.
- Porter Stemmer [65]: This is a library used for stemming words in the sentences. Stemmed words were used as a feature in our classifier.
- Dragon tool [58]: We used this tool to obtain word lemmas, which were used as a feature in our classifier.
- WordNet [59]: As described in Chapter 3, we created a list of synonyms for verbs in each class. We used WordNet to obtain synonyms.

4-3 Metrics

We used the standard metrics of Precision, Recall, and F-measure to evaluate the performance of our system. To compute these metrics, the organizers of the DDIExtraction 2013 Challenge provided a code in Java that takes as input a text file of predictions and calculates the precision, recall and F-measure. In these metrics, a DDI is correctly detected if the system assigned the correct class label to it. A prediction is correct if both extraction label (Yes, None) and classification label (*Advise*, *Mechanism*, *Effect*, *Int*) are correct. To clarify the meaning of these metrics, let's look at the meaning of precision for type *Advise*:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

TP indicates the true positives, i.e., the number of truly detected pairs of the *advise* class class and FP indicates the false positives, i.e., the number of pairs that are predicted as *advise* incorrectly.

For each run in the results discussed below, we present these three metrics. F-measure is calculated with the macro-averaged method[12], i.e., by taking precision to be the average of the precision calculated for each type, and similarly for recall.

4-4 Results

In our experiments, we first explored different combinations of features (described in Chapter 3) with SVM in order to identify the best performing feature set. We then used this feature set to further explore the performance of additional classifiers, thus

[12] http://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/semeval-2013-task-9_2-evaluationmetrics.pdf

identifying the best machine learning classifier for the task. Post-processing rules were applied to the output of the best performing classifier. As the test set had not been made available until the time of writing of this thesis, the experiments we conducted after the challenge are done by using only the training data, by splitting the training data into a training set (90%) and a test set (10%). In this section, we present and discuss the results of all our experiments. For each experiment, precision, recall and F scores are given.

4-4-1 Feature Selection

To determine which features are most useful in the system, we added them incrementally to our classifiers and examined their effect on system performance. We used SVM for identifying the best performing features, in both stages of the system.

In the first experiment, we explored the role of words as features. We compared the effect of using all words as features with using only informative words, where informative words are chosen by calculating the Mutual Information (MI) score for words in each class (described in 3-3-1). We further explored different cut-offs for choosing informative words in each class with high MI score: 100, 200, 500 and 1000, which yielded 296, 476, 940, and 1417 unique words, respectively, as features. Table 4-1 presents the results for this experiment. The first row shows the result with all words used as features, whereas the remaining rows show the result of using informative words as features.

Table 4-1: Results of using words with high mutual information score compared to all words.

Features	# Of features	Precision	Recall	F-Measure
All Words	1599	0.22	0.74	0.34
100 words of each class with high MI	296	0.33	0.68	0.34

200 words of each class with high MI	476	0.32	0.68	0.34
500 words of each class with high MI	940	0.28	0.70	0.34
1000 words of each class with high MI	**1417**	**0.30**	**0.74**	**0.37**

As Table 4-1 illustrates, using informative words to train the classifiers performs better than using all words. Furthermore, with informative words, the best result was obtained with 1000 words. Therefore, in all remaining experiments, we used 1000 words of each class with high MI score, with a total of 1417 unique words.

In the next experiment, we evaluated the impact of POS as a feature in combination with the 1000 high MI words. POS is a binary feature for word POS tags and indicates the presence or absence of each POS between the two main drugs. The second row of Table 4-2 illustrates the results of adding the POS feature. The first row repeats the scores for the best result from Table 4-1, The second row shows that adding POS markedly increases the F-Measure from 0.37 to 0.59, suggesting that POS is a highly useful feature. In the next set of experiments, we evaluated the impact of adding additional features individually to the combination of the high MI words and POS features.

Table 4-2: System performance for different sets of features

Features	Precision	Recall	F-Measure
1000 words with high MI	0.30	0.74	0.37
1000 words with high MI +POS	**0.48**	**0.83**	**0.59**

1000 words with high MI +POS +Verb features	0.45	0.80	0.56
1000 words with high MI +POS +Numbers features	0.38	0.75	0.50
1000 words with high MI +POS +Path Between drugs	0.42	0.71	0.51
1000 words with high MI +POS +BiGrams	0.34	0.78	0.46
1000 words with high MI +POS +Negation	0.45	0.83	0.58
1000 words with high MI +POS +Cosine Similarity	0.48	0.83	0.59
1000 words with high MI +POS +Clause Boundary	0.45	0.80	0.56

Verb features, shown in the third row, represents the number of verbs in the sentence, and the number of verbs from each class-specific verb list (discussed in Chapter 3). As can be seen in the table, the F-Measure dropped by 0.03% after adding these features to the system.

The fourth row of Table 4-2 shows the impact of *number features* on the system. The *number features* contain three features, the number of words in the sentence, the number of words between main drugs, and the number of drug names in the sentence. The result indicates that *number features* decreased the system performance.

One of the features we explored was *path between main drugs in the parse tree*, shown in the sixth row of the table. We traversed the parse tree to extract the path between the main drugs, while reducing the length of the path by collapsing identical adjacent non-terminal labels. Using this feature decreased the performance by 0.08.

The next experiment explored the effect of using Bi-grams as feature in the classifiers. For each word Bi-grams that appeared more than once in the corpus, we assigned a feature to show the presence or absence of the bi-gram in the sentence. The sixth row of Table 4-2 presents the results, highlighting that adding Bi-grams to the system decreased the F-Measure.

One of our explored features was *negation*. We added three features to the classifiers that indicated presence of negation in different parts of sentence: before the first drug, between two main drugs, and after the second drug. The results are presented in the seventh row of Table 4-2. This feature decreased the F-Measure from 0.59 to 0.58.

We also explored the *Cosine similarity feature* that presented the cosine similarity between a sentence and the centroid vector of each class. The eighth row of Table 4-2 shows the performance of the system after adding this feature. Unlike the other features, this feature didn't decrease the performance of the system. But it did not increase the F-Measure either, so we decided not to use this feature in the final system.

The last explored feature is *Clause Boundary*, which indicates whether main drugs are in the same clause or not. The performance of the system after adding this feature is presented in the last row of table 4-2. It didn't increase the F-Measure.

After exploring different sets of features, we found that the best result is obtained with using 1000 words with high Mutual Information together with part of speech tags. In the following experiments, therefore, we used this set of features to explore additional classifiers.

4-4-2 Classifier Selection

Given the best set of features, we explored 6 different classifiers for each stage: Naïve Bayes, Multinomial Naïve bayes, J48, Jrip, Random forest, and SVM. To find the best fit for the first stage classifier, we used SVM as the classifier in the second stage, and evaluated all the classifiers in the first stage. The results in Table 4-3 illustrates that SVM is the best fit for the first stage based on its F-Measure. It is worth noting, however, that with Random Forest, the precision was 30% more than with SVM. Therefore, for a system aiming at high precision, Random Forest would be the best choice for this stage.

We note that to handle the unbalanced class problem, we explored different approaches and algorithms, including SMOTE [68] and other resampling algorithms, but they were not effective. Instead, we found that assigning weights to the classes in SVM was more effective. Therefore, we learned the best weights for each class, based on cross-validation over the training set. In the first classifier, we assigned weight 1 to the *None* class and 6.5 to the *positive* class. In the second classifier, the best weights were 800, 600, 3200, and 500, respectively, for *advise*, *effect*, *int*, and *mechanism*.

Table 4-3: System performance for different classifiers in the first stage

Classifier in the first stage	**Precision**	**Recall**	**F-Measure**
Naïve Bayes	0.20	0.72	0.31
Multinomial Naïve bayes	0.32	0.60	0.41
J48	0.69	0.50	0.57
Jrip	0.75	0.36	0.48
Random forest	0.78	0.45	0.56
SVM	**0.48**	**0.83**	**0.59**

We ran the same experiment for finding the best classifier for the second stage. In these experiments, SVM is used as the classifier in the first stage and the best features are utilized to train the classifiers. Table 4-4 illustrates the results of these experiments. As the results show, we obtained the best F-measure by using SVM in the second stage.

Table 4-4: System performance for different classifiers in the second stage

Classifier in the second stage	**Precision**	**Recall**	**F-Measure**
Naïve Bayes	0.30	0.59	0.39
Multinomial Naïve bayes	0.41	0.68	0.48
J48	0.36	0.75	0.48
Jrip	0.36	0.72	0.48
Random forest	0.47	0.69	0.51
SVM	**0.48**	**0.83**	**0.59**

4-4-3 Impact of Post-Processing

After finding the best set of features and classifiers, we explored the impact of the post-processing rules. We developed two sets of rules -- one applied after the first stage and the second on the final results. Table 4-5 presents the results, showing that using post-processing rules increased the F-measure. A larger increase is observed with the first stage rules, which may be attributed to the larger number of rules that are likely to have covered more examples.

Table 4-5: Impact of post-processing rules.

Post-processing	**Precision**	**Recall**	**F-Measure**
None	0.48	0.83	0.59
After the first stage	0.53	0.82	0.63
On the final results	0.53	0.80	0.62

After the first stage +On the final results	0.55	0.80	0.64

4-4-4 Class-wise Performance

Apart from the overall results given above, it is also useful to examine the class-wise performance of the system. This is given in Table 4-6. What is interesting to observe is that the "Int" class shows the highest F-measure, even though this class had the fewest instances in the training data. The hardest class to identify was the *Mechanism* class, which is probably due to its confusability with the *Effect* class, which also shows a lower F-measure compared to *Int* and *Advise*.

Table 4-6: Class-wise Precision, Recall, and F-Measure

Type	Precision	Recall	F-Measure
Advise	0.53	0.80	0.64
Effect	0.46	0.77	0.57
Mechanism	0.32	0.74	0.45
Int	0.88	0.88	0.88

4-4-5 Stage 1 Performance

All results above present the performance of the system after the second stage classification. But it is also useful to assess the performance of the first stage classification alone, as it provides insight into the task of DDI identification, i.e., classification as positive vs. negative. For this task taken alone, the precision, recall, and F-Measure after applying SVM and post-processing were 0.48, 0.86, and 0.61 respectively. The low precision relative to recall seen here is not surprising since the positive class was assigned a higher weight. However, what is interesting is that although

the impact of the first stage post-processing on the overall result was significant (Table 4-5), it's absolute impact on the precision in the first stage is obviously not strong enough. As we discuss in the error analysis later in this chapter, there is much room for improving the precision in the first stage with better post-processing.

4-4-6 Comparison with the competition system

Since our submitted system to the DDIExtraction-2013 challenge used a different combination of features, we compared its performance with our current system. As shown in Table 4-7, the F-measure in our new system is 12% higher than the competition system. This comparison shows the effect of careful feature selection carried out for our current experiments, which we were not able to carry out for the competition system due to time constraints. In the competition system, we used a large number of features, given lack of knowledge about which particular features might be most effective.

We note that comparison of our augmented system with the other systems from the competition is not possible because of the unavailability of the challenge's evaluation test set until the time of writing of this thesis.

Table 4-7: Comparison of best system with DDIExtraction-2013 competition system

Features	Precision	Recall	F-Measure
The best system (1000 words with high MI +POS)	0.55	0.80	0.64
The competition system (All words + BiGrams + POS + Verb +Number features +Cosine Similarity)	0.52	0.73	0.52

4-5 Error Analysis

As the results above show, the F-Measure of our system is not very high, but this was true of all the systems in the competition. To some extent, the poor performance in general can be attributed to the corpus itself, specifically to the unbalanced distribution of the types. This is further compounded by the unbalanced proportion of sentences from the two sources from where the corpus was drawn: only 6% of the sentences in the corpus come from Medline.

However, for our system alone, one of the major reasons for the low F-measure was the assignment of a higher weight to the positive class, which resulted in a high false positive rate. Our error analysis shows that 87% of the errors were stage 1 errors, and that more sophisticated features for learning, or rules for post-processing, should be developed for further improvement, as discussed next.

Some errors occurred because the post-processing rules identify grouped mentions of drugs based only on lexical and punctuation identifiers. Most of these errors could have been averted if the identification of grouped mentions also utilized syntactic information. For example, in the following sentence,

- *"Drugs that Lower Seizure Threshold: Concurrent administration of WELLBUTRIN and agents (e.g., antipsychotics, other antidepressants,* ***theophylline****, systemic* ***steroids****, etc.) that lower seizure threshold should be undertaken only with extreme caution."*

Our current rules couldn't detect that "*theophylline*" and "*steroids*" were mentioned in the same group because group identification in the current rules requires adjacency of the

drug names. Using syntactic constituency instead and recognizing intervening words as modifiers could have identified grouped mentions more systematically.

Our analysis also revealed other features that are important to exploit, such as lexical semantics, scope of negation and hypothetical markers, scope of salient keywords, and noun phrase referential status.

The following example shows that it may be possible to exploit the syntactic and semantic scope of the interaction keyword, *concurrent*, to the two drug names following it, but not the one preceding it. This example, as an instance for the two drugs, "*etretinate*" and "*acitretin*", was annotated as *None* (no interaction), whereas our system labeled it as *Effect*.

- *"Ethanol:Clinical evidence has shown that **etretinate** can be formed with concurrent ingestion of **acitretin** and ethanol."*

The next example illustrates the role of hypothetical marking and its scope. The interaction in question is the two drugs, "*Argatroban"* and "*heparin",* highlighted in the hypothetical if-clause, which the system ought to accordingly treat as a hypothetical interaction and label as *None*. Instead, our system labeled this instance as *Effect.*

- *"However, if **Argatroban** is to be initiated after cessation of **heparin** therapy, allow sufficient time for heparins effect on the aPTT to decrease prior to initiation of Argatroban therapy."*

Of course, some errors were related to the annotation. For example, PAH in the following sentence does not refer to a drug at all. Note, though, that even if PAH *were* considered to be a drug, the class label assigned to it seems to be incorrect, both by the annotation

(labeled as *Effect*) and the system (labeled as *Advise*). Unfortunately, information about the level of noise to be expected of the corpus isn't available, so it is difficult to quantify the contribution of such errors and use it to set an upper bound on performance.

- *"Renal clearance measurements of **PAH** cannot be made with any significant accuracy in patients receiving **sulfonamides**, **procaine**, or **thiazolesulfone**."*

4-6 Conclusions and Future Work

In this thesis, we have presented a system to extract and classify DDI mentions from biomedical text. As our corpus contains a highly unbalanced class distribution, we applied a two-stage classifier to handle this problem. In the first stage, a binary classifier classified drug pairs into non-interaction and interaction classes. Then, drug pairs that were detected as interacting in the first stage are classified via a multi-class classifier into *Advise*, *Effect*, *Mechanism*, and *Int* classes. We explored various features in a selective way to find the best set of features for the classifier. We also experimented with six different classifiers in each stage to choose the best classifier for each. We further applied post-processing rules after each stage to improve the results. We have argued that handling the unbalanced class distribution is one of the advantages of our approach. In addition, our approach allows for using different sets of features and classifiers in the different stages. We learned that for this specific application, using SVM in both stages obtains the best F-Measure, although Random Forest in the first stage obtained 30% more precision than SVM. In this application, we used the same features to train both classifiers. The best feature set included 1000 top informative words and part of speech tags between two main drugs. The F-Measure of our system is 0.64, which is 0.12 higher

than our submitted system to the DDIExtraction 2013. This result shows the effectiveness of feature selection, because in the submitted system, we used a longer list of features without applying any feature selection approach. Unfortunately, we couldn't compare our results with the other systems in the competition because the competition test set wasn't available at the time of writing of this thesis.

As future work for this thesis, we plan to:

- Train our system on the competition training set and test on its test set
- Use two different lists of features for each classifier: As the classifiers are different, binary and multi-class classifier, we will investigate different sets of features for each of them. We plan to exploit linguistic features in a more sophisticated way, including scope of negation, hypothetical marking, and salient keywords.
- Add a 5th class ("*None*") to the second classifier to detect some false positive instances generated by the first classifier. As the second classifier classifies instances into *Advice*, *Effect, Mechanism, and Int,* it is not able to detect false positive instances generated by the first classifier. So, we will add "*None*" class to this classifier and convert it into a five-class classifier.
- Use syntactic information in addition to lexical/punctuation signals for post-processing.
- Explore kernel based SVM: We only used linear SVM in this thesis but we will explore the effect of kernel based SVM.
- Explore ensemble classification in both stages.

- Train two separate systems for Medline and DrugBank sentences: Since DrugBank and Medline exhibit different structures in their sentences and documents, we expect that having separate classifiers for each will lead to better performance.

References

[1] “To Err is Human: Building A Safer Health System - Institute of Medicine.” [Online]. Available: http://www.iom.edu/Reports/1999/to-err-is-human-building-a-safer-health-system.aspx. [Accessed: 15-Aug-2013].

[2] D. P. Phillips, N. Christenfeld, and L. M. Glynn, “Increase in US medication-error deaths between 1983 and 1993,” *Lancet*, vol. 351, no. 9103, pp. 643–644, Feb. 1998.

[3] *Second National Report on Patient Safety: Improving Medication Safety*. Australian Council for Safety and Quality in Health Care, 2002.

[4] J. U. Rosholm, L. Bjerrum, J. Hallas, J. Worm, and L. F. Gram, “Polypharmacy and the risk of drug-drug interactions among Danish elderly. A prescription database study,” *Dan. Med. Bull.*, vol. 45, no. 2, pp. 210–213, Apr. 1998.

[5] M. Pirmohamed, S. James, S. Meakin, C. Green, A. K. Scott, T. J. Walley, K. Farrar, B. K. Park, and A. M. Breckenridge, “Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients,” *BMJ*, vol. 329, no. 7456, pp. 15–19, Jul. 2004.

[6] W. B. Runciman, E. E. Roughead, S. J. Semple, and R. J. Adams, “Adverse drug events and medication errors in Australia,” *Int. J. Qual. Health Care*, vol. 15, no. suppl 1, pp. i49–i59, Dec. 2003.

[7] J. R. Nebeker, P. Barach, and M. H. Samore, “Clarifying adverse drug events: a clinician’s guide to terminology, documentation, and reporting,” *Ann. Intern. Med.*, vol. 140, no. 10, pp. 795–801, May 2004.

[8] L. Magro, U. Moretti, and R. Leone, “Epidemiology and characteristics of adverse drug reactions caused by drug-drug interactions,” *Expert Opin. Drug Saf.*, vol. 11, no. 1, pp. 83–94, Jan. 2012.

[9] J. H. Gurwitz, 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, “Incidence and preventability of adverse drug events among older persons in the ambulatory setting,” *JAMA J. Am. Med. Assoc.*, vol. 289, no. 9, pp. 1107–1116, Mar. 2003.

[10] J. H. Gurwitz, 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, “The incidence of adverse drug events in two large academic long-term care facilities,” *Am. J. Med.*, vol. 118, no. 3, pp. 251–258, Mar. 2005.

[11] L. E. Hines and J. E. Murphy, “Potentially harmful drug-drug interactions in the elderly: a review,” *Am. J. Geriatr. Pharmacother.*, vol. 9, no. 6, pp. 364–377, Dec. 2011.

[12] P. Aspden, J. Wolcott, J. L. Bootman, L. R. Cronenwett, "*Preventing Medication Errorr: Quality Chasm Series,* 2007.

[13] Y.-F. Chen, A. J. Avery, K. E. Neil, C. Johnson, M. E. Dewey, and I. H. Stockley, "Incidence and possible causes of prescribing potentially hazardous/contraindicated drug combinations in general practice," *Drug Saf. Int. J. Med. Toxicol. Drug Exp.*, vol. 28, no. 1, pp. 67–80, 2005.

[14] L. E. Hines, D. C. Malone, and J. E. Murphy, "Recommendations for generating, evaluating, and implementing drug-drug interaction evidence," *Pharmacotherapy*, vol. 32, no. 4, pp. 304–313, Apr. 2012.

[15] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A. C. Guo, and D. S. Wishart, "DrugBank 3.0: a comprehensive resource for 'omics' research on drugs," *Nucleic Acids Res.*, vol. 39, no. Database issue, pp. D1035–1041, Jan. 2011.

[16] J. S. Carter, S. H. Brown, B. A. Bauer, P. L. Elkin, M. S. Erlbaum, D. A. Froehling, M. J. Lincoln, S. T. Rosenbloom, D. L. Wahner-Roedler, and M. S. Tuttle, "Categorical Information in Pharmaceutical Terminologies," *AMIA. Annu. Symp. Proc.*, vol. 2006, pp. 116–120, 2006.

[17] C. Chute, J. Carter, M. Tuttle, M. Haber, and S. Brown, "Integrating Pharmacokinetics Knowledge into a Drug Ontology As an Extension to Support Pharmacogenomics," AMIA. Annu. Symp. Proc., vol. 2003, pp. 170–174, 2003.

[18] M. Rastegar-Mojarad, B. Harrington, and S. M. Belknap, "Automatic detection of drug interaction mismatches in package inserts," in 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 373–377.

[19] A. I. Vitry, "Comparative assessment of four drug interaction compendia," *Br. J. Clin. Pharmacol.*, vol. 63, no. 6, pp. 709–714, Jun. 2007.

[20] V. S. Shah, R. J. Weber, and M. C. Nahata, "Contradictions in contraindications for drug-drug interactions," *Ann. Pharmacother.*, vol. 45, no. 3, pp. 409–411, Mar. 2011.

[21] A. Li, S. Zhao, and T. Z. Jodlowski, "How Up-to-Date Is Your Drug-Drug Interaction Database?," *Ann. Pharmacother.*, vol. 45, no. 12, pp. 1591–1592, Dec. 2011.

[22] P. M. I Segura-Bedmar, "The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts," pp. 1–9, 2011.

[23] A. R. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," *Proc. AMIA Annu. Symp. AMIA Symp.*, pp. 17–21, 2001.

[24] S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski, "Comparative analysis of five protein-protein interaction corpora," *BMC Bioinformatics*, vol. 9, no. Suppl 3, p. S6, Apr. 2008.

[25] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts," in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[26] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications," *J. Am. Med. Inform. Assoc. JAMIA*, vol. 17, no. 5, pp. 507–513, Oct. 2010.

[27] I. Segura-Bedmar, P. Martínez, and M. Segura-Bedmar, "Drug name recognition and classification in biomedical texts. A case study outlining approaches underpinning automated systems," *Drug Discov. Today*, vol. 13, no. 17–18, pp. 816–823, Sep. 2008.

[28] K. M. Hettne, R. H. Stierum, M. J. Schuemie, P. J. M. Hendriksen, B. J. A. Schijvenaars, E. M. van Mulligen, J. Kleinjans, and J. A. Kors, "A dictionary to identify small molecules and drugs in free text," *Bioinformatics*, vol. 25, no. 22, pp. 2983–2991, Nov. 2009.

[29] T. Rocktäschel, M. Weidlich, and U. Leser, "ChemSpot: a hybrid system for chemical named entity recognition," *Bioinforma. Oxf. Engl.*, vol. 28, no. 12, pp. 1633–1640, Jun. 2012.

[30] I. Segura-Bedmar, P. Martínez, and C. de Pablo-Sánchez, "A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents," *BMC Bioinformatics*, vol. 12, 2011.

[31] I. Segura-Bedmar, P. Martínez, and C. de Pablo-Sánchez, "Using a shallow linguistic kernel for drug-drug interaction extraction," *J Biomed Inform*, pp. 789–804, Oct. 2011.

[32] J. Mata, R. Santano, D. Blanco, M. Lucero, and M. J. Maña, "A Machine Learning Approach to Extract Drug-Drug Interactions in an Unbalanced Dataset," in *The 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, 2011, vol. 761, pp. 59–65.

[33] M. F. M. Chowdhury and A. Lavelli, "Drug-drug interaction extraction using composite kernels," in *The 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, 2011, vol. 761, pp. 27–33.

[34] P. Thomas, M. Neves, I. Solt, D. Tikk, and U. Leser, "Relation Extraction for Drug-Drug Interactions using Ensemble Learning," in *The 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, 2011, vol. 761, pp. 11–18.

[35] J. Bjorne, F. Ginter, J. Heimonen, A. Airola, T. Pahikkala, and T. Salakoski, *TEES: Event Extraction Software*. 2011.

[36] A.-L. Minard, L. Makour, A.-L. Ligozat, and B. Grau, “Feature Selection for Drug-Drug Interaction Detection Using Machine-Learning Based Approaches,” in *The 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, 2011, vol. 761, pp. 43–50.

[37] S. Karnik, A. Subhadarshini, Z. Wang, L. M. Rocha, and L. Li, “Extraction Of Drug-Drug Interactions Using All Paths Graph Kernel,” in *The 1st Challenge Task on Drug-Drug Interaction Extraction*, Huelva, Spain, 2011, vol. 761.

[38] M. F. M. Chowdhury and A. Lavelli, “FBK-irst : A Multi-Phase Kernel Based Approach for Drug-Drug Interaction Detection and Classification that Exploits Linguistic Information,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[39] T. Joachims, “Making large-scale support vector machine learning practical,” in *Advances in Kernel Methods: Support Vector Machines*, C. Sch"olkopf, Ed. MIT Press, Cambridge, MA, 1998.

[40] E. Charniak and M. Johnson, “Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2005, pp. 173–180.

[41] D. McClosky, “Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing,” Department of Computer Science, Brown University, 2010.

[42] D. Klein and C. D. Manning, “Accurate unlexicalized parsing,” in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, Stroudsburg, PA, USA, 2003, pp. 423–430.

[43] P. Thomas, M. Neves, T. Rocktaschel, and U. Leser, “WBI-DDI: Drug-Drug Interaction Extraction using Majority Voting,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[44] D. Tikk, P. Thomas, P. Palaga, J. Hakenberg, and U. Leser, “A Comprehensive Benchmark of Kernel Methods to Extract Protein–Protein Interactions from Literature,” *PLoS Comput Biol*, vol. 6, no. 7, p. e1000837, Jul. 2010.

[45] A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski, “All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning,” *BMC Bioinformatics*, vol. 9, no. Suppl 11, p. S2, Nov. 2008.

[46] C. Giuliano, A. Lavelli, and L. Romano, “Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature,” in *In Proc. EACL 2006*, 2006.

[47] S. V. N. Vishwanathan and A. Smola, *Fast Kernels for String and Tree Matching*. 2004.

[48] M. Collins and N. Duffy, “Convolution Kernels for Natural Language,” in *Advances in Neural Information Processing Systems 14*, 2001, pp. 625–632.

[49] T. Kuboyama, K. Hirata, H. Kashima, K. F.Aoki-Kinoshita, and H. Yasuda, “A Spectrum Tree Kernel,” *Trans. Jpn. Soc. Artif. Intell.*, vol. 22, pp. 140–147, 2007.

[50] R. S. Sampo Pyysalo, “Why Biomedical Relation Extraction Results are Incomparable and What to do about it,” in *SMBM’08*, 2008, pp. 149–152.

[51] J. Bjorne, S. Kaewphan, and T. Salakoski, “UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[52] B. Bokharaeian and A. Diaz, “NIL-UCM: Extracting Drug-Drug interactions from text through combination of sequence and tree kernels,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[53] C. D. Paice, “Another stemmer,” *SIGIR Forum*, vol. 24, no. 3, pp. 56–61, Nov. 1990.

[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *SIGKDD Explor Newsl*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[55] M. Rastegar-Mojarad, R. D. Boyce, and R. Prasad, “UWM-TRIADS : Classifying Drug-Drug Interactions with Two-Stage SVM and Post-Processing,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[56] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, 2011.

[57] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, Stroudsburg, PA, USA, 2003, pp. 173–180.

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

[59] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[60] T. Bobic, J. Fluck, and M. Hofmann-Apitius, “SCAI: Extracting drug-drug interactions using a rich feature vector,” in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[61] N. D. Hailu, L. E. Hunter, and B. Cohen, "UColorado-SOM: Extraction of Drug-Drug Interactions from BioMedical Text using Knowledge-rich and Knowledge-poor Features," in *7th International Workshop on Semantic Evaluation*, Atlanta, 2013.

[62] L. Hunter, Z. Lu, J. Firby, W. A. Baumgartner, H. L. Johnson, P. V. Ogren, and K. B. Cohen, "OpenDMAP: An open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression," *BMC Bioinformatics*, vol. 9, no. 1, p. 78, Jan. 2008.

[63] L. Tari, S. Anwar, S. Liang, J. Cai, and C. Baral, "Discovering drug-drug interactions: a text-mining and reasoning approach based on properties of drug metabolism," *Bioinformatics (Oxford, England)*, pp. i547–53, 2010.

[64] R. Boyce, G. Gardner, and H. Harkema, "Using Natural Language Processing to Extract Drug-Drug Interaction Information from Package Inserts," in *The 2012 Workshop on Biomedical Natural Language Processing*, Montreal, Canada, 2012, pp. 206–213.

[65] M. F. Porter, "An algorithm for suffix stripping," *Program Electron. Libr. Inf. Syst.*, vol. 14, no. 3, pp. 130–137, Dec. 1980.

[66] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik, "The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes," *BMC Bioinformatics*, vol. 9, no. Suppl 11, p. S9, Nov. 2008.

[67] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[68] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.