Comparison of Wrapper Based Feature Selection and Classifier Selection Methods for Drug Named Entity Recognition

Saman Sharifian Razavi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçioğlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Ekrem Varoğlu
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Ekrem Varoğlu
2. Asst. Prof. Dr. Nazife Dimililer
3. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

Bioinformatics is a new yet quickly evolving interdisciplinary field that combines different other branches of science like biology and computer science. This field of science mainly relates to the process of extracting, categorizing and finally analyzing relevant biological data from large and not organized sources of information available. In this thesis, two machine-learning approaches, namely SVM and CRF have been performed for the recognition and classification of drugs and chemicals. These tasks are named as DrugNER and DrugNEC and have gained significant attention from the biomedical text mining community in recent years. Train and test datasets used in this work are derived from The DDI Corpus [1]. Three groups of features, morphological, lexical and orthographic are used. Wrapper based feature selection methods are used to find an optimal feature ensemble. In addition, wrapped based classifier selection algorithms are used in order to find an optimal set of classifiers from a large pool of CRF and SVM based classifiers. Results of both approaches have been compared. Finally a new majority voting algorithm, referred to as ranked-weighted majority voting is proposed and used during the combination of classifiers.

Keywords: Biomedical Text Mining, Drug Name Entity Recognition, Feature Selection, Ranked-Weighted Majority Voting, Classifier Selection, Machine Learning, Support Vector Machines, Conditional Random Fields.

ÖZ

Biyoi-bilişim yeni ve ayni zamanda hızla gelişen,biyoloji ve bilgisayar bilimleri alanlarını birleştiren multidisipliner bir alandır. Çoğunlukla iyi organize edilmemiş, büyük very kaynaklardan biyolojik bilginin çıkarılması, sınıflandırılması ve analiz edilmesi ile ilgilenen bir alandır. Bu tezde, otomatik öğrenmeye dayalı sınıflandırcılar olan Vektör Destek Makineleri (VDM) ve Koşullu Rastegele Alanlar (KRA) sınıflandırıcıları kullanılarak kimyasal ve ilaç isimlerinin metinden çıkarılarak sınıflandırılması yapılmıştır. İlaç İsimlendirilmiş Nesne Tanıma ve Sınıflandırılması diye tanımlanan bu işlemler biyo-medikal veri madenciliği alanında son yıllarda araştırmacıların büyük ilgisini çekmiştir. Bu çalışmada kullanılan eğitim kümesi ve test kümesi DDI Bütünce'sinden [1] üretilmiştir. Çeşitli yapılarda morfolojik, sözlüksel, ve ortografik öznitelikler kullanılmıştır. En iyi öznitelik alt kümesini elde edebilmek için sargı yöntemine dayalı algoritmalar olarak İleri Seçim, ve algoritmaları kullanılmıştır. Buna ilave olarak en iyi sınıflandırıcı alt kümesini bulmak için de ayni algoritmalar denenmiştir. Her iki yöntemin sonuçları çalışmada karşılaştırılmıştır. Son olarak, sınıflandırıcıların birleştirilmesinde ağırlık katmanlı çoğunluk oylama diye adlandırılmış yeni bir çoğunluk oylama yöntemi önerilmiştir.

Anahtar kelimeler: Biyo-medikal Metin Madenciliği, İlaç İsimlendirilmiş Nesne Tanıma, Öznitelik Seçme, Ağırlık Katmanlı Çoğunluk Oylama, Sınıflandırıcı Seçme, Otomatik Öğrenme, Vektör Destek Makineleri, Koşullu Rastegele Alanlar.

Dedicated to my family

ACKNOWLEDGMENT

I would like to thank Assoc. Prof. Dr. Ekrem Varoğlu for his continuous support and guidance in the preparation of this study. Without his invaluable supervision, all my efforts could have been short-sighted.

I also want to thank my family who motivated me for continuing my studies and supported me through my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CRF	Conditional Random Fields
SVM	Support Vector Machine
INN	International Nonproprietary Names
FS	Forward Selection
BS	Backward Selection
Bio-NER	Biomedical Named Entity Recognition
CV	Cross-Validation
POS	Part of Speech
DDI	Drug-Drug Interaction
PPI	Protein-Protein Interaction
PI	Package Insert
ATC	Anatomical Therapeutic Chemical
NP	Noun Phrase
VP	Verb phrase
PP	Preposition phrase
TM	Text Mining

Chapter 1

INTRODUCTION

1.1 Background

As Miller, T. W. suggests in their work [2], Text Mining (TM) can be considered as an automatic or semi-automatic processing of text. We can name many useful objectives for text mining tasks including being able to analyze news data that is being added daily in online archives of news agencies in an automatic or semi-automatic manner or predicting possible overlap effects of two drugs on a person who takes them by mining a corpus made from biomedical texts. As other examples for applications of text mining, we can name spam filtering, fighting cyberbullying or cybercrime in online chat records, automatic labeling of documents in electronic libraries, monitoring public opinions in online domain and so on. The most important concept of text mining is the ability to interpret structured data based on knowledge and patterns that we have received from unstructured text - known as text corpus - and store it in a database. Typical text mining tasks include text clustering, categorization, relationship extraction, document summarization, automatic content extraction, and exploratory data analysis.

Both supervised and unsupervised learning methods are employed in text mining tasks. Supervised approaches typically make use of annotated datasets, usually known as a corpus, whereas unsupervised methods do not need such labeled data. Text classification and Named Entity Recognition (NER) are typical tasks that make use of supervised approaches. Text clustering on the other hand usually makes use of unsupervised methods.

Text classification or categorization is the process of assigning a predefined class (category) for each document. Text clustering on the other hand is the task of grouping similar documents together.

Natural Language Processing (NLP) tools have been used recently very extensively in many TM tasks. NLP techniques have proven to be very useful in extracting meaningful representations from free text. Many earlier NLP systems relied on hand written rules and grammars. However, machine learning (ML) systems are now widely accepted and used in many NLP related tasks.

The most basic problem in many automatic text extraction tasks is Named Entity Recognition (NER) [3]. The objective of NER is to detect named entities in text from different domains such as news or biomedicine. In the widely used Newswire domain, this accounts to detection of names of persons, locations etc. In the biomedical domain, the focus is on naming genes, proteins, drugs etc. Achieving a high level performance in any NER system is a vital step in many further information extraction tasks such as identifying the relationships between entities, genes, proteins or drugs [4] [5]. Named Entity Classification (NEC) is the next step following NER where a specific class is assigned to each recognized named entity. Several methods, such as dictionary based [6], rule based [7] and machine learning based [8] methods have been used recently for NER and NEC tasks.

1.2 Thesis Contribution

In this thesis, the focus is on DrugNER and DrugNEC which mainly involve detection and classification of drug names in biomedical literature which serves an important role in Biomedical Natural Language Processing (BioNLP) tasks including extraction of pharmaco-genomic, pharmaco-dynamic and pharmaco-kinetic parameters [9]. Two well-known Machine Learning (ML) methods, namely Support Vector Machines (SVM) and Conditional Random Fields (CRFs) are used for the NER and NEC tasks [10] [11] .The data used for training and testing of the classifiers is the corpus from the SemEval 2013 Drug name recognition task (DDIExtraction 2013) [1]. Several orthographic, syntactic and lexical features have been extracted from this dataset and used separately and in combination, in order to test the effects of using these features solely as well as in combination on the DugNER and DrugNEC tasks. Furthermore, wrapper based selection algorithms are employed for both feature and classifier selection. In particular, the Forward Search (FS) and Backward Search (BS) algorithms are utilized and the effects of both feature subset selection and classifier subset selection on the final classification performance is analyzed and compared to one another.

1.3 Thesis Outline

This thesis is organized as follows: Chapter 2 provides an overview of related works that are carried out in biomedical NER with emphasis on Drug Name Recognition and Drug Name Classification (DrugNER and DrugNEC). Chapter 3 presents all stages of the machine learning based NER system used. Chapter 4 presents the results obtained and compares different methods used with respect to the results obtained. Chapter 5 summarizes the work done and makes overall conclusions as well as making suggestions for future work.

Chapter 2

LITERATURE REVIEW

2.1 Biomedical Text Mining

In recent years, there has been a big increase in the amount of data available in biomedical domain. This increase has been taking place especially in the field of pharmacology, genomics and proteomics. For instance we can name Medline[12] database that contains over 21 million references to journal articles in life sciences with a concentration on biomedicine. Another example is the Drugbank [13] database that contains 7740 drug entries where each entry contains more than 200 data fields with half of the information being devoted to drug/chemical data and the other half devoted to drug target or protein data [13]. We should also consider the fact that these databases with biomedical content are being updated and become larger on a regular basis. Considering this huge amount of data, in order to obtain proper information and knowledge extracted from these databases, sophisticated text mining methods must be applied since most of the data is kept within journal articles in the form of free text. This task is carried out on literature which involves contents on biology, chemistry, medicine, pharmacology and genetics and is referred to as "biomedical text mining". Like all other text mining branches, it consists of sub tasks which include information extraction that leads to searching and selecting relevant information from biomedical databases using methods from Natural Language Processing (NLP) and/or Artificial Intelligence (AI). Some of the main information extraction tasks in this domain involve:

- Naming of drugs and chemical compounds or genes or proteins. [14]
- Classification of drugs or proteins or chemical compounds. [15]
- Discovering possible interactions that might occur between drugs and proteins in human body. [16]
- Identifying possible relations that might exist between drugs and proteins and some genetic mutations or diseases. [14] [15] [16]
- Predicting some new effects of these drugs etc. [17]

In order to develop methods and tools for each of the tasks mentioned above as well as encourage those involved in these studies to make new innovations or improve their existing systems, biomedical text mining tasks and workshops are carried out for the last twenty years. Some main events in this field are as follows:

The first challenge that can be mentioned in the biomedical domain is Knowledge Discovery and Data Mining (KDD) challenge cup task 1 [18], which involved extracting information from biomedical articles. From years 2003 to 2007, Text Retrieval Conference (TREC) which is a major center of work and evaluation in information retrieval community, introduced TREC Genomics Track [19] [20] [21] which was mainly focused on ad hoc retrieval, text summarization, text categorization and question–answering in biomedical domain. In 2004, Critical Assessment of Information Extraction systems in Biology (BioCreAtIvE) [22] and Joint Workshop in Natural Language Processing in Biomedicine and its Applications (JNLPBA) [23] were held. Tasks discussed in BioCreative I (2004) [24] [25] were Gene Mention Identification, Gene Normalizations and Functional Annotations. In BioCreative II (2006) [26] tasks were Gene Mention Tagging, Human Gene Normalizations, protein-protein Interactions. In BioCreative III (2010) [27] tasks were Gene Normalization, Interactive Demostration and a task for Gene Indexing and Retrieval and Protein-

Protein Interactions. BioCreative IV (2013) [28] involved Chemical compound and drug name recognition tasks [29]. The main task attempted in JNLPBA was Bio-medical NER [23]. In 2005, Learning Language in Logic (LLL) challenge [30] was held and task of extracting relations from bio-medical texts that mainly were about protein-gene interactions, evaluated by the organizers. Another shared task series that has been introduced to biomedical text mining community since 2002 is ACL-associated BioLINK and BioNLP [31] [32]. The last three events from this challenge are namely BioNLP-ST 2009, BioNLP-ST 2011 and BioNLP-ST 2013 that has gained a high recognition among those participating in biomedical text mining [33]. Critical Assessment of protein Structure Prediction (CASP), is another center that its aim is to help improving methods of identifying protein structure from sequence [34]. This center has been active since 1994 up to 2015 [35]. The Pacific Symposium on Biocomputing (PSB) is a major conference, which lists among its topics development of tools and computational methods with focus on biological literature, especially in the area of molecular biology [36]. More specifically, computational methods and infrastructure for integrative analysis of cancer, high-throughput "omics" data to enable precision oncology, new methods for understanding the etiology of complex traits and disease, genotypes, molecular phenotypes, cancer pathways, automatic extraction, representation and reasoning in big data are the general fields of interest in this conference [37]. PSB has been active since 1996 until 2015. Another challenge in this domain is Conference and Labs of the Evaluation forum – Entity Recognition (CLEF-ER 2013) [38] that its focus is on some different tasks like entity mention annotation, entity normalization and multilingual analysis of a corpus [39].

Two important and well-known conferences with focus on biology are Intelligent Systems for Molecular Biology Conference (ISMB) and Conference on Semantics in

Healthcare & Life Sciences (CSHALS). ISMB has been running since 1993. Some of the topics of latest ISMB included in ISMB 2014 are population genomics, protein interactions and molecular networks, protein structure and function, RNA bioinformatics and sequence analysis [40]. CSHALS has been organized by international society for computational biology (ISCB) since 2008 up to now with focus on pharmaceutical applications of semantic technologies. It focuses on subjects like clinical information management, integrated healthcare and semantics in electronic health records, translational medicine/safety and discovery information integration. Collaborative annotation of a large biomedical corpus (CALBC) is a European workshop that is devoted to creation of a broadly scoped and diversely annotated corpus [41]. The project started in January 2009 and finished in June 2011. During this time partners of this project organized first challenge in autumn of 2009 and the second challenge in autumn of 2010. The challenge consists of two tasks, the first one is about named entity recognition in which participants were supposed to provide annotations of the boundaries and semantic groups of the found entities and the second task was about concept identification in which participants were supposed to provide annotations of the boundaries and concept identifiers of the found entities [42]. Informatics for integrating biology and the bedside (i2b2) is a platform for biomedical computing with focus on healthcare systems. i2b2 /UTHealth Shared-Task and Workshop, 2008, 2009, 2011, 2012 and 2014 are examples of previous challenges based on this platform. Two tracks are defined for this challenge, The first one is de-identification that is about removing protected health information (PHI) from medical records in order to make them accessible for public and the second task is identifying risk factors for heart disease over time. The final goal in this task is to recognize information that is medically relevant to identifying heart disease risk, and tracking their progression in patient's records [43].

Drug-drug interaction extraction 2011 (DDIExtraction2011) and drug-drug interaction extraction 2013 (DDIExtraction2013) are two recent workshops that has been organized recently in Carlos III university mostly based on BioCreAtIvE challenge evaluation guidelines. This task as it can be seen in its title is about recognizing and classifying possible interactions between drugs in the given corpus. Prior to this task, recognizing and classifying drug entities themselves is another task. DDI corpus 2011 and DDI corpus 2013 are two corpuses for respective challenges that were manually annotated by organizers [44] [45].

As the last and most recent workshop in biomedical text-mining domain, we can name BioASQ that is focused mainly on large-scale biomedical semantic indexing and question answering [46]. BioASQ challenges include some tasks and sub tasks related to information retrieval, question answering from texts and structured data, machine learning, hierarchical text classification and so on [46].

2.2 Named Entity Recognition (NER) and Named Entity Classification (NEC)

The objective in NER is to detect named entities in text from different natures like news or biomedicine. In general, in a NER system, a word or a combination of words will be labeled as named entity (NE). In the widely used Newswire domain, this means detection of names of persons, locations etc. Named Entity Classification (NEC) is the next step following NER where a specific class is assigned to each recognized named entity.

2.2.1 Biomedical Named Entity Recognition (BioNER)

BioNER is generally a NER task in the biomedicine domain. In the context of BioNER, usually recognition of entities like drugs, chemical compounds, genes, proteins etc. is considered to be the main goal [47]. Usually the NER task is followed by another task for discovering the relations or interactions between previously found NEs such as drug-drug interaction (DDI) or protein-protein interaction (PPI) and alike [17].

2.2.2 Drug Named Entity Recognition (DrugNER)

DrugNER can be considered as a more specific application of BioNER that focuses specifically on recognition of drug entities in the biomedical literature, which in most of the cases refers to chemical substances that are used in pharmacology for prevention, diagnosis and treatment of diseases [48]. Drug-NER can be considered as an important component of research and development sections in Pharmaceutical industries because of its ability to help specialists who work in that section to manage big biomedical data that need to be explored before and after drug production for reasons like dealing with possible interactions between drugs or improving the effects of drugs. Main motivations behind developing DrugNER systems are discovery information integration, translational medicine and safety, text mining and information extraction, search and document management, integrated healthcare and semantics in electronic health records and clinical information management etc. [49][50]. DrugNER is an important part in biomedical natural language processing (BioNLP) tasks including extraction of pharmaco-genomic, pharmaco-dynamic and pharmaco-kinetic parameters [9]. It can be followed by DrugNEC that includes classifying the drug entities which has already been discovered in text. We can name DDIExtraction 2013 and DDIExtraction 2011 challenges as recent works focused especially on this task. The other example would be C-SHALS challenges that are dedicated to semantics systems with focus on healthcare and life sciences. According to the discussions in the C-SHALS 2008 challenge, [51] main problems that should be faced and answered for the DrugNER task are using semantics for discovering drug mentions in order to reduce Phase 2 attrition, use of semantics to help pharmacologists or pharmacy industry in general to understand compound efficacy and safety of drugs. Patient record standardization, healthcare policy management, adverse event capturing / handling and problem of alternative indications discovery are mentioned to be areas of work in this challenge [51].

2.3 Methods Used in Recognition and Classification of Named Entities

2.3.1 Dictionary Based Approaches

The basis of this approach involves looking up a token in a database, here referred to as a dictionary that has already been formed using different corpora. The existence of a token in the dictionary marks it as a named entity. From the NER and NEC tasks participants' point of view, this dictionary can be added as a component of the system. Examples of dictionaries that can be found online are different kinds of ontologies in that system's specific domain. In the bioinformatics domain, we can name these ontologies: Standards and Ontologies for Functional Genomics (SOFG), Ontology for Biomedical Investigations (OBI), Plant Ontology (POC), Master Drug Data Base (MDDB), National Drug File (NDF) and so on [52]. One most frequent method that is used for looking up a token inside dictionaries in NER tasks is simply exact matching. Some other methods can be partial matching in which just matching few letters or words of the token with the one in dictionary is sufficient. Matching based on stemming or lemmatization are other methods that are used in dictionary-based NER. Dictionary based approaches usually have high precision but suffer from low recall.

2.3.2 Rule Based Approaches

Rule based approaches use a set of handmade rules and patterns to detect named entities in text. The application of these rules to new domains is usually very difficult. This is a significant drawback in the biomedical domain since naming conventions often vary among different research groups.

2.3.3 Machine Learning Based +Approaches

In this approach, a learning algorithm is used in order to train the system with set of labeled train data and test the system on unseen data for the labeling of named entities. The basic principle in this approach is based on two main phases: train phase and test phase. In the train phase, a model is made using the labeled data and during the test phase, the model is applied on new and unlabeled data for predicting the named entity labels. This annotation task of the train data that is a preprocessing task, is usually performed by hand and requires work of some experts in the field of interest; for example a pharmacist in the case that NER is being performed on a chemical corpus. Two common supervised learning approaches that are widely used in DrugNER are Conditional Random fields (CRFs) and Support Vector Machines (SVMs).

2.4 Data Sources for DrugNER

2.4.1 Databases and Dictionaries

PubMed [53], which includes more than 24 million citations for biomedical literature from Medline[12], life science journals and online books, serves as the primary source for data in the biomedical text mining field. Drugbank [13] can be named as a specific example for drug related data with online access which contains 7740 drug entries where each entry contains more than 200 data fields. Half of the information kept is devoted to drug/chemical data and the other half is devoted to drug target or protein data. Other example would be PubChem [54], which includes substance information and compound structures, bioactivity data in Pcsubstance [55] as well as Pccompound [56] and PCBioAssay [57] databases. Pcsubstance database contains more than 140 million records, Pccompound contains more than 51 million unique structures and PCBioAssay contains more than 1 million BioAssays. Another online dictionary of chemical entities available online is ChEBI [58]. ChEBI is a freely available dictionary of molecular entities focused on "small" chemical compounds. A final example of databases available online is the Medical Subject Headings (MeSH) [59]. As stated in its webpage, MeSH is a thesaurus with focus on NLM controlled vocabulary that is often used for indexing articles from PubMed.

2.4.2 Labeled Corpora

In this section, we review some different corpora that are annotated based on drug names and especially drug-drug interactions (DDIs). Annotated or labeled corpus is one of the most critical resources that are needed in the field of biomedical text mining. There are some kinds of corpora that are labeled (annotated) based on some different aspects of tokens which are usually words but can be sentences etc. Annotation is usually performed based on semantic aspects of contents of corpus but sometimes it might be performed according to lexical and grammatical aspects of them [60]. Cohen et al. claim that annotation of a corpus based on structural and linguistic features of its contents will result in a high quality corpora that will be more useful in biomedical research [61]. We can name several labeled corpora in biomedical domain, for example Genia corpus [62]. This corpus is made from research abstracts in Medline database. Substances are classified based on their both biological roles and chemical structures. A special ontology is defined for their work in which there were three main categories as source, substance and other. Substance category was focused on chemical structures while source corresponded to biological location that those substances are placed and their reactions happen and other category was for those that does not belong to any of first two categories. There are several sub categories for each one of them also like names of atoms, proteins, DNAs, RNAs etc. that are subcategories of substances and as subcategories of sources, organisms, body parts and tissues etc. [62].

As another example of labeled corpora, we can mention GENETAG [63] that is made from twenty thousand sentences that are tagged with gene/protein names from Medline abstracts. Lorraine Tanabe et al. have classified words (tokens) into four classes as domains, complexes, subunits and promoters in which domain means a discrete portion of a protein with its own function, complex means combination of two or more compounds into a larger molecule in a way they do not bind, subunit means a single biopolymer separated from a larger structure and promoter refers to a segment of DNA [63].

Another example is Clinical E-Science Framework (CLEF) [64]. This corpus is made from clinical texts like clinic letters, radiology, and histopathology reports that are from two categories of structured records and free text documents from 20,234 deceased patients. Those free text documents are from three different sources namely clinical narratives, histopathology reports and imaging reports. In CLEF, nine entities such as condition, intervention, result etc. are modeled and built. Sixteen different relationships between these nine entities are defined such as “has-indication”, “has-finding”, “has-target” and “has-location”. Some properties are also defined for each entity that must be extracted during the annotation process [65].

All of corpora that are mentioned above are annotated and labeled by pharmacological experts with semantic categories that are related to molecular biology domain like protein, gene, drugs or diseases. Because of the need to extract semantic and lexical information from corpora for the annotation purpose, linguistic rules should be applied [60]. As an examples of a corpus that is annotated specifically with drug entities, we can name BioText [66]. For building this corpus, Barbara Rosario et al. used first 100 titles and first 40 abstracts from each of the 59 Medline 2001 documents [67]. They have defined two classes as treatment and diseases and according to them seven different relations between those two classes are specified. Namely "Cure" that means treatment T cures disease D, "Only DIS" that means no treatment is mentioned for disease D, "Only TREAT" that means no specific disease is mentioned in the sentence, "Prevent" that means treatment prevents a specific disease, "Vague" that means a very unclear relationship between treatment and disease, "Side Effect" that means it is mentioned in the sentence that a specific disease is made because of a treatment and finally "No Cure" that indicates in the sentence, it is mentioned that a treatment does not cure a disease [67].

As we can see comparing these corpuses that are reviewed briefly so far, based on a specific task, motivation and field of interest of those building a corpus, there are different classes and therefore different types of relations between them that a corpus should be labeled with respect to them [68].

In order to show the differences in types of entities and relations in different corpora in a more precise manner, we can name Adverse Drug Effect (ADE) corpus [69], Exploring and Understanding Adverse Drug Reactions (EU-ADR) corpus [70] and Tissue Expressions and Protein–Protein Interactions (ITI TXM) corpus [71] as corpora that use just one single entity for labeling drugs and chemicals but BioCaster corpus

[72] makes difference between substances that are supposed to be for treatment of diseases and chemicals that are not considered to be for medication [68].

Another example of related work that has been done recently with focus on medical corpus annotation is PK corpus [73] [74]. Heng-Yi Wu et al. has manually annotated a corpus consists of four classes that are namely “in vivo pharmacogenetic studies”, “in vivo pharmacokinetics studies”, “in vitro drug interaction studies” and “in vivo drug interaction studies”. They used several databases like Human Cytochrome P450 (CYP) Allele Nomenclature Database [75] for extracting enzyme names and genetic variants, Transporter Classification Database [76] for mapping transport proteins’ names and Drugbank 3.0 [77] for creating drug names. They annotated three layers of pharmacokinetics information within their manually annotation process that were namely key terms, DDI sentences and DDI pairs in which DDI sentences annotation depend on key terms and DDI pairs annotations depend on both two others. They defined drug names, enzyme names, PK parameters, numbers, mechanisms, and change as key terms in which mechanisms mean drug metabolism and interaction mechanisms and Change indicate the change of PK parameters.

Two closest annotated corpuses to the DDI corpus [68] are PK DDI corpus [78] [79] and the corpus that is developed by Rubrichi and Quaglini for their work [80] [60]. PK-DDI corpus was created from FDA-approved drug package inserts (PIs). They divided PIs into two main categories as those before 2000 and those after this year and labeled them accordingly as “older” and “newer”. DailyMed [81] was used as the source of PIs. For the annotation purpose, they focused specifically on pharmacokinetic (PK) DDIs. As a step before annotation, they defined a scheme to model drugs into role and type classes as their characteristics. Type itself has three subcategories as active ingredient, drug product and metabolite. Role itself also has two subcategories as object and precipitant. They also defined two properties to model PK-DDIs: The first property indicates existence or absence of some words about observed effects of those two drugs in that statement which has already been discovered to contain PK-DDIs. The second property indicates whether there is quantitative or qualitative information about interaction or lack of interaction in the statement being annotated [80]. S. Rubrichi et al. created a corpus made of 100 manually annotated interactions derived from monographs of Farmadati Italia Database [82]. They used thirteen semantic labels namely "Posology", "PharmaceuticalForm", "InteractionEffect", "OtherSubstance", "PharmaceuticalForm", "OtherSubstance", "IntakeRoute", "ActiveDrugIngredient", "AgeClass", "ClinicalCondition", "DiagnosticTest", "PhysiologicCondition", "RecoveringAction" and "None". None label is to indicate those drugs that are not relevant to DDI interaction topic.

DDI corpus is a gold standard corpus that is manually annotated especially for DDI Extraction 2013 task [68]. According to those involved in its creation, this corpus is developed with the purpose of assisting information extraction techniques applied to drug named entity recognition and drug-drug interaction detection from pharmacological texts by creating a common framework for evaluation of their performances. This corpus is made of 1,025 documents from Drugbank [77] and Medline [12] databases.

Texts that are derived from Medline and Drugbank are from two different sources therefore in the process of annotation they have been dealt with differently. Documents that Drugbank is their origin, has a language more like PIs that is less technical and are focused mostly on description of DDIs but Medline abstracts has a more scientific and complex language that go more to the details and explaining different aspects around the subject. Four classes for drugs are specified: drug that is a generic name of drug, brand that corresponds to those drugs which are usually mentioned in biomedical literature with their brand name, group which is a group of drugs that usually come together in biomedical literature and drug_n that is for those chemical substances that are known as drugs but are not suitable for human use. Mechanism, effect, advice and interaction are also four classes of DDIs in this corpus [60].

2.5 Recent Related Work

There has been a considerable amount of reserach in the area of DrugNER in recent years. Main work in this area is summarized below according to the methods used.

2.5.1 Conditional Random Fields

Tim Rocktaschel et al. participated in SemEval 2013 NER task using a system based on CRFs and which used different groups of features in different runs and compared the results. They trained and tested their system on the given DDI corpus (dataset 2013). They used general features and also some domain-specific features which were extracted from the output of components of Jochem and ChemSpot as well as ontology based features that they constructed from PHARE and the ChEBI ontology. They conclude that by using domain-specific features, performance of chemical NER systems increases. They achieved an F-score of 0.71 when the system was tested on both Drugbank and Medline datasets together and 0.87 and 0.58 respectively when tested on Drugbank and Medline alone [5]. Anup Kumar Kolya et al. introduced a temporal information extraction system based on a CRF approach for participating in the TempEval-3 task. They chose an implementation of CRF for this work, named as CRF++. This is the same implementation of the CRF that is used in this thesis. They trained their system on the given DDI corpus. They used variety of features including morphological features, syntactic features, wordnet features and features based on semantic roles. Their system was tested and evaluated on the TempEval-3 Platinum data [83]. Their system achieves an overall F-score of 0.86 based on relaxed match scheme and 0.75 based on strict match scheme [84]. Stefania Rubrichi et al. participated in DDI-Extraction2011 challenge and used CRF as a part of their hybrid method which uses a CRF and a rule based technique. They trained their system on the given train dataset provided by challenge organizers and it was tested and evaluated by the challenge organizers on test dataset [85]. In the pre-processing step, they used different features such as, orthographical features, Part of Speech (PoS) punctuations, semantic features and context features with window size of three. Their CRF based system achieved F-score of 0.3695 [7]. Another recent work on chemical compound and drug name recognition, which makes use of CRFs, is the work of Andre Lamurias et al. They participated in the BioCreative IV challenge and used both the CHEMDNER and DDI corpus dataset for their work. They used Mallet as the implementation tool for CRF. They have also made use of ChEBI ontology in their work. They used classifiers that were obtained by applying cross-validation on training set that was provided by challenge organizers to train some Weka classifiers using different methods. Random forests method returned the best performance so they used it on their test set predictions. They did five runs corresponding to each subtask. For first run, they used all the classifiers. For second run, they used those classifiers that were trained with the CHEMDNER corpus with a confidence score and ChEBI mapping score threshold equal to 0.8, for third run they used all classifiers' results including those that were trained on the DDI and patents documents corpus. For fourth run, they used all classifiers that were trained with the CHEMDNER corpus but they omitted those that had a semantic similarity measure lower than 0.6 and for fifth run, they did the same thing they did in fourth run only this time all of the classifiers were used. Their best F-score was 0.79 [86].

2.5.2 Support Vector Machines

Md. Faisal Mahbub Chowdhury et al. introduced a system based on SVMs during their participation in the SemEval 2013 DDI detection and classification task. They used a filtering method in which they discard less informative instances by using semantic roles and contextual evidence. Then they train the system on the remaining training instances. They trained and tested their system on the given DDI corpus (dataset 2013). They apply hybrid kernels using the SVM-Light-TK toolkit [87]. They used contextual and shallow linguistic features to train the binary SVM classifier. Their system obtained the overall F-score of 0.80 for detection of drug-drug interaction and 0.65 for DDI detection and classification [88]. Behrouz Bokharaeian et al. participated in Semeval 2013 DDI Extraction challenge and used a combination of different kernels in SVM and added linguistic and dependency tree features to them. They trained and tested their system on the given DDI corpus (dataset 2013). They have used the following feature groups: Word features, morphosyntactic features (PoS lemma and PoS stem), constituency parse tree features and conjunction features and their combinations. Their system achieved 0.54 F-score [89]. Majid Rastegar-Mojarad et al. participated in DDIExtraction-2013 shared task of classifying Drug-Drug interactions and used an SVM classification approach using another implementation of SVM, known as LibSVM [128]. They trained and tested their system on the given DDI corpus (dataset 2013). Features that they used include stemmed words, lemmas, bigrams, PoS tags, verb lists and similarity measures. Their system has 0.47 F-score [90].

Negacy D. Hailu et al. 2013 participated in SemEval-2013 Task 9.2. They used an SVM based approach for this Drug-Drug interaction detection task. They trained and tested their system on the given DDI corpus (dataset 2013). They also used the LibSVM tool for this purpose. To deal with multiple classes' problem in SVM, they used the one vs. all multi-class classification technique. They used three groups of features: Morphosyntactic features (distance feature), PoS tags and dependency parser related features, lexical features such as bigrams and semantic features such as interaction words. Their system achieved 0.50 F-score in DDI detection and 0.34 F-score in classification task [91].

Anne-Lyse Minard et al. presented an SVM based system in the DDI extraction 2011 challenge making use of LibSVM and SVMPerf [92] tools. They trained and tested their system on the given DDI corpus (dataset 2011). They extracted classical and corpus-specific features and used feature selection before they train their system with a subset of features. The features selected were surface features, which provide information about position of the two drugs in the sentence, lexical features, morpho-syntactic features, semantic features and corpus-specific features. Their best system obtained an F-measure of 0.5965 [93].

2.5.3 Dictionary Based Approaches

As a recent work related to this approach in the biomedical NER domain, we can mention the work of Daniel Sanchez-Cisneros et al. They participated in task 9.1 of Semeval 2013, which is recognition and classification of drug names. Their system works in both NE recognition and NE classification tasks. During the NER phase, they used an analyzer named Mgrep for sentence by sentence analysis of the DDI corpus. They mention that by using Mgrep, they can obtain information about the ontology concept recognition, term information and snippet of the original text. In the NER phase, they design a rule based system by extracting some rules from resources like Drugbank, Pubchem, ATC Index, Kegg and MESH. They tested their system on DDI corpus 2013 test dataset. Their system achieves F-score of 0.52 for NEC task and 0.60 for NER [48]. The work of Isabel Segura-Bedmar et al. can be considered as another example of a DrugNER system using dictionary based methods. Their system is utilized in both tasks of DrugNER and drug name classifiacation. They used PubMed as the main data source. They created DrugDDI corpus consists of 849 medical abstracts that were downloaded from PubMed by getting a query of word "drug interaction" and used it to evaluate their system. As they stated in their paper, this system is a combination of some rules that are extracted from two different sources: MetaMap Transfer (MMTx) program, which works based on the Unified Medical Language System (UMLS) and stems recommended by the World Health Organization International Nonproprietary Names (WHOINN) Program. They worked on a corpus made of 849 abstracts that were downloaded from PubMed by submitting the query "drug interaction". Their system achieved a very good performance using only the MMTx program with 0.975 recall and precision equal to 1. Using a combination of MMTx program and stems, the system achieved a recall of 0.99 and a precision of 0.99 [94].

Chapter 3

SYSTEM OVERVIEW

3.1 The Architecture of NERC System

We present a machine learning based system for drug name recognition and classification using the DDI-Corpus [68]. The details of the system will be described in details in this chapter. The system uses both SVM and CRF algorithms for the classification task. Two different approaches are implemented for improving the performance of the NER system. The first approach is based on feature selection using wrapper based algorithms. The second approach is based on combination of classifiers selected using wrapper based algorithms. The implementation details of both methods is discussed in Section 3.4 and 3.5 respectively and the results obtained from the two different approaches are compared in Chapter 4. In both approaches the system makes use of a tokenizer which is based on white space and some lexical rules and tokenizes the text which is originally in the XML format. During tokenization the resulted tokens are tagged as entities in the IOB2 format [95] using the exact offset of the drugs provided in the DDI-Corpus. The next step involves feature extraction. Feature subset selection follows this step in the first approach. In both systems, there is a training phase and a test phase. 3-fold cross validation on train data is used to get the performance of each individual classifier. For evaluating the performance of system, it is trained using the full train data and the model is tested using the test data to predict the classes. Precision, recall and F-score [96] are used as the performance measure. In the second approach where selection is employed, majority voting algorithms are used to combine and select the best classifiers in each system.

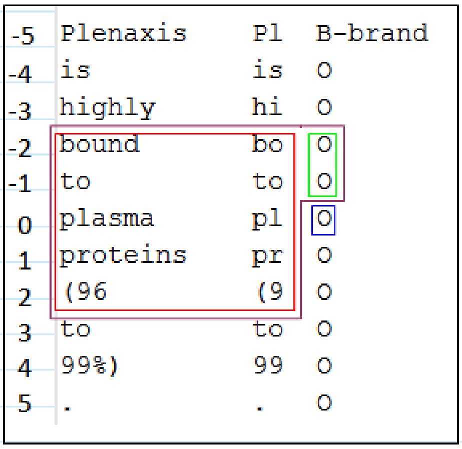
3.1.1 SVM

As Vladimir Vapnik suggests, support vector machine (SVM) is a specific learning procedure that relies on statistical learning theory [10]. SVM can be considered as a binary classifier which by finding the optimized hyper lane divides the input space into two classes. Optimized hyper lane is the one that has the maximum margin from the support vectors [97]. Another important concept in support vector machines is the use of kernels. Kernels are used when input space is not linearly separable. In this case, by using kernels we map the input space into a feature space that is now linearly separable. One of the most famous kernels that are used in NER tasks when using SVM is polynomial kernel. Two other well-known kernels are namely Gaussian and Sigmoid [98]. By default, SVM is designed to solve binary class problems. In order to adopt it for multi class problems, two solutions have been proposed. First solution is one versus rest and the second one is pair wise combination [99].

3.1.1.1. Using YamCha for SVM Implementation

In this work, we used Yet Another Multipurpose CHunk Annotator (YamCha) [100] as one tool to train and test already tokenized data derived from Medline and Drugbank datasets. YamCha is known as a general purpose, adjustable and freely available text chunker that has been used for plenty of NLP tasks, such as Named Entity Recognition, POS tagging, Text Chunking and base NP chunking. YamCha uses TinySVM [101] as its learning algorithm. It only supports polynomial kernels [100]. As a work that has been done using this tool, we can name CoNLL-2000 Shared Task [102].

These are characteristics of this chunker: one important requirement of YamCha is that the format of train and test data file should be the same. Format of input file should be as follows: First column must contain tokens, second column until the one before last column should be associated to features and the last one will be for class labels. Columns must be separated by spaces and an empty line indicates the end of the sentence. There is no limitation in number of features. Another advantage of YamCha is that we can define and change the window size in it. In addition, there is an option to train the SVM based on both static and dynamic features. Dynamic features here are those that include class labels [100]. For a better understanding of window size option and static and dynamic features, we describe them in an example that is a sentence from Drugbank corpus. Let's consider the default window size and feature space that is: "F:-2..2:0.. T:-2..-1", in this command, F defines the static features boundaries and T defines the dynamic ones.



```
-5  Plenaxis    Pl  B-brand
-4  is          is  O
-3  highly      hi  O
-2  bound       bo  O
-1  to          to  O
 0  plasma      pl  O
 1  proteins    pr  O
 2  (96         (9  O
 3  to          to  O
 4  99%)        99  O
 5  .           .   O
```

Figure 3.1: Illustration of Window Size and Static/Dynamic Features in YamCha

In figure 3.1, the red square indicates the window size and static feature space. That here is from token and features in line "-2" until token and features in line "2". Green square indicates the dynamic features and the purple one demonstrates all the data that is being processed for training and predicting the class of current token that is line "0" (blue square) [100]. Here, dynamic features are actually classes of previous tokens.

Another option that should be discussed with more details is MULTI_CLASS option that can enable user to define the nature of the multi class problem and change it from pair wise case that is the default case, to one vs. rest problem [100]. There is another option that provides the user the opportunity to have output file in two formats, one that is the default case, only has the predicted class with its score in the last column but the other case shows all the existent classes with their corresponding scores. Score of the class in multi class problem has two meanings, if the problem is pair wise, score means summation of distances of this class and if it is one vs. rest case, score is the distance from the separating hyper lane [100].

3.1.2 CRF

In NER tasks, If we consider tokens that are made previously from the text and now are ready to be labeled, as input sequence X, in the way to find their corresponding labels that here we consider them as label sequence Y, we can use Conditional Random Fields (CRFs) [11] to calculate the probability P(y|x). A CRF in this context is considered to be a probabilistic, undirected graphical model [103]. A CRF can be shown in the form of a graph in which nodes are random variables and their relationships are represented as the edges. A linear chain CRF that is a common classifier tool and is used in many NER tasks, can be depicted as a graph in which nodes can be either one of token sequence members (usually shown as X) or label sequence members (usually shown as Y). These X nodes are connected to their corresponding Y nodes and Y nodes themselves are connected to each other's neighbors. Features used in linear chain CRFs can be considered as encoders of relationships between the nodes that are represented by edges in the graph [11].

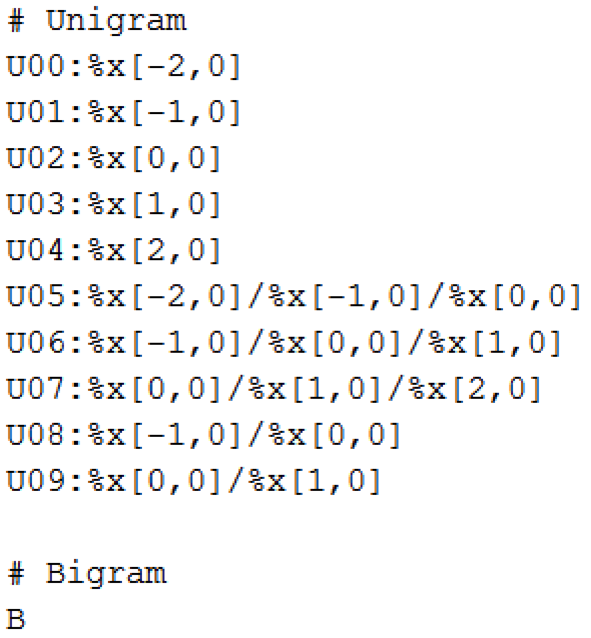
3.1.2.1 Using CRF++ as CRF Implementation

In this work, we used CRF++ as another tool to train and test already tokenized data derived from Medline and Drugbank datasets. CRF++ is an open source implementation of Conditional Random Fields (CRFs) for segmenting and/or labelling sequential data [104]. CRF++ is designed as a tool with comprehensive capabilities so that it can be applied to a vast range of NLP tasks such as Named Entity Recognition, Information Extraction and Text Chunking [102] [104].

Like YamCha, train and test files should be in the same format. Format of input file should be as follows: First column must contain tokens, from second column until the one before last column should be associated for features and the last one will be for class labels [104].

Columns must be separated by spaces and an empty line indicates the end of the sentence. There is no limitation for the number of features that can be defined but when defined, all of the tokens should have the same number of features as the first token has [104].

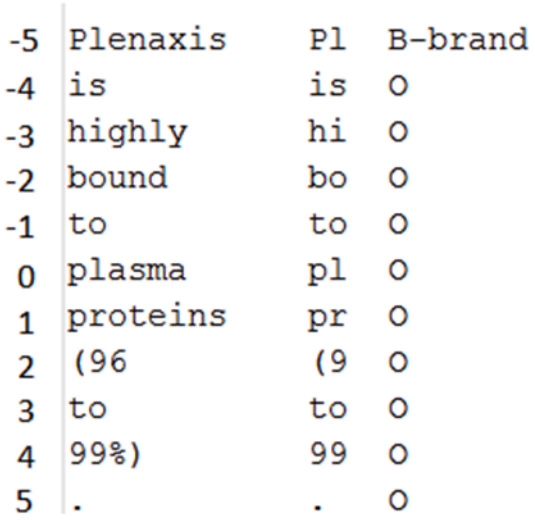
One major preparation task for using CRF++ is to make the proper template file for input data file. Here we describe the important parts of it and specific characteristics of this file.



```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-2,0]/%x[-1,0]/%x[0,0]
U06:%x[-1,0]/%x[0,0]/%x[1,0]
U07:%x[0,0]/%x[1,0]/%x[2,0]
U08:%x[-1,0]/%x[0,0]
U09:%x[0,0]/%x[1,0]

# Bigram
B
```

Figure 3.2: Illustration of a CRF++ Template File

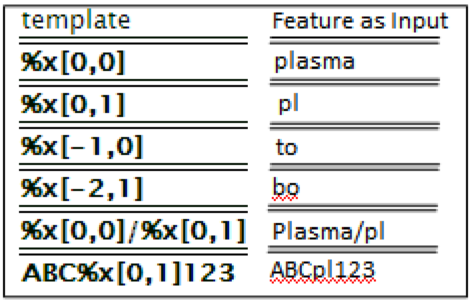


```
-5  Plenaxis    Pl  B-brand
-4  is          is  O
-3  highly      hi  O
-2  bound       bo  O
-1  to          to  O
 0  plasma      pl  O
 1  proteins    pr  O
 2  (96         (9  O
 3  to          to  O
 4  99%)        99  O
 5  .           .   O
```

Figure 3.3: Example of a Train File with Tokens, Features and Labels

Figure 3.2 is an example of a template file for CRF++. Figure 3.3 is an example of a train file that CRF++ gets as input. In figure 3.3 "plasma" is the current token. In template above, a context window with size 2 is defined. U00 and U01 etc are unigram templates that define the feature space. If we want to have a bag of word feature, there is no need for identifiers like 00 or 01 etc. In this case, all the features will be seen by

CRF++ as one string altogether [104]. We can define two types of templates, one as unigram that its identifier starts with “U” and the other one is bigram that its identifier is “B”. Bigram features are for adding combination of the current output token and previous output token into current unique features that are extended. This type of template may cause inefficiency when dealing with large input data [104].



template	Feature as Input
%x[0,0]	plasma
%x[0,1]	pl
%x[-1,0]	to
%x[-2,1]	bo
%x[0,0]/%x[0,1]	Plasma/pl
ABC%x[0,1]123	ABCpl123

Figure 3.4: Illustration of Extended Features as Input of CRF++

In figure 3.4, we can see the extended feature that CRF++ gets as input according to the contents of the template in left column [104]. As for additional features of CRF++, we can name an option that enables us to run the program on multiple CPUs if available, an option that can change the hyper-parameter for the CRFs and an option to set the cut-off threshold for the features. This option is very useful when data is very big in terms of number of features because number of unique feature sets that can be made by CRF++ will become very large and this consumes lots of memory. By defining bigger threshold value, the memory consumption becomes lower [104].

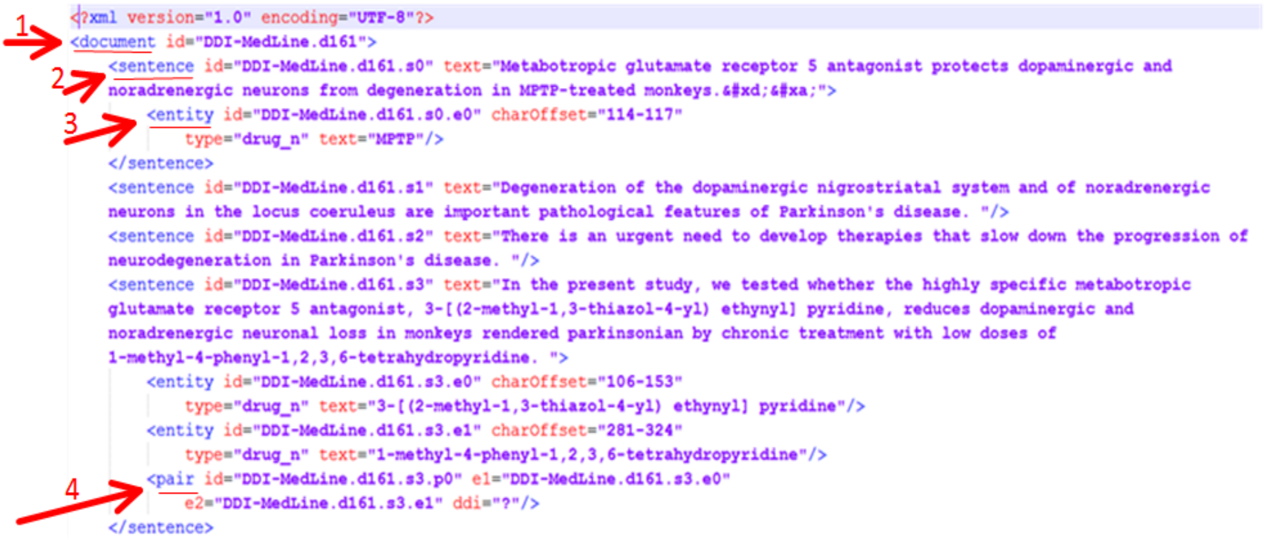
3.2 Data Used

In this work, the DDI corpus is used for training and testing the classifiers. The DDI corpus is a corpus made of pharmacological entities as well as their possible interactions [68]. It also contains pharmacodynamic (PD) and pharmacokinetic (PK) DDIs. The first one occurs when effects of one drug are modified by the other one and the second happens when one drug interferes the mechanism and actions of the other one inside consumer's body. This corpus has been manually annotated specifically for DDI Extraction 2013 challenge [1] with the focus on making a framework for evaluation of Drug-NER systems and also DDI detection systems [105]. Most of this corpus is based on a previous version of this corpus named as DDI corpus 2011 [106]. The entire corpus is made from texts from Medline and Drugbank databases therefore it consists of two different sub-corpuses: DDI-Drugbank corpus and DDI-Medline corpus. The whole corpus consists of 1,025 documents (792 Drugbank and 233 Medline) with a total of 18502 annotated entities and 5028 DDIs. The corpus is divided into two separate sections, one for training and the other one for testing. Training part consists of 714 texts (572 from Drugbank and 142 Medline abstracts). The test dataset for the Drug NER subtask that we use in this thesis consists of 52 Drugbank texts and 58 Medline abstracts. On the other hand, the test dataset which was used for the DDI extraction subtask, consists of 158 Drugbank Texts and 33 Medline abstracts [107]. Table 3.1 summarizes the data used.

Table 3.1: Summarization of Data

Corpus	**Medline**		**Drugbank**	
	Train	Test	Train	Test
Number of texts	142	58	572	52
Total	200		624	

Each dataset in this corpus is in XML format and has an appearance that shown in figure 3.5.



```xml
<?xml version="1.0" encoding="UTF-8"?>
<document id="DDI-MedLine.d161">
    <sentence id="DDI-MedLine.d161.s0" text="Metabotropic glutamate receptor 5 antagonist protects dopaminergic and
    noradrenergic neurons from degeneration in MPTP-treated monkeys.&#xd;&#xa;">
        <entity id="DDI-MedLine.d161.s0.e0" charOffset="114-117"
            type="drug_n" text="MPTP"/>
    </sentence>
    <sentence id="DDI-MedLine.d161.s1" text="Degeneration of the dopaminergic nigrostriatal system and of noradrenergic
    neurons in the locus coeruleus are important pathological features of Parkinson's disease. "/>
    <sentence id="DDI-MedLine.d161.s2" text="There is an urgent need to develop therapies that slow down the progression of
    neurodegeneration in Parkinson's disease. "/>
    <sentence id="DDI-MedLine.d161.s3" text="In the present study, we tested whether the highly specific metabotropic
    glutamate receptor 5 antagonist, 3-[(2-methyl-1,3-thiazol-4-yl) ethynyl] pyridine, reduces dopaminergic and
    noradrenergic neuronal loss in monkeys rendered parkinsonian by chronic treatment with low doses of
    1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine. ">
        <entity id="DDI-MedLine.d161.s3.e0" charOffset="106-153"
            type="drug_n" text="3-[(2-methyl-1,3-thiazol-4-yl) ethynyl] pyridine"/>
        <entity id="DDI-MedLine.d161.s3.e1" charOffset="281-324"
            type="drug_n" text="1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine"/>
        <pair id="DDI-MedLine.d161.s3.p0" e1="DDI-MedLine.d161.s3.e0"
            e2="DDI-MedLine.d161.s3.e1" ddi="?"/>
    </sentence>
```

Figure 3.5: Illustration of an XML document and its Elements and Attributes

As we can see in this figure, each XML file consists of 4 elements namely document, sentence, entity and pair. Document element is the root element that has “id” as its attribute. This attribute indicates which corpus this document belongs to, Drugbank or Medline, and what is its exact identifier in that database. Sentence element includes “id” and “text” attributes that first one indicates the exact “id” of that sentence and the second one contains the sentence itself. Entity element provides us information about the drugs that are present in the sentence. It has “id” attribute that provides the drug’s identification and number in that sentence. The other attribute is “charOffset” that provides the exact location of that drug in the sentence. “text” attribute is another attribute of Entity element that contains the name of drug itself. The final attribute is “type” that indicates type of that drug. The last element is “pair” that provides information about two drugs in the sentence that might interact with each other. It consists of id, e1, e2 and ddi attributes. e1 and e2 are two drugs’ identifiers and ddi is a binary attribute that indicates whether an interaction exists or not [44]. Table 3.2 shows a summary about all elements and their attributes in the XML file. There are four types of entities in this corpus: “drug”, “drug_n”, “group” and “brand” that are explained in more details next.

Table 3.2: Summary of XML Format of Corpus

Element’s name	**document**	**sentence**	**entity**	**pair**
Attributes of element	id	id, text	id, charOffset, text, type	id, e1,e2, ddi

3.2.1 Drug

According to comments that builders of DDI corpus has made on drug entity, drug is any chemical that is used as a cure for any sickness and can be taken by humans. This type of entity has a name like a chemical name not a brand or commercial name. Therefore, any pharmacological material that is not a brand name or a group of drugs should be labelled as drug entity. Usually drug names should be in Anatomical Therapeutic Chemical (ATC) system [105] that provides the INN of chemicals. Some other references that can be checked for validity of a drug name are FDA, EMA, AEMA, Drugbank etc. [105]. Drug names can be in these forms: generic names, chemical names, abbreviations, synonyms, salts, alcohol, stereoisomer, etc.

3.2.2 Brand

Any pharmacological name that is a commercial or brand name should be labelled as brand entity. First letter of these names usually is in uppercase form [105].

3.2.3 Drug_n

Any drug that is not approved for human use must be labelled as drug_n. This type of drug is important to be classified separately because there are so many cases of interactions between drugs and chemical substances that are mentioned in biomedical literature as not intended to be used by humans. drug_n entities can be in these forms: experimental drugs, animal drugs, endogenous substances that are made inside an organism, toxins, excipients, metabolites etc. [105].

3.2.4 Group

A group of words in a sentence that their target organ in body is the same or their characteristics and properties are same should be labelled as group. Group entities can be in these forms: those that derived from ATC system, those with MeSH origin, variations and synonyms, nested named entities etc. [105].

3.3 Feature Extraction

The goal of the feature extraction task is to convert a high dimensional input data into a set of features, by removing redundant data and hence reducing the size of the feature space [108]. It is a very important concept used in various pattern recognition tasks, which involve the use of machine learning approaches. Features used in this study are selected with the aim of representing the structural properties of chemical names and drugs. In this respect, features used follow those that are used by many researchers in similar work [109] [110] [111]. The set of features used are shown in Table 3.3 and are presented in more details in the following sections. Most of the features that are made in this work are well-known and common features that are frequently used in many biomedical text mining tasks. But we added two new orthographic features to those other common ones namely BeforeHasParentheses (f2) and BeforeHasBracket (f9). By observing the structure of datasets, we noticed that there are so many cases of chemical formulas and groups of drugs that are located inside parentheses and brackets. For this reason, we decided to obtain characteristics of the previous token regarding the existence or absence of parentheses and brackets. There are also seven frequency based morphological features that are used namely f20, f21, f22, f23, f24, f25 and f26. We will discuss the effects of extraction and using these features in chapter 4.

Table 3.3: Presentation of Extracted Features

Feature Identifier	Feature Name	Type of Feature
f1	FirstLetterIsUppercase	Orthographic
f2	BeforeHasParentheses	
f3	HasBracket	
f4	NextHasHyphen	
f5	HasParentheses	
f6	NextHasColon	
f7	NextHasComma	
f8	NextHasSemicolon	
f9	BeforeHasBracket	
f10	NumbrInside	
f11	HasCaps	
f12	LENGTH	
f13	allLettersUpperCase	
f14	HasHyphen	
f15	HasSlash	

f16	3-GramSuffix	Morphological
f17	2-GramSuffix	
f18	2-GramPrefix	
f19	3-GramPrefix	
f20	10PercentMostFrequent2-GramSuffixs	
f21	10PercentMostFrequent2-GramSuffixsInDrugNames	
f22	10PercentMostFrequent3-GramPrefixsInDrugNames	
f23	10PercentMostFrequent3-GramSuffixsInDrugNames	
f24	10PercentMostFrequent2-GramPrefixsInDrugNames	
f25	10PercentMostFrequent4GramSuffixsInDrugNames	
f26	NPercentMostFrequent3GramSuffixsInDrugNames	
f27	PhrasalCategories	Lexical
f28	PartOfSpeech	

3.3.1 Tokens

These are words or single characters found in the text and correspond to each unit of text after tokenization.

3.3.2 Lexical Features

These features are kinds of features that provide us information about grammatical aspects of the token. Part of speech tags and Phrasal Category are two examples of these type of features that we used in our work. We used GDep [112] for extracting both POS and Phrasal category features. GDep [1] is a dependency parser that is designed specifically for biomedical texts. Here we describe these two features in more details:

3.3.2.1 Part-of-Speech (POS) Tags

In natural language processing, there is usually a necessity to have some lexical information about tokens in order to use them in next steps. That's why Part of Speech tagging becomes an important subtask in NLP tasks. In this special kind of tagging, we assign a suitable part of speech to each token that is already extracted from the corpus. Most important POS tags are nouns, verbs, proper nouns, adjectives, adverbs and determiners. One should notice that POS taggers are language-dependent and this is because each language has its own lexical rules that based on them, tagger must decide which part of speech should be selected for a given token.

Examples of this feature are noun (N), verb (V) and preposition (P) [113].

3.3.2.2 Phrasal Category Feature

This feature provides information about a type of phrase that is in the form of set of words that circle around one unit not just a word. It is an informative feature because helps us find patterns in biomedical texts in a phrasal scale not word by word. Examples of this feature are Noun Phrase (NP), Verb phrase (VP) and Preposition phrase (PP) [113].

3.3.3 Morphological Features

These features provide information about structure of a token. This includes different n-gram suffixes and prefixes of a token. They are simply made of a specific sequence of letters, words, syllables, etc. of a token. If the token is considered as a word, this feature would consist of letters. If the N equals to one it will be called a unigram, If N is two, a bigram, If three a trigram and more than three are called as four-gram and so

[1] http://people.ict.usc.edu/~sagae/parser/gdep/

on. Usually they are used in order to build a probabilistic and/or statistical model upon a given corpus [114].

3.3.4 Orthographic Features

These features describe the appearance of a token like characteristics that provides information about whether token starts with upper case character or has number inside them etc. [115]. The list of orthographic features that are used are given in Table 3.4.

Table 3.4: Presentation of Orthographic Features

Name of Orthographic Feature	Example
FirstLetterIsUppercase	Repeated
BeforeHasParentheses	(IV) injection
HasBracket	N-[N-(3,
NextHasHyphen	contortrostatin - induced
HasParentheses	1,25(OH)2D3
NextHasColon	Jacalin : an IgA-binding lectin
NextHasComma	desipramine , in the nonfailing heart
NextHasSemicolon	by supplementary iron ;
BeforeHasBracket	accumulation of [(14)C] aminopyrine
NumberInside	8-cyclopentyl-1,3-dipropylxanthine
HasCaps	vitamin-D
LENGTH	vitamin-D = 9
allLettersUpperCase	DPCPX
HasHyphen	3-hydroxy-1,4-benzodiazepine
HasSlash	Drug/Laboratory

3.3.5 Dictionary Based Features

These kinds of features are made based on a dictionary that has already been made. That dictionary can be a bag of words or collection of any relevant data like N-most frequent bigrams in the corpus etc. These features can be used in both machine learning approaches and pure dictionary based methods. These features have been proven to have big positive effects on increasing the performance of Named Entity Recognition and classification systems [116]. The features used in this category are as follows:

- Token is among 10 percent most frequent 2-gram suffixes of whole dataset.
- Token is among 10 percent most frequent 2-gram suffixes of drug names.
- Token is among 10 percent most frequent 3-gram suffixes of drug names.
- Token is among 10 percent most frequent 4-gram suffixes of drug names.
- Token is among 10 percent most frequent 2-gram prefixes of drug names.
- Token is among 10 percent most frequent 3-gram prefixes of drug names.
- Token is among N percent most frequent 3-gram suffixes of drug names (here N is between 0 and 9, where 0 means the token doesn't belong to any of these ranks, 1 means it belongs to the first 10 percent most frequent 3-gram suffixes of drug names that are highest frequencies and 9 means it belongs to the last 10 percent that are lowest frequencies).

3.4 Feature Selection

Several reasons have been mentioned for doing feature selection but the main reason is to find the best combination of features known as the best feature ensemble in order to optimize the performance of a recognition and classification system, designed for a specific task [93]. When the output class labels were included in the feature set, feature selection is known as supervised feature selection and unsupervised otherwise [117]. There are various methods used for feature selection but in general, they can be categorized as embedded feature selection, wrapper based and filter approaches. In the embedded feature selection method, features are selected while the training of the system is being performed simultaneously. In the wrapper based method however, selection is performed after training is finished and the class labels are predicted. In this method there is a need for a search algorithm to search for the optimal feature ensemble among all possible feature sets. This requires that the performances of different feature ensembles are evaluated and a decision factor is needed in order to decide where to stop the search. Consumption of resources (time, memory etc.) in wrapper-based selection is higher than embedded one, but in general, efficiency and accuracy of wrapper-based approaches is higher than embedded feature selection. In filter approaches, features are selected based on information that is prior to classification. This means that in a machine learning approach, when using a filter based feature selection method, features should be selected and ensembles will be made before training of the classifiers begin [118] [119]. For feature selection we should have a searching algorithm to collect a subset and a criterion to define the stop point of searching and a method to evaluate the performance of that found subset [120].

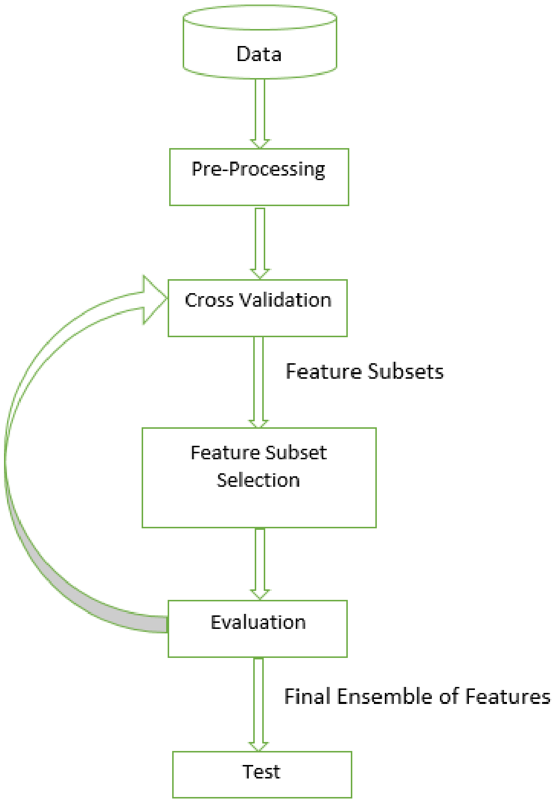


Figure 3.6: Architecture of Feature Selection System

Figure 3.6 shows a general overview of the feature selection system.

3.4.1 Wrapper Based Feature Selection Algorithms

3.4.1.1 Forward Selection (FS) Algorithm

This well-known greedy searching algorithm starts with the single best feature and its performance is considered as the reference value for evaluation of next ensembles. In each iteration, a new feature is selected randomly among all other remaining features and will be combined to previously selected features. If the performance of this new ensemble improves, it will be added to the selected feature ensemble, otherwise it will be discarded and it never will be investigated in the process of finding feature ensembles. The algorithm will continue until all the features are investigated. This algorithm fails to guarantee the optimal solution [121]. Its main weakness is that it is unable to fix the negative effects of ensembles that are selected in previous steps [122].

3.4.1.2 Backward Selection (BS) Algorithm

In this approach, the starting point is the set of all features combined. In each iteration, one feature is to be randomly selected and removed and the performance of the resulting ensemble will be evaluated and compared with the best performance so far. If the performance improves, this feature will be removed, otherwise it will remain, and the algorithm repeats until all the features are checked. This greedy algorithm does not guarantee to find the optimized ensemble [121]. BS consumes many resources due to its extensive computation from the beginning and it usually works better when the number of features are much smaller than the input data [122].

3.4.2 Single Best (SB)

In this approach, each feature is individually selected and used to train the classifier. Obviously, the single feature with the highest performance will be considered as the final selection. Although this heuristic method is the simplest one, it cannot be considered as the optimized option. Generally, the performance of the single feature with maximum value will be considered as a reference for other methods like forward selection. We separately trained CRF and SVM classifiers with all 28 features and evaluated their individual performances.

3.4.3 Grouping

In this method features are grouped according to their types such as lexical, orthographic etc. in order to further investigate the effect of each feature type. The system is separately trained with each group and the performance of each group is evaluated. Finally, a comparison is made between the groups and the best performing group is identified. In this thesis, we arranged all features into three groups as orthographic, morphological and lexical features. We also investigated the effects of combining all six possible combinations of those three groups together.

3.4.4 Combination of All Features

This heuristic method can be considered as a part of the N-best method where N is the number of all features that are being investigated to find an optimized ensemble among them [123]. The performance of this combination of all features can be used as a reference to some other methods like FS and BS. We performed tests with the combination of all 28 features.

3.4.5 Cross Validation (CV)

Cross validation is a method for evaluation of performance of a classifier for a given classification task. It plays an important role when there is no access to evaluation test data or in order to come up with the best parameters or best feature ensembles for the given classification task. In CV, we split the data into N parts, train the system on N-1 parts, and test on the remaining part and we repeat the process until all data parts are tested once [124]. In general, 10-fold CV is used however in this study we performed a 3-fold cross validation instead, mainly due to the large size of data and lack of resources.

3.5 Classifier Selection

The goal behind classifier selection is to choose the best ensemble of classifiers from a pool of all classifiers in order to get the highest classification performance. There are several methods to achieve this goal but in general, they can be categorized into two main categories: Static Classifier Selection (SCS) and Dynamic Classifier Selection (DCS). In Static Classifier Selection, The task of selecting best classifier is performed in training time before testing phase and final classification begins. Therefore in classifier selection phase, there is an optimal selection solution in hand that is fixed. In Dynamic Classifier Selection, selection task is performed during the process of classification and based on evaluation information that are obtained in training phase [125] [126] [127]. In this study, we used wrapper based selection algorithms in a similar manner as the ones used for feature subset selection described in sections 3.4.1.1 and 3.4.1.2. The selected classifiers are then combined according to one of the three different majority voting combination rules explained next.

3.5.1 Simple Majority Voting

In this approach, predication of each classifier is counted as one vote for the predicted class. The class that receives the maximum votes is considered as the predicted class by the ensemble.

3.5.2 Weighted Majority Voting

This is a variant method to the simple approach. The predicted class receives the F-score of the predicting classifier as the vote and again the class that receives the maximum votes is considered as the predicted class by the ensemble.

3.5.3 Ranked-Weighted Majority Voting

In this approach, we combined ranked majority voting and weighted majority voting together. This way we defined median of F-scores of all classifiers as a threshold and if the F-score of a classifier is more than this threshold, difference between two values was considered as a coefficient, *K*. Then the vote of that classifier for the predicted class is calculated as the product of its F-score and its *K* value. For those classifiers that their weights were under threshold, *K* value was considered as 1.

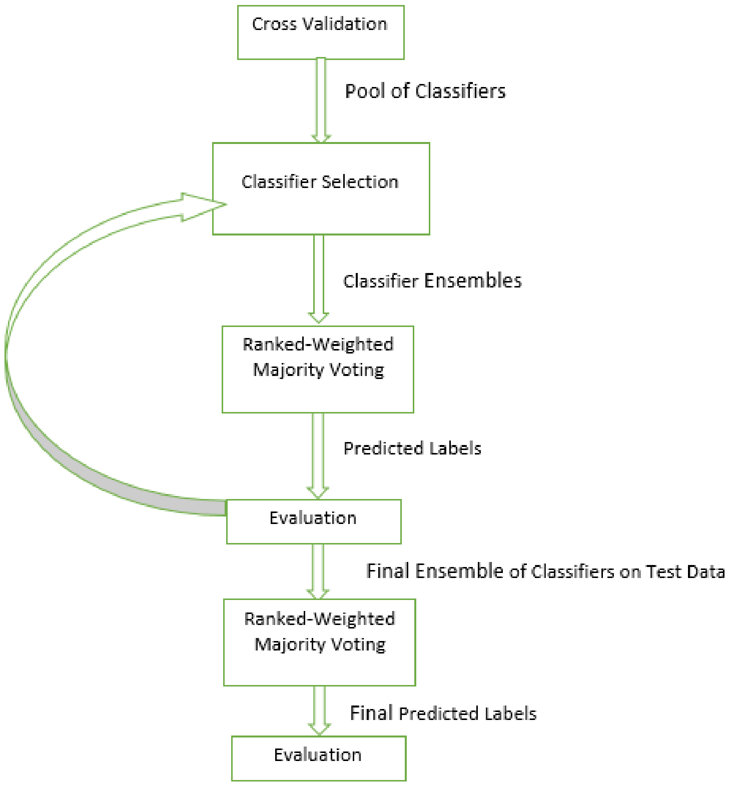


Figure 3.7: Architecture of Classifier Selection System

Figure 3.7 shows a general overview of the classification system.

Chapter 4

RESULTS and DISCUSSION

In this chapter, we present and discuss the results obtained by employing different approaches with the aim of DrugNER. In particular we compare the results obtained using single features, grouped features, feature ensembles obtained using FS and BS approaches and using classifier ensembles obtained using FS and BS approaches. Both CRF and SVM classifiers are employed and all experiments are conducted using the dataset presented in Section 3.2.

4.1 Classification Using Single Features

4.1.1 Entity Classification

We have extracted 28 single features from the datasets and evaluated the performance of both SVM and CRF classifiers individually. The cross validated results obtained will be used in forming ensembles of features and the results received for the test set will serve as a baseline for evaluating the performance of other combinations of features.

Table 4.1 shows the results using single features on CRF based classifiers that are trained on Medline DDI corpus train dataset and tested on its test dataset. As can be seen in this table, feature number 23 has the best performance based on the overall F-score. This feature is among those seven frequency based morphological features and indicates whether a token belongs to the first 10 percent of most frequent 3-Gram suffixes in drug names or not. Feature number sixteen has the highest performance in detection of drug class. This feature provides the 3-Gram suffix of a token. Feature number 1 has the highest performance in detection of Drug_n class. It indicates whether the first letter of a token is uppercase or not. POS feature has the highest performance in detecting group class. The performance of all these classifiers with respect to brand class is equal to zero due to very small number of brand entities in train dataset. In other words, these classifiers fail in classifying any token in test dataset as brand class because the corresponding model file lacks the necessary learnt patterns regarding brand class. The same is true for drug_n class here but the number of this type of entity is higher in the train dataset resulting in slightly better classification performance. It can be observed from this table that classifiers with two new orthographic features (f2 and f9) that were discussed before in chapter 3, achieved F-Scores less than average of F-Score values of all classifiers with orthographic features that is equal to 0.2066. It also can be observed from this table that among those seven classifiers with frequency based morphological features, f23, f26 and f21 achieved an F-Score above average of F-Scores of all 28 classifiers that is equal to 0.2399.

Table 4.1: Classification Performance of CRF Classifiers Using Single Features (Medline corpus)

			CLASS			
Feature No.	**Feature Name**	**Micro-Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
1.	FirstLetterIsUppercase	0.2133	0.3113	**0.0615**	0.0000	0.2157
2.	BeforeHasParentheses	0.1978	0.2936	0.0333	0.0000	0.1980
3.	HasBracket	0.1982	0.3028	0.0168	0.0000	0.1980
4.	NextHasHyphen	0.2058	0.3000	0.0500	0.0000	0.1980
5.	HasParentheses	0.2067	0.3099	0.0323	0.0000	0.2157
6.	NextHasColon	0.2022	0.3091	0.0169	0.0000	0.1980
7.	NextHasComma	0.1946	0.2870	0.0336	0.0000	0.1980
8.	NextHasSemicolon	0.1995	0.3056	0.0169	0.0000	0.1980
9.	BeforeHasBracket	0.1896	0.2870	0.0167	0.0000	0.1980

10.	NumbrInside	0.2108	0.3167	0.0339	0.0000	0.1980
11.	HasCaps	0.2124	0.3192	0.0606	0.0000	0.1980
12.	Length	0.2536	0.3643	0.0168	0.0000	0.2642
13.	allLettersofTokenAreUpperCase	0.2110	0.3070	0.0606	0.0000	0.2157
14.	HasHyphen	0.2022	0.2949	0.0496	0.0000	0.1980
15.	HasSlash	0.2027	0.3105	0.0169	0.0000	0.1980
16.	3-GramSuffix	0.3887	**0.6053**	0.0168	0.0000	0.1980
17.	2-GramSuffix	0.3699	0.5311	0.0331	0.0000	0.2330
18.	2-GramPrefix	0.2298	0.3264	0.0168	0.0000	0.2642
19.	3-GramPrefix	0.2237	0.3005	0.0168	0.0000	0.3119
20.	10PercentMostFrequent2-GramSuffixs	0.1429	0.1865	0.0333	0.0000	0.1980
21.	10PercentMostFrequent2-GramSuffixsInDrugNames	0.2792	0.4067	0.0167	0.0000	0.2308
22.	10PercentMostFrequent3-GramPrefixsInDrugNames	0.1935	0.2785	0.0169	0.0000	0.2115
23.	10PercentMostFrequent3-GramSuffixsInDrugNames	**0.3965**	0.5889	0.0331	0.0000	0.2000
24.	10PercentMostFrequent2-GramPrefixsInDrugNames	0.2103	0.3091	0.0169	0.0000	0.2330
25.	10PercentMostFrequent4Gram SuffixsInDrugNames	0.2338	0.3660	0.0167	0.0000	0.1980
26.	NPercentMostFrequent3Gram SuffixsInDrugNames	0.3811	0.5352	0.0496	0.0000	0.2703
27.	PhrasalCategories	0.2532	0.3837	0.0492	0.0000	0.1980
28.	PartOfSpeech	0.3158	0.4255	0.0476	0.0000	**0.3119**

Table 4.2 presents similar results for classifiers with CRF based features that are trained on Drugbank DDI corpus train dataset and tested on its test dataset. As can be seen from this table, classifier with feature number 16 shows the best performance based on the overall F-score. This feature provides 3-gram suffix of the token. Classifier with feature number 17 has the highest performance in detection of drug class. This feature provides the 2-gram suffix of the token. Classifier with feature number 1 has the highest performance in detection of brand class. The performance of all these classifiers with respect to drug_n class is equal to zero due to very small number of drug_n entities in train dataset. In other words, these classifiers fail in

classifying any token in test dataset as drug_n class because the corresponding model file lacks the necessary learnt patterns regarding drug_n class. The same thing happens with brand class here, but the number of this type of entity is higher in the train dataset, therefore resulting in slightly better classification performance. As be observed from this table, those classifiers with two new orthographic features (f2 and f9) achieved F-Scores less than average of F-Score values of all classifiers with orthographic features that is equal to 0.7205. It also can be observed from this table that among those seven classifiers with frequency based morphological features, f26, f24 and f21 achieved an F-Score above average of F-Scores of all 28 classifiers that is equal to 0.7308.

Table 4.2: Classification Performance of CRF Classifiers Using Single Features (Drugbank corpus)

			CLASS			
Feature No.	**Feature Name**	**Micro-Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
1.	FirstLetterIsUppercase	0.7621	0.8025	0.0000	**0.6667**	0.7731
2.	BeforeHasParentheses	0.7088	0.7805	0.0000	0.3529	0.7500
3.	HasBracket	0.7126	0.7829	0.0000	0.3768	0.7500
4.	NextHasHyphen	0.7075	0.7781	0.0000	0.3529	0.7500
5.	HasParentheses	0.7137	0.7829	0.0000	0.3768	0.7541
6.	NextHasColon	0.7013	0.7654	0.0000	0.3529	0.7603
7.	NextHasComma	0.7218	0.7953	0.0000	0.3529	0.7603
8.	NextHasSemicolon	0.7126	0.7791	0.0000	0.3768	0.7603
9.	BeforeHasBracket	0.7075	0.7768	0.0000	0.3529	0.7541
10.	NumbrInside	0.7162	0.7818	0.0000	0.3768	0.7667
11.	HasCaps	0.7546	0.8000	0.0000	0.6392	0.7667
12.	Length	0.7330	0.7988	0.0000	0.4722	0.7377
13.	allLettersofTokenAreUpperCase	0.7402	0.8012	0.0000	0.5476	0.7458
14.	HasHyphen	0.7063	0.7706	0.0000	0.3529	0.7667
15.	HasSlash	0.7102	0.7791	0.0000	0.3768	0.7500
16.	3-GramSuffix	**0.7972**	0.8539	0.0000	0.5823	0.8125
17.	2-GramSuffix	0.7959	**0.8586**	0.0000	0.5067	**0.8160**
18.	2-GramPrefix	0.7473	0.7940	0.0000	0.5135	0.7939
19.	3-GramPrefix	0.7368	0.7882	0.0000	0.4658	0.7879

20.	10PercentMostFrequent2-GramSuffixs	0.7105	0.7801	0.0000	0.3077	0.7667
21.	10PercentMostFrequent2-GramSuffixsInDrugNames	0.7382	0.8215	0.0000	0.3529	0.7480
22.	10PercentMostFrequent3-GramPrefixsInDrugNames	0.6875	0.7355	0.0000	0.4167	0.7581
23.	10PercentMostFrequent3-GramSuffixsInDrugNames	0.7266	0.8012	0.0000	0.3636	0.7460
24.	10PercentMostFrequent2-GramPrefixsInDrugNames	0.7395	0.8000	0.0000	0.5063	0.7705
25.	10PercentMostFrequent4Gram SuffixsInDrugNames	0.7124	0.7791	0.0000	0.4225	0.7377
26.	NPercentMostFrequent3Gram SuffixsInDrugNames	0.7745	0.8514	0.0000	0.4571	0.7742
27.	PhrasalCategories	0.7263	0.7955	0.0000	0.4384	0.7317
28.	PartOfSpeech	0.7631	0.7978	0.0000	0.5714	0.8154

Table 4.3 presents results using single features on SVM based classifiers that are trained on Medline DDI corpus train dataset and tested on its test data set. As can be seen in this table, classifier with feature number 23 has the best performance based on the overall F-scores and it also has the highest performance in detection of drug class. Classifier with feature number 13 has the highest performance in detection of Drug_n class. This feature indicates whether all letters of a token are uppercase or not. Classifier with feature number 5 has the highest performance in detecting group class. This feature indicates whether there is a parenthesis in a token or not. Similar to the case with CRF classifiers, the performance of all these classifiers with respect to brand class is equal to zero due to very small number of brand entities in train dataset. In other words, these classifiers fail in classifying any token in test dataset as brand class because the corresponding model file lacks the necessary learnt patterns regarding brand class. The same is true for drug_n class here but the number of this type of entity is higher in the train dataset resulting in slightly better classification performance.

Table 4.3: Classification Performance of SVM Classifiers Using Single Features (Medline corpus)

			CLASS			
Feature No.	**Feature Name**	**Micro-Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
1.	FirstLetterIsUppercase	0.3708	0.4941	0.0635	0.0000	0.4500
2.	BeforeHasParentheses	0.3597	0.4803	0.0328	0.0000	0.4516
3.	HasBracket	0.3676	0.4867	0.0342	0.0000	0.4500
4.	NextHasHyphen	0.3748	0.5000	0.0339	0.0000	0.4538
5.	HasParentheses	0.3770	0.5000	0.0342	0.0000	**0.4628**
6.	NextHasColon	0.3708	0.4924	0.0342	0.0000	0.4500
7.	NextHasComma	0.3633	0.4922	0.0339	0.0000	0.4298
8.	NextHasSemicolon	0.3683	0.4885	0.0342	0.0000	0.4500
9.	BeforeHasBracket	0.3683	0.4885	0.0342	0.0000	0.4500
10.	NumbrInside	0.3777	0.5077	0.0342	0.0000	0.4500
11.	HasCaps	0.3651	0.4961	0.0484	0.0000	0.4333
12.	Length	0.2926	0.4387	0.0169	0.0000	0.2453
13.	allLettersofTokenAreUpperCase	0.3661	0.4766	**0.0945**	0.0000	0.4370
14.	HasHyphen	0.3755	0.5077	0.0339	0.0000	0.4426
15.	HasSlash	0.3708	0.4924	0.0342	0.0000	0.4500
16.	3-GramSuffix	0.4298	0.6506	0.0336	0.0000	0.2075
17.	2-GramSuffix	0.3832	0.5479	0.0500	0.0000	0.2115
18.	2-GramPrefix	0.2724	0.3843	0.0325	0.0000	0.2963
19.	3-GramPrefix	0.3058	0.4163	0.0333	0.0000	0.3717
20.	10PercentMostFrequent2-GramSuffixs	0.2652	0.3467	0.0339	0.0000	0.3604
21.	10PercentMostFrequent2-GramSuffixsInDrugNames	0.3570	0.4656	0.0331	0.0000	0.4274
22.	10PercentMostFrequent3-GramPrefixsInDrugNames	0.3458	0.4470	0.0336	0.0000	0.4500
23.	10PercentMostFrequent3-GramSuffixsInDrugNames	**0.4885**	**0.6557**	0.0500	0.0000	0.4407
24.	10PercentMostFrequent2-GramPrefixsInDrugNames	0.2992	0.4400	0.0331	0.0000	0.2883
25.	10PercentMostFrequent4GramSuffixsInDrugNames	0.3931	0.5255	0.0339	0.0000	0.4628
26.	NPercentMostFrequent3GramSuffixsInDrugNames	0.4237	0.5903	0.0496	0.0000	0.3333
27.	PhrasalCategories	0.2984	0.4317	0.0164	0.0000	0.2909
28.	PartOfSpeech	0.3385	0.4489	0.0336	0.0000	0.3333

It can be observed from this table 4.3, that among those two classifiers with two new orthographic features (f2 and f9), the one with feature f9 achieved F-Score above average of F-Score values of all classifiers with orthographic features that is equal to 0.3645 but the one with feature f2 obtained F-Score less than this average value.

It also can be observed from this table that among those seven classifiers with frequency based morphological features, f23, f25 and f26 achieved an F-Score above average of F-Scores of all 28 classifiers that is equal to 0.3596.

Table 4.4 shows the results obtained using single features on SVM based classifiers that are trained on DDI corpus Drugbank train dataset and tested on its test dataset. As can be seen in this table, classifier with feature number 17 has the best performance based on the overall F-score. Classifier with feature number 26 has the highest performance in detection of drug class. This feature indicates the token belongs to which 10 to 90 percent most frequent 3Gram suffixes in drug names. Classifier with feature number 18 has the highest performance in detection of brand class. It provides two-gram prefix of the token. Classifier with feature number 16 has the highest performance in detecting group class. Again, the performance of all these classifiers with respect to drug_n class is equal to zero due to very small number of drug_n entities in train dataset. In other words, these classifiers fail in classifying any token in test dataset as drug_n class because the corresponding model file lacks the necessary learnt patterns regarding drug_n class. The same is true for the brand class here but the number of this type of entity is higher in the train dataset therefore there is a little bit better training, resulting in slightly better classification performance.

Table 4.4: Classification Performance of SVM Classifiers Using Single Features (Drugbank corpus)

			CLASS			
Feature No.	**Feature Name**	**Micro- Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
1.	FirstLetterIsUppercase	0.7559	0.8062	0.0000	0.6458	0.7460
2.	BeforeHasParentheses	0.7169	0.7729	0.0000	0.4000	0.7752
3.	HasBracket	0.7316	0.7988	0.0000	0.4225	0.7597
4.	NextHasHyphen	0.7422	0.8129	0.0000	0.4286	0.7597
5.	HasParentheses	0.7104	0.7755	0.0000	0.4000	0.7385
6.	NextHasColon	0.7306	0.7976	0.0000	0.4225	0.7597
7.	NextHasComma	0.7151	0.7816	0.0000	0.3768	0.7500
8.	NextHasSemicolon	0.7339	0.8000	0.0000	0.4000	0.7752
9.	BeforeHasBracket	0.7316	0.7988	0.0000	0.4225	0.7597
10.	NumebrInside	0.7399	0.8094	0.0000	0.4058	0.7692
11.	HasCaps	0.7536	0.8025	0.0000	0.6458	0.7460
12.	Length	0.7554	0.8198	0.0000	0.4800	0.7786
13.	allLettersofTokenAreUpperCase	0.7602	0.8119	0.0000	0.6237	0.7597
14.	HasHyphen	0.7353	0.8047	0.0000	0.4225	0.7597
15.	HasSlash	0.7316	0.7988	0.0000	0.4000	0.7692
16.	3-GramSuffix	0.7881	0.8436	0.0000	0.5263	**0.8244**
17.	2-GramSuffix	**0.7960**	0.8413	0.0000	0.6000	0.8217
18.	2-GramPrefix	0.7599	0.7859	0.0000	**0.6500**	0.7939
19.	3-GramPrefix	0.7442	0.7884	0.0000	0.4865	0.8060
20.	10PercentMostFrequent2-GramSuffixs	0.7269	0.7943	0.0000	0.3582	0.7692
21.	10PercentMostFrequent2-GramSuffixsInDrugNames	0.7665	0.8338	0.0000	0.4658	0.7874
22.	10PercentMostFrequent3-GramPrefixsInDrugNames	0.7266	0.7855	0.0000	0.4675	0.7634
23.	10PercentMostFrequent3-GramSuffixsInDrugNames	0.7450	0.8174	0.0000	0.4324	0.7656
24.	10PercentMostFrequent2-GramPrefixsInDrugNames	0.7623	0.8182	0.0000	0.5432	0.7910
25.	10PercentMostFrequent4Gram SuffixsInDrugNames	0.7376	0.8000	0.0000	0.4865	0.7538
26.	NPercentMostFrequent3Gram SuffixsInDrugNames	0.7840	**0.8455**	0.0000	0.5405	0.7969
27.	PhrasalCategories	0.7298	0.7912	0.0000	0.5135	0.7143
28.	PartOfSpeech	0.7604	0.7922	0.0000	0.6341	0.7874

It can be observed from this table that classifiers with new orthographic features f2 and f9 achieved F-Score values less than average of F-Score values of all classifiers with orthographic features that is equal to 0.7362. It also can be observed from this table that among those seven classifiers with frequency based morphological features, f21, f24 and f26 achieved an F-Score above average of F-Scores of all 28 classifiers that is equal to 0.7454.

It can be understood after observing these four tables, that there is always three classifiers with frequency based morphological features that their F-Scores are above average value of all 28 F-Scores and feature f26 is always one of these three features. This observation emphasizes on the importance of this type of feature in this domain and more specifically effectiveness of feature f26. The other subject that should be investigated in these four tables is the effectiveness of those two new orthographic features (f2 and f9). As can be observed in all these four tables, in three out of four cases, both of these features are less than the average of F-Scores of orthographic features and in that one remaining case, just one of these two features are above that average value. This indicates that these two new orthographic features do not have a strong and obvious positive effect on classification of drug-name entities on this corpus. We decided to combine these features with other features and investigate the positive or negative effects of them on the other ones.

4.1.2 Entity Recognition Using Single Features

Entity recognition task involves the recognition of drug names regardless of their class. In other words, the classifier simply classifies the entities as “Drug Entities” and “Non-Drug Entities”.

Table 4.5 shows performances of CRF and SVM based classifiers on both Medline and Drugbank corpora for entity recognition. It can be seen that classifier with feature number 17 that is 2-gram suffix feature, performs the best for 3 out of 4 cases.

Furthermore, in general the SVM classifiers perform better than CRF classifiers. Also average recognition performance on the Drugbank corpus is much better compared to that of the Medline corpus.

Table 4.5: NER Performance of Classifiers Using Single Features

		Micro- Average F-score			
		CRF		**SVM**	
Feature No.	**Feature Name**	Medline	Drugbank	Medline	Drugbank
1.	FirstLetterIsUppercase	0.2578	0.7993	0.4142	0.8101
2.	BeforeHasParentheses	0.2472	0.7816	0.4032	0.8088
3.	HasBracket	0.2432	0.7816	0.4032	0.8088
4.	NextHasHyphen	0.2461	0.7839	0.4103	0.8154
5.	HasParentheses	0.2517	0.7824	0.4048	0.8051
6.	NextHasColon	0.2472	0.7707	0.4063	0.8081
7.	NextHasComma	0.2353	0.8045	0.3912	0.8203
8.	NextHasSemicolon	0.2449	0.7816	0.404	0.8147
9.	BeforeHasBracket	0.2393	0.7763	0.404	0.8088
10.	NumbrInside	0.2511	0.7924	0.4095	0.8205
11.	HasCaps	0.2611	0.7955	0.4048	0.8043
12.	Length	0.3108	0.7919	0.3567	0.8094
13.	allLettersofTokenAreUpperCase	0.2549	0.7963	0.4134	0.8206
14.	HasHyphen	0.2427	0.7793	0.4111	0.8125
15.	HasSlash	0.2477	0.7793	0.4063	0.8088
16.	3-GramSuffix	0.4755	0.8256	0.5435	0.8231
17.	2-GramSuffix	**0.5582**	**0.8571**	0.558	**0.8600**
18.	2-GramPrefix	0.2979	0.7949	0.3333	0.8244
19.	3-GramPrefix	0.2685	0.7949	0.3636	0.8193
20.	10PercentMostFrequent2-GramSuffixs	0.1667	0.8008	0.2913	0.8354
21.	10PercentMostFrequent2-GramSuffixsInDrugNames	0.4302	0.8218	0.5173	0.8378
22.	10PercentMostFrequent3-GramPrefixsInDrugNames	0.2839	0.7383	0.4204	0.7963
23.	10PercentMostFrequent3-GramSuffixsInDrugNames	0.4982	0.7963	**0.5967**	0.8137
24.	10PercentMostFrequent2-GramPrefixsInDrugNames	0.255	0.7778	0.3443	0.8131
25.	10PercentMostFrequent4Gram SuffixsInDrugNames	0.2727	0.7505	0.4239	0.7817
26.	NPercentMostFrequent3Gram SuffixsInDrugNames	0.5295	0.8218	0.5729	0.8167
27.	PhrasalCategories	0.3207	0.8301	0.407	0.8316
28.	PartOfSpeech	0.4421	0.8293	0.4513	0.8229

4.2 Classification Performance of Classifiers Using All Features

The performance of classifiers using combination of all features are investigated and the results are shown in Table 4.6. These classifiers have an important role in comparing and evaluating the other classifiers that are made by combining different features using FS and BS methods that we discuss in the following sections.

As can be seen in table 4.6, the classification performance of SVM classifiers is better than CRF classifiers. On the other hand, for both CRF and SVM classifiers, we can see that the F-Score of these classifiers with all features combined, are higher than any single feature classifier individually.

Table 4.6: Classification Performance of SVM and CRF Classifiers Using All Features on Medline and Drugbank Corpora

Classifier											
CRF						SVM					
Corpus						Corpus					
Medline			Drugbank			Medline			Drugbank		
R	P	F	R	P	F	R	P	F	R	P	F
0.4136	0.6371	0.5016	0.8454	0.8712	0.8581	0.4607	0.6132	0.5262	0.8586	0.8847	0.8715

As an additional experiment, we have investigated the effect of adding the predicted output of the SVM classifier as an additional feature in training CRF classifiers using combination of all features. As can be seen in Table 4.7 for Medline dataset, this experiment results in one percent improvement in terms of F-Score and for Drugbank dataset, we can see almost two percent increase in terms of F-Score.

Table 4.7: Comparison of Classification Performances of CRF Classifier with All Features Combined and the Classifier with SVM Output Feature

Corpus											
Medline						**Drugbank**					
classifiers						**classifiers**					
All Features			All + SVM output			All Features			All + SVM output		
R	P	F	R	P	F	R	P	F	R	P	F
0.4136	0.6371	0.5016	0.4372	0.6231	**0.5138**	0.8454	0.8712	0.8581	0.8618	0.8822	**0.8719**

4.3 Feature Ensembles Based on Feature Types

Feature grouping is a common method for selecting feature subsets as well as understanding the usefulness of a particular set of features according to their types. The 28 features used in this study are grouped into three groups as orthographic (G1), morphological (G2) and lexical features (G3). We investigate the final classification performance of each one of the three groups and their combinations on Medline and Drugbank corpora in following tables.

Table 4.8: Classification Performance of CRF and SVM Classifiers from Different Feature Groups on Medline Data

Group	**Performance Using CRF (R / P / F) train/test**	**Performance Using SVM (R / P / F) train/test**
G1 (orthographic features)	(0.1963 / 0.5245 / 0.2857)	(0.2749 / 0.6250 / 0.3818)
G2 (morphological features)	(0.3874 / 0.6352 / 0.4813)	(0.4319 / 0.6371 / 0.5148)
G3 (lexical features)	(0.2173 / 0.4213 / 0.2867)	(0.2356 / 0.4286 / 0.3041)
G1+G2	(0.3979 / 0.6154 / 0.4833)	(0.4764 / 0.6276 / **0.5417**)
G1+G3	(0.3010 / 0.4228 / 0.3517)	(0.3194 / 0.4766 / 0.3824)
G2+G3	(0.3770 / 0.6344 / 0.4729)	(0.4372 / 0.6448 / 0.5211)
G1+G2+G3	(0.4136/0.6371/**0.5016**)	(0.4607 / 0.6132 / 0.5262)

Table 4.9: Classification Performance of CRF and SVM Classifiers from Different Feature Groups on Drugbank Data

Group	Performance Using CRF (R / P / F) Train/Test	Performance Using SVM (R / P / F) Train/Test
G1 (orthographic features)	(0.7500 / 0.8476 / 0.7958)	(0.7763 / 0.8252 / 0.8000)
G2 (morphological features)	(0.8257 / 0.8685 / 0.8465)	(0.8191 / 0.8527 / 0.8356)
G3 (lexical features)	(0.7303 / 0.8222 / 0.7735)	(0.7237 / 0.7914 / 0.7560)
G1+G2	(0.8322 / 0.8724 / 0.8519)	(0.8651 / 0.8855 / **0.8752**)
G1+G3	(0.8125 / 0.8517 / 0.8316)	(0.8322 / 0.8405 / 0.8364)
G2+G3	(0.8257 / 0.8685 / 0.8465)	(0.8224 / 0.8562 / 0.8389)
G1+G2+G3	(0.8454/0.8712/**0.8581**)	(0.8586/0.8847/0.8715)

From table 4.8 and 4.9, it can be observed that when CRF classifiers ae used, the best performance is obtained when all features are combined. On the other hand, the combination of orthographic and morphological features (G1+G2) achieves the best results when SVM classifiers are used.

4.4 Wrapper Based FS and BS Feature Selection

Feature selection and combination are performed to exclude those features that have a negative effect on overall performance of the NEC system and gain the optimized performance by making the best well fitted combination of features. In the following tables, we show the effects of applying FS and BS feature selection algorithms on SVM and CRF based classifiers on both Drugbank and Medline datasets. Table 4.10 shows that almost the same number of features are selected by applying FS and BS methods on Medline data. Nineteen features are selected using BS method and 17 features are selected using FS method. There is a balance in number of orthographic and morphological features that are selected. In the FS method 8 morphological and 8 orthographic features are selected whereas in the BS method, 9 morphological and 10 orthographic features are selected. There are ten common features among both FS and

BS methods namely f10, f12, f13, f14, f17, f18, f19, f23, f26 and f4. Five of these features belong to feature group number one and five other ones belong to feature group number two. Again, this points to the importance of features in these two groups and their combination. There is no common feature from third group, lexical features. We can observe from this table, regarding those two new orthographic features that we discussed in chapter 3, section 3.3, (f2 and f9), by using BS method, f2 is selected. By using FS method, none of these two features are selected.

Table 4.10: Feature Ensembles Obtained Using FS and BS Methods for CRF Classifiers on Medline Corpus

Method	Feature Set	Number Of Features selected	Number Of Lexical Features	Number Of Orthographical Features	Number Of Morphological Features	F- Score
BS	f20, f23, f24, f25, f26, f27, f19, f18, f17, f15, f14, f13, f12, f10, f8, f7, f4, f3, f2	19	1	10	8	0.5127
FS	f4, f5, f6, f10, f11, f12, f13, f14, f16, f17, f18, f19, f21, f22, f23, f26, f28	17	1	8	8	0.4869

According to the results that are represented in table 4.11, 18 features are selected using BS method among them 10 features belong to Orthographical group, 7 belong to Morphological group and one feature belongs to lexical group of features. Thirteen features are selected using FS method among them 6 features belong to orthographic group and 6 features also belong to morphological group. One feature belongs to lexical group. There are eleven common features among both FS and BS methods that are f10, f11, f12, f13, f16, f19, f22, f24, f25, f26, f28. Four of these common ones belong to orthographic features; six of them are morphological features and feature number 28 that is phrasal categories feature, belong to third group. From those two new orthographic features (f2 and f9), by using FS method, f9 is selected. By using BS method, none of these two features are selected.

Table 4.11: Feature Ensembles Obtained Using FS and BS Methods for CRF Classifiers on Drugbank Corpus

Method	Feature Set	Number Of Features selected	Lexical Features	Orthographical Features	Morphological Features	F- Score
BS	f1, f5, f7, f8, f10, f11, f12, f13, f14, f15, f16, f19, f20, f22, f24, f25, f26, f28	18	1	10	7	0.8514
FS	f6, f9, f10, f11, f12, f13, f16, f19, f22, f24, f25, f26, f28	13	1	6	6	0.8508

Table 4.12 represents common selected features in FS and BS selection algorithms on both Medline and Drugbank datasets. Among common features between FS and BS methods on Medline data, there is no feature from lexical group of features and five of them belong to orthographic group of features and the other five features belong to morphological group of features. Among common features on Drugbank data, the POS feature belongs to lexical group of features and four features belong to orthographic features and six features belong to morphological group. f10, f12, f13, f19 and f26 are five common features among common features between FS and BS methods on both Drugbank and Medline datasets. It can be deduced that these features may constitute a

good feature ensemble when experiments are carried out on the combination of these datasets.

It should be noted that wrapper based feature selection algorithms are applied using CRF classifiers due to time limitations regarding the conclusion of this thesis since SVM classifiers take about 5 more times training time on the average. This part of the work should be done using SVM classifiers in order to obtain more comprehensive results.

Table 4.12: Common Features in Feature Ensembles Obtained From FS and BS Methods for CRF Classifiers

Methods	Corpus	Common features
FS,BS	Medline	f4, f10, f12, f13, f14, f17, f18, f19, f23, f26
FS,BS	Drugbank	f10, f11, f12, f13, f16, f19, f22, f24, f25, f26, f28
FS,BS	Medline and Drugbank	f10, f12, f13, f19, f26

Table 4.13 presents and compares the classification performance of classifiers which use feature ensembles obtained using FS and BS methods and compares them to the performance of the classifier which uses all 28 features and to the performance of the single best classifier on Medline dataset. It can be observed that BS algorithm works better than two others in terms of overall F-Score and drug class. For group and drug_n class, all features combined classifier works the best. As mentioned before because of small number of entities from brand class in Medline training dataset, all these CRF classifiers were unable to classify any token as brand correctly.

Table 4.13: Comparison of Classification Performance of Final Combination of Features Selected for CRF Classifiers using FS and BS Methods (Medline Corpus)

		CLASS			
Method	**Micro- Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
FS	0.4869	0.6667	0.1194	0.0000	0.3826
BS	**0.5127**	**0.6739**	0.1805	0.0000	0.4160
Single Best	0.3965	0.5889	0.0331	0.0000	0.2000
All Features	0.5016	0.6573	**0.1818**	0.0000	**0.4480**

Similarly, Table 4.14 presents and compares the classification performance of classifiers which use feature ensembles obtained using FS and BS methods. At the same time, it compares those performance results with the performance of the classifier, which uses all 28 features, and the performance of the single best classifier on Drugbank dataset. It can be seen that the classifier which uses the combination of all features perform the best in terms of overall F-Score and F-Scores of drug and group classes. The FS algorithm performs better than the other two ensembles only for the brand class.

Table 4.14: Comparison of Classification Performance of Final Combination of Features Selected for CRF Classifiers Using FS and BS Methods (Drugbank Corpus)

		CLASS			
Method	**Micro- Average F-score**	**Drug**	**Drug_n**	**Brand**	**Group**
FS	0.8508	0.8825	0.0000	**0.8713**	0.7910
BS	0.8514	0.8889	0.0000	0.8400	0.8000
Single Best	0.7972	0.8539	0.0000	0.5823	0.8125
All Features	**0.8581**	**0.8933**	0.0000	0.8515	**0.8088**

In table 4.15 a general comparison between two CRF and SVM classifiers has been made. We made a new dataset consists of both Drugbank DDI corpus and Medline DDI corpus and trained and tested both CRF and SVM classifiers with feature ensemble made of combination of morphological and orthographic features. The comparison shows us that SVM classifier works noticeably better than CRF classifier. It should be mentioned here that in most of the experiments that has been performed in this work, SVM classifiers show a better performance in terms of F-Score.

Table 4.15: Comparison of Classification Performance of CRF and SVM Classifiers with Combination of Feature ensembles of Group One and Group Two on the Complete Corpus (Medline + Drugbank)

feature ensemble	**Group one and two combined**	
classifier	CRF	SVM
Micro- Average F-score	0.6783	**0.7243**

4.5 Wrapper Based FS and BS Classifier Selection and Combination

In this section, we present the results obtained using wrapper based classifier selection algorithms stated in Chapter 3 and compare the results with previously presented results.

Table 4.16 shows the effectiveness of using three different majority-voting algorithms explained in Chapter 3 on the Medline and Drugbank corpora for CRF and SVM classifiers. It can be observed that for both FS and BS selection methods, in both CRF based and SVM based classifiers, Ranked-Weighted majority voting algorithm by far, leads to the highest performance among all three different voting algorithms.

Table 4.16: Classification Performance Comparison of Classifier Ensembles Using CRF Classifiers for Three Different Voting Methods (Medline and Drugbank Corpora)

Selection Method			**FS**			**BS**		
Voting Method			**Simple**	**Weighted**	**Ranked-Weighted**	**Simple**	**Weighted**	**Ranked-Weighted**
Micro - Average F-score	**Corpus**	**Medline**	0.3811	0.3230	**0.4020**	0.2613	0.3852	**0.4279**
		Drugbank	0.4237	0.3877	**0.4795**	0.3625	0.3894	**0.4911**

Based on the results that are represented in Table 4.16, we decided to use ranked-weighted majority voting algorithm that is described in section 3.5.3 for the remainder of the classifier selection algorithms. The K value that is used during combination in this voting algorithm, is chosen between 1 and 1.98 throughout experiments.

Tables 4.17, 4.18 and 4.19 show the classifier ensembles selected for each method when they are formed from a pool of only CRF classifiers, only SVM classifiers and both SVM and CRF classifiers respectively. These tables also represent the classification performance of those classifier ensembles. In these tables, “ec” stands for a CRF classifier and “es” stands for a SVM classifier. For example “ec17” corresponds to CRF classifier with single feature 17 or “esG1G2” is a SVM classifier with feature ensemble made of combination of feature groups one and two. These classifier ensembles are selected by using FS and BS selection methods on both Medline and Drugbank corpora.

Table 4.17: CRF Based Classifier Ensembles Formed Using FS and BS Methods (Medline and Drugbank Corpora)

Method	**FS**		**BS**	
Corpus	**Medline**	**Drugbank**	**Medline**	**Drugbank**
Number Of Classifiers	7	1	27	25
Classifier Ensemble	ecG1G2G3, ec6, ec9, ec27, ec16, ecG2G3, ecG1G2	ecG1G2G3	ecG1G2G3, ecG1G2, ecG1G3, ecG2G3, ec26, ec17, ec16, ec28, ec25, ec27, ec22, ec12, ec20, ec18, ec11, ec2, ec5, ec1, ec4, ec13, ec10, ec14, ec24, ec19, ec15, ec8, ec9	ec1, ec2, ec3, ec4, ec5, ec6, ec7, ec8, ec9, ec10 ec11, ec12, ec13, ec14, ec15, ec16, ec18, ec19, ecG1, ecG2, ecG3, ecG1G2, ecG1G3, ecG2G3, ecG1G2G3
F-score	**0.5176**	**0.8581**	0.5007	0.8558

Table 4.18: Classifier Ensembles Formed Using FS and BS Methods on SVM Classifiers (Medline and Drugbank Corpora)

Method	**FS**		**BS**	
Corpus	**Medline**	**Drugbank**	**Medline**	**Drugbank**
Number Of Classifiers	9	1	30	34
Classifier Ensemble	esG1G2G3, es8, es1, es15, es12, es27, es25, es16, esG1G2	esG1G2	esG1, esG1G2, es26, es23, es17, es16, es21, es25, es27, es22, es12, es5, es2, es11, es4, es10, es20, es13, es14, es15, es18, es1, es6, es3, es9, es8, es7, es24, es19, esG1G2G3	es1, es2, es3, es4, es5, es6, es7, es8, es9, es10, es11, es12, es13, es14, es15, es16, es17, es18, es19, ce20, es21, es22, es23, es24, es25, es26, es27, es28, esG1, esG2, esG3, esG1G2, esG1G3, esG1G2G3
F-score	**0.5501**	**0.8752**	0.5393	0.8522

Table 4.19: Classifier ensembles Formed Using FS and BS Methods on both CRF and SVM Classifiers (Medline and Drugbank Corpora)

Method	**FS**		**BS**	
Corpus	**Medline**	**Drugbank**	**Medline**	**Drugbank**
Number Of Classifiers	10	1	57	68
Classifier Ensemble	esG1G2G3, ec19, ec20, es8, es4, es11, esG1G2, esG1, ecG2G3, ecG1	esG1G2	ec1, ec2, ec4, ec5, ec6, ec7, ec8, ec9, ec11, ec12, ec13, ec14, ec15, ec16, ec18, ec19, ec20, ec21, ec22, ec24, ec25, ec26, ec27, ecG2, ecG3, ecG1G2, ecG1G3, ecG2G3, ecG1G2G3, es1, es2, es3, es4, es5, es6, es7, es8, es9, es10, es11, es12, es13, es14, es15, es17, es18, es19, es20, es21, es22, es24, es25, es26, es27, esG1, esG2, esG2G3	ec1, ec2, ec3, ec4, ec5, ec6, ec7, ec8, ec9, ec10, ec11, ec12, ec13, ec14, ec15, ec16, ec17, ec18, ec19, ec20, ec21, ec22, ec23, ec24, ec25, ec26, ec27, ecG1, ecG2, ecG3, ecG1G2, ecG2G3, ecG1G2G3, es1, es2, es3, es4, es5, es6, es7, es8, es9, es10, es11, es12, es13, es14, es15, es16, es17, es18, es19, es20, es21, es22, es23, es24, es25, es26, es27, es28, esG1, esG2, esG3, esG1G2, esG1G3, esG2G3, esG1G2G3
F-score	**0.5538**	**0.8752**	0.5496	0.8621

As can be observed from these three tables, by applying FS method on pools made of CRF classifiers, SVM classifiers and both, on Medline dataset, higher classification performances are achieved rather than BS method. According to these three tables, by applying FS method on pools made of CRF classifiers, SVM classifiers and both, on Drugbank dataset, no other classifier than the single best classifier is selected mainly because of the fact that majority of classifiers trained on Drugbank dataset are strong classifiers.

In tables 4.20, 4.21 and 4.22, common classifiers that are selected among both FS and BS methods from CRF, SVM and both of them respectively, are presented. It can be observed that in all of these common classifiers, there is at least one classifier with feature grouping origin (G1, G2, G3, etc.). Results of applying each of these two classification algorithms, indicate that those classifiers with feature grouping origin, has a positive influence on other classifiers.

Table 4.20: Common Classifiers Obtained Using FS and BS Methods for Classifier Subset Selection from CRF Classifiers (Medline and Drugbank Corpora)

Corpus	Common classifiers
Medline	ecG1G2G3, ec27, ec16, ecG2G3, ecG1G2
Drugbank	ecG1G2G3

Table 4.21: Common Classifiers Obtained Using FS and BS Methods for Classifier Subset Selection from SVM Classifiers (Medline and Drugbank Corpora)

Corpus	Common classifiers
Medline	es8, es1, es15, es12, es27, es25, es16, esG1G2
Drugbank	esG1G2

Table 4.22: Common Classifiers Obtained Using FS and BS Methods for Classifier Subset Selection from both CRF and SVM Classifiers (Medline and Drugbank Corpora)

Corpus	Common classifiers
Medline	esG1, ecG2G3, ec19, ec20, es8, es4, es11
Drugbank	esG1G2

Table 4.23 compares the performance of classifier ensembles obtained using the discussed methods from a pool made of CRF classifiers. It can be seen that the best result can be obtained using the FS method using Medline data however, the single best classifier that uses the set of all features performs better than all other classifier ensembles in the case of Drugbank dataset.

Table 4.23: Comparison of Classification Performance of Best Single CRF Classifier and the Final Ensemble of Selected CRF Classifiers (Medline and Drugbank Corpora)

Corpus	**Medline**			**Drugbank**		
Type of classifier ensemble	single best (All Features)	all classifiers combined	final ensemble of selected classifiers (FS)	single best (All Features)	all classifiers combined	final ensemble of selected classifiers (BS)
Micro- Average F-score	0.5016	0.4835	**0.5176**	**0.8581**	0.8396	0.8558

It can be observed in Table 4.24 that similar results are obtained when the pool of classifiers used in forming the ensembles are SVM classifiers.

Table 4.24: Comparison of Classification Performance of Best Single SVM Classifier and the Final Ensemble of Selected SVM Classifiers (Medline and Drugbank Corpora)

Corpus	**Medline**			**Drugbank**		
Type of classifier ensemble	single best (features of group 1 and 2 combined)	all classifiers combined	final ensemble of selected classifiers (FS)	single best (features of group 1 and 2 combined)	all classifiers combined	final ensemble of selected classifiers (BS)
Micro- Average F-score	0.5417	0.5103	**0.5501**	**0.8752**	0.8504	0.8522

Table 4.25 compares the final performance of classifier ensembles that are formed after classifier selection and combination, using FS and BS methods on three different pools of classifiers that are all made previously based on Medline and Drugbank datasets. First one is made from just CRF classifiers, the other one is made from SVM classifiers and the last one is made from both of them together. As can be seen, final performances of those classifiers that are made from a pool consists of both CRF and SVM classifiers using both FS and BS methods, are higher than both of them

individually. It means we got an improvement in performance by increasing the diversity of classifiers in the pool.

Table 4.25: Comparison of Different Classifier Ensembles on Medline and Drugbank data

		Method			
		FS		**BS**	
	Corpus	MEDLINE	Drugbank	MEDLINE	Drugbank
	Classifier Origin				
Micro- Average F-score	CRF (35 classifiers)	0.5176	0.8581	0.5007	0.8558
	SVM (35 classifiers)	0.5501	0.8752 (single best classifier)	0.5393	0.8522
	BOTH (70 classifiers)	**0.5538**	0.8752 (single best classifier)	**0.5496**	**0.8621**

Table 4.26, represents the general comparison between wrapper based feature selection and classifier selection experiments on CRF based classifiers created on Medline data and for each of them compares FS and BS methods with each other. Table 4.27 does the same representation but for classifiers made on Drugbank dataset. By observing the results on table 4.26, we can conclude that for Medline data, classifier selection plays a more effective role in improving the performance of NEC system rather than the feature selection alone. FS classifier selection on Medline data (here better than BS), achieves the F-score of 0.5176 and exceeds the final F-Score that BS feature selection alone (here better than FS), could reach that is equal to 0.5127. This Table also indicates that feature selection alone, has improved the performance of the NEC

system considering the fact that best single classifier's performance before performing feature selection belonged to the classifier with combination of all features that has the F-score of 0.5016 and feature selection using BS method increase this score up to 0.5127. Results in table 4.27 indicates that none of two feature selection and classifier selection approaches could have exceeded the performance of single best classifier that corresponds to all features merged with F-Score of 0.8581. However, the same results show us like on Medline data, applying classifier selection approach on classifiers from Drugbank dataset, leads into better final performance than feature selection alone.

Table 4.26: Overall Comparison between Feature Selection and Classifier Selection Approaches on Medline Dataset

	ID	**F-Score**
Single Best feature	f23 (10Percent Most Frequent 3-GramSuffixs In Drug Names)	0.3965
Single Best Classifier	ecG1G2G3	0.5016
Selection Method	FS	BS
	F-Score	F-Score
Final ensemble of features After Feature Selection	0.4869	0.5127
Final ensemble of classifiers after Classifier Selection	**0.5176**	0.5007

Table 4.27: Overall Comparison between Feature Selection and Classifier Selection Approaches on Drugbank Dataset

	ID	**F-Score**
Single Best feature	f16 (3-GramSuffixFeature)	0.7972
Single Best Classifier	ecG1G2G3	0.8581
Selection Method	FS	BS
	F-Score	F-Score
Final ensemble of features After Feature Selection	0.8508	0.8514
Final ensemble of classifiers after Classifier Selection	**0.8581** (Single Best Classifier)	0.8558

Chapter 5

CONCLUSION

In this thesis, the drug name entity recognition problem is investigated using FS and BS wrapper based feature selection and classifier selection algorithms. CRF and SVM machine learning methods are used for these tasks.

Medline and Drugbank corpora are used for this work for both training and testing. Three groups of features were extracted; orthographic, morphological and lexical. Wrapper based feature subset selection is applied on CRF based classifiers to obtain the best ensemble of features. In order to improve the performance of the NERC system, on both CRF based and SVM based classifiers, wrapper based classifier selection is applied to find the optimal ensemble of classifiers.

According to our results, 2gram suffix feature classifier is one of the best single feature classifier among CRF and SVM based classifiers in this domain. Our results show that by combining orthographic and morphological features, SVM classifiers obtain the best performance. For CRF classifiers, the best combination of features before applying feature selection methods belongs to all features. We concluded based on our results, SVM classifiers usually work better than CRF classifiers. Based on our results, feature selection is more effective in terms of increasing the classification performance on Medline data rather than Drugbank data and furthermore BS feature selection obtains better results than FS. We conclude that this difference is mainly because all CRF classifiers trained on Drugbank data are relatively strong classifiers (all of them with F-Scores more than 0.70).

In classifier selection experiments, our results indicate that combination of ranked majority voting and weighted majority voting methods show better performance than simple and weighted voting methods individually. According to the results, applying wrapper based FS and BS classifier selection on a pool of classifiers which consists of both CRF and SVM based classifiers, shows better results than applying classifier selection to CRF or SVM classifiers only as expected.

Some tasks which can be considered as future work are:

1. Applying feature selection on SVM classifiers.
2. Investigating the effects of applying another search and selection algorithm like random forest.
3. Investigating the performance of a dictionary-based classification system on same data that is used in this work.

REFERENCES

[1] Bedmar, S. I., Martínez, P., & Zazo, H. M. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). (2013). In *Proceedings of the 7th International Workshop on Semantic Evaluation.* Spain.

[2] Miller, W. T. (2004). *Data and Text Mining: A Business Application Approach*. Prentice-Hall Publication, USA.

[3] Leaman, R., & Gonzalez, G. (2008). Banner: An executable survey of advances in biomedical named entity recognition. *Pacific Symposium on Biocomputing,* Arizona State University, USA. 13:652-663.

[4] Settles, B. (2005). ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Journal of Bioinformatics.* 21(14): 3191–3192.

[5] Rocktaschel, T., Huber, T., Weidlich, M., & Leser, U. (2013). The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, USA, June, pages: 356–363.

[6] Boldyrev, A. (2013). Dictionary-Based Named Entity Recognition. Master's Thesis, Universitat des Saarlandes, Germany.

[7] Rubrichi, S., Gabetta, M., Bellazzi, R., Larizza, C., & Quaglini, S. (2011). Drug-Drug Interactions Discovery Based on CRFs, SVMs and Rule-Based Methods. Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDI Extraction 2011), Huelva, Spain, September, pages: 67-74.

[8] Mata, J., Santano, R., Blanco, D., Lucero, M., & Maña, J. M. (2011). A Machine Learning Approach to Extract Drug – Drug Interactions in an Unbalanced Dataset. *Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction (DDI Extraction)*, Huelva, Spain, September, pages: 67-74.

[9] Piliouras, D., Korkontzelos, I., Dowsey, A., & Ananiadou S. (2013). Dealing with Data Sparsity in Drug Named Entity Recognition. *Healthcare Informatics (ICHI), 2013 IEEE International Conference*, Philadelphia, USA, September, Pages 14 - 21.

[10] Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Leaming,* Kluwer Academic Publishers, 20: 273-297.

[11] Lafferty, J., McCallum, A., & ePreira, C.N. F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the 18th International Conference on Machine Learning*, Montreal, Canada, June, pages: 282-289.

[12] http://www.nlm.nih.gov/pubs/factsheets/Medline.html.

[13] http://www.Drugbank.ca.

[14] Shatkay, H., & Feldman, R. (2003). Mining the Biomedical Literature in the Genomic Era: An Overview. *Journal of Computational Biology.* 10(6): 821-855.

[15] Ananiadou, S., Thompson, P., Nawaz, R., McNaught, J., & Kell, B. D. (2014). Event-based text mining for biology and functional genomics. *Briefings in Functional Genomics,* doi:10.1093/bfgp/elu015

[16] Rinaldi, F., Clematide, S., Marques, H., Ellendorff, T., Romacker, M., & Esteban, R. R. (2013). OntoGene web services for biomedical text mining. *BMC Bioinformatics*, 15(14):S6

[17] Krallinger, M. (2013). Trends in biomedical text mining. *MAVIR Conference*, Spanish National Cancer Research Centre, (CNIO), Madrid, Spain, November 18th

[18] Yeh, A., Hirschman, L., & Morgan, A. (2003). Background and Overview for KDD Cup 2002 Task 1: Information Extraction from Biomedical Articles. *SIGKDD Explor. Newsl.* 4(2): 87-89.

[19] http://ir.ohsu.edu/genomics.

[20] Hersh, W., & Voorhees, E. (2008). TREC genomics special issue overview. *Information Retrieval*, 12(1): 1.

[21] Hersh, W., Bhupatiraju, & T. R. (2003). TREC Genomics Track Overview. *Hersh03trecgenomics*, pages: 14-23.

[22] http://www.biocreative.org/.

[23] Collier, N., Ruch, P., & Nazarenko, A. (2004). JNLPBA: Proceedings of the International Joint Workshop on Natural Language, *Processing in Biomedicine and its Applications*. COLING Post-Conference Workshop, Geneva, Switzerland, August.

[24] http://www.biocreative.org/events/biocreative-i/biocreative-i/.

[25] http://www.biocreative.org/events/biocreative-i/workshop-i/.

[26] http://www.biocreative.org/events/biocreative-ii/.

[27] http://www.biocreative.org/events/biocreative-iii.

[28] http://www.biocreative.org/events/biocreative-iv.

[29] http://www.biocreative.org/tasks.

[30] http://genome.jouy.inra.fr/texte/LLLchallenge.

[31] http://compbio.ucdenver.edu/BioNLP2013/index.shtml.

[32] http://2013.bionlp-st.org/Intro.

[33] Nédellec, C., Bossy, R., Kim, D. J., Kim, J. J., Ohta, T., Pyysalo, S., & Zweigenbaum, P. (2013). Overview of BioNLP Shared Task 2013. *Proceedings of the BioNLP Shared Task Workshop*, Sofia, Bulgaria, August, pages: 1–7.

[34] http://predictioncenter.org.

[35] Kryshtafovych, A., Monastyrsky, B., & Fidelis, K. (2013). CASP prediction center infrastructure and evaluation measures in CASP10 and CASP ROLL. *Proteins,* 82: 7–13. doi: 10.1002/prot.24399

[36] http://psb.stanford.edu/welcome.html.

[37] http://psb.stanford.edu/cfp.html.

[38] https://sites.google.com/site/mantraeu/clef-er-challenge.

[39] Schuhmann, R. D., Clematide, S., Rinaldi, F., Kafkas, S., Mulligen, M. E., Bui, C., Hellrich, J., Lewin, I., Milward, D., Poprat, M., Yepes, J. A., Hahn, U., & Kors, A. J. (2013). *Multilingual semantic resources and parallel corpora in the biomedical domain: the CLEF-ER Challenge*. CLEF Evaluation Labs and Workshop Online Working Notes, Valencia, Spain.

[40] Batzoglou, S., & Schwartz, R. (2014). ISMB 2014 PROCEEDINGS PAPERS COMMITTEE. *Bioinformatics* 30(12): i3-i8

[41] http://www.ebi.ac.uk/Rebholz-srv/CALBC.

[42] http://www.ebi.ac.uk/Rebholz-srv/CALBC/challenge_tasks.html.

[43] https://www.i2b2.org/NLP/HeartDisease.

[44] Bedmar, S. I., Martinez, P., & Cisneros, S. D. (2011). The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. *Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction*, Huelva, Spain, September, pages: 1-9.

[45] Bedmar, S. I., Martinez, P., & Zazo, H. M. (2013). SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: *Proceedings of the Seventh International Workshop on Semantic Evaluation*, Atlanta, Georgia, USA, June, pages : 341–350.

[46] http://www.bioasq.org.

[47] Cohen, M. A., & Hersh, R. W. (2004). A survey of current work in biomedical text mining. *Briefings in Bioinformatics*. 6(1): 57–71.

[48] Cisneros, S. D., & Gali, A. F. (2013). An Ontology-based named entity recognition system for biomedical texts. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, USA, June, pages: 622–627.

[49] http://www.iscb.org/cms_addon/conferences/cshals2008.

[50] Bjorne, J., Kaewphan, S., & Salakoski, T. (2013). UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, USA, June, pages: 651–659.

[51] http://www.iscb.org/cms_addon/conferences/cshals2008/discussion.php.

[52] Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, J. L., Eilbeck, K., Ireland, A., Mungall, J. C., Leontis, N., Serra, R. P., Ruttenberg, A., Sansone, A. S., Scheuermann, H. R., Shah, N., Whetzel, L. P., & Lewis, S. (2007). The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology,* 25(11): 1251-1255.

[53] http://www.ncbi.nlm.nih.gov/pubmed.

[54] http://pubchem.ncbi.nlm.nih.gov/help.html.

[55] http://www.ncbi.nlm.nih.gov/pcsubstance.

[56] http://www.ncbi.nlm.nih.gov/pccompound.

[57] http://www.ncbi.nlm.nih.gov/pcassay.

[58] http://www.ebi.ac.uk/chebi/.

[59] http://www.ncbi.nlm.nih.gov/mesh.

[60] Zazo, H. M., Bedmar, S. I., & Martínez, P. (2013). Annotation Issues in Pharmacological Texts. *Procedia - Social and Behavioral Sciences,* 95: 211-219.

[61] Cohen, B. K., Fox, L., Ogren, V. P., & Hunter, L. (2005). Empirical data on corpus design and usage in biomedical natural language processing. *AMIA Annual Symposium Proceedings 2005*, Pages: 156–160.

[62] Ohta, T., Tateisi, Y., & Kim, D. J. (2002). The GENIA Corpus: an Annotated Research Abstract Corpus in Molecular Biology Domain. In *Proceedings of the second international conference on Human Language Technology Research*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pages: 82-86.

[63] Tanabe, L., Xie, N., Thom, H. L., Matten, W., & Wilbur, W. J. (2005). GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics,* 6 (Suppl 1):S3.

[64] http://nlp.shef.ac.uk/clef.

[65] Roberts, A., Gaizauskas, R., Hepple, M., Demetriou, G., Guo, Y., Roberts, I., & Setzer, A. (2009).Building a semantically annotated corpus of clinical texts. *Journal of Biomedical Informatics,* 42(5): 950-966.

[66] http://biotext.berkeley.edu.

[67] Rosario, B., & Hearst, A. M., (2004). Classifying Semantic Relations in Bioscience Texts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL '04)*. Stroudsburg, PA, USA, Article 430.

[68] Zazo, H. M., Bedmar, S. I., Martínez, P., & Declerck, T. (2013). The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics,* 46(5): 914-920.

[69] Gurulingappa, H., Rajput, M. A., Roberts, A., Fluck, J., Apitius, H. M., & Toldo, L. (2012). Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics,* 45(5): 885-892.

[70] Mulligen, V. E., Reglat, F. A., Gurwitz, D., Molokhia, M., Nieto, A., Trifiro, G., Kors, A. J., & Furlong, I. L. (2012). The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics,* 45(5): 879-884.

[71] Bea, A., Claire, G., Barry, H., Kabadjov, M., Klein, E., Matthews, M., Roebuck, S., Tobin, R., & Wang, X. (2008). The ITI TXM Corpora: Tissue Expressions and Protein-Protein Interactions. In *Proceedings of LREC'08*, Marrakech, Morocco, May.

[72] Collier, N., Doan, S., Kawazoe, A., Goodwin, R. M., Conway, M., Tateno, Y., & Taniguchi, K. (2008). BioCaster: detecting public health rumors with a Web-based text mining system. Bioinformatics, 24(24): 2940–2941.

[73] http://rweb.compbio.iupui.edu/corpus/.

[74] Wu, H., Karnik, S., Subhadarshini, A., Wang, Z., Philips, S., Han, X., Chiang, C., Liu, L., Boustani, M., Rocha, L. M., Quinney, S. K., Flockhart, D., & Li, L. (2013). An integrated pharmacokinetics ontology and corpus for text mining. *BMC Bioinformatics,* 14:35 doi:10.1186/1471-2105-14-35

[75] http://www.cypalleles.ki.se.

[76] http://www.tcdb.org.

[77] Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, C. A., & Wishart S. D. (2011). Drugbank 3.0: a comprehensive resource for 'Omics' research on drugs. *Nucl. Acids Res.* 39 (suppl 1): D1035-D1041.

[78] http://dbmi-icode-01.dbmi.pitt.edu/dikb-evidence/package-insert-DDI-NLP-corpus.html.

[79] Boyce, R., Gardner, G., & Harkema, H. (2012). Using Natural Language Processing to Extract Drug-Drug Interaction Information from Package Inserts. *Proceedings of the Workshop on BioNLP*. Montreal, Quebec, Canada, June.

[80] Rubrichi, S., & Quaglini S. (2012). Summary of Product Characteristics content extraction for a safe drugs usage. *J Biomed Inform,* 45(2): 231–239.

[81] http://dailymed.nlm.nih.gov/dailymed/index.cfm.

[82] http://www.farmadati.it/Default.aspx.

[83] https://bitbucket.org/leondz/te3-platinum.

[84] Kolya, K. A., Kundu, A., Gupta, R., Ekbal, A., & Bandyopadhyay, S. (2011). A CRF Based Approach to Annotation of Temporal Expression, Event and Temporal Relations. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, June, pages: 64–72.

[85] http://labda.inf.uc3m.es/DDIExtraction2011/dataset.html.

[86] Lamurias, A., Grego, T., & Couto M. F. (2013). Chemical compound and drug name recognition using CRFs and semantic similarity based on ChEBI. *BioCreative Challenge Evaluation Workshop* 2, 75

[87] http://svmlight.joachims.org.

[88] Chowdhury, M. F., Lavelli, A., & Kessler, B. F. (2013). A Multi-Phase Kernel Based Approach for Drug-Drug Interaction Detection and Classification that Exploits Linguistic Information. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, June, pages: 351–355.

[89] Bokharaeian, B., & Diaz, A. (2013). Extracting Drug-Drug interactions from text through combination of sequence and tree kernels. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, June, pages: 644–650.

[90] Mojarad, R. M., Boyce, D. R., & Prasad, R. (2013). Classifying Drug-Drug Interactions with Two-Stage SVM and Post-Processing. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, June, pages: 667–674.

[91] Hailu, D. N., Hunter, L. E., & Cohen, B. K. (2013). Extraction of Drug-Drug Interactions from BioMedical Text using Knowledge-rich and Knowledge-poor Features. *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, June, pages: 684–688.

[92] http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html.

[93] Minard, L. A., Makour, L., Ligozat, L. A., & Grau, B. (2011). Feature Selection for Drug-Drug Interaction Detection Using Machine-Learning Based Approaches. *Proceedings of the 1st Challenge task on Drug-Drug Interaction Extraction*, Huelva, Spain, September, pages: 43-50.

[94] Bedmar, S. I., Martı´nez, P., Bedmar, & S. M. (2008). Drug name recognition and classification in biomedical texts, A case study outlining approaches underpinning automated systems. *Drug Discovery Today*, 13(17–18): 816-823.

[95] Tjong, F. E., Sang, K., & Veenstra, J. (1999). Representing Text Chunks. In *Proceedings of EACL'99*, Bergen, Norway. pages: 173-179..

[96] Goutte C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In *Proceedings of the European Colloquium on IR Research (ECIR'05)*, LLNCS 3408 (Springer), Santiago de Compostela, Spain, March, pages: 345-359.

[97] Han, J., & Kamber, M. (2006). Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann Publishers. ISBN 1-55860-901-6.

[98] Byun, H., & Lee, S. W. (2002). Applications of Support Vector Machines for Pattern Recognition: A Survey. Pattern Recognition with Support Vector Machines, Volume 2388, ISBN : 978-3-540-44016-1.

[99] Weston, J., & Watkins, C. (1998). Multi-class support vector machines. *Technical Report CSD-TR-98-04*, Department of Computer Science, Royal Holloway, University of London.

[100]. http://chasen.org/~taku/software/yamcha.

[101] http://chasen.org/~taku/software/TinySVM/.

[102] Tjong, F. E., Sang, K., & Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language*

learning - Volume 7 (ConLL '00), Vol. 7. Association for Computational Linguistics, Stroudsburg, PA, USA, pages: 127-132.

[103] Klinger, R., Koláˇrik, C., Fluck, J., Apitius, H. M., & Friedrich, M. C. (2008). Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, 24:i268-i276.

[104] http://crfpp.googlecode.com/svn/trunk/doc/index.html.

[105] Zazo, H. M., Bedmar, S. I., & Martinez, P. (2012). Annotation Guidelines for DDI Corpus. Working Paper, University Carlos III of Madrid, Spain.

[106] Bedmar, S. I., Martınez, P., & Cisneros, S. D. (2011) The 1st DDIExtraction-challenge task: extraction of drug–drug interactions from biomedical texts. In: *Proceedings of the 1st Challenge task on drug–drug interaction Extraction* (DDIExtraction 2011), Huelva, Spain, September.

[107] http://labda.inf.uc3m.es/doku.php?id=en:labda_ddicorpus.

[108] Liu, H., & Hiroshi, M. (1998). Feature extraction, construction and selection: A data mining perspective. Kluwer Academic Publishers, Norwell, MA, USA. ISBN: 0792381963.

[109] Klinger R., Kolarik C., Fluck J., Hofmann, A. M., & Friedrich C. M. (2008). *Detection of IUPAC and IUPAC-like Chemical Names. Bioinformatics. 24(13):i268–i276.*

[110] Mitsumori, T., Fation, S., Murata, M., Doi, K., & Doi, H. (2005). Gene/protein name recognition based on support vector machine using dictionary as features. *BMC Bioinformatics,* 6:S8. doi:10.1186/1471-2105-6-S1-S8.

[111] Zhou G., Zhang J., Su J., Shen D., & Tan C. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics* 20 (7): 1178-1190.

[112] Sagae, K., & Tsujii, J. (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. *Proceedings of the CoNLL 2007 Shared Task.* (EMNLP-CoNLL'07). Prague, Czech Republic. pages: 1044–1050.

[113] Collier, N., & Takeuchi, K. (2004). Comparison of character-level and part of speech features for name recognition in biomedical texts. *Journal of Biomedical Informatics* 37: 423–435.

[114] Tseng, H., Jurafsky, D., & Manning, C. (2005). Morphological features help POS tagging of unknown words across language varieties. *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

[115] Malmberg, K. J., Steyvers, M., Stephens, J. D., & Shiffrin, R. M. (1997). Feature Frequency Effects in Recognition Memory. *Journal Article, Springer-Verlag,* pages: 607-613.

[116] Qiu, Q., Jiang, Z., & Chellappa, R. (2011). Sparse Dictionary-based Representation and Recognition of Action Attributes. IEEE Conference on Computer Vision, Pages: 707-714.

[117] Liu, T., Liu, S., Chen, Z., & Ma, W. Y. (2003). An Evaluation on Feature Selection for Text Clustering. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, Washington DC.

[118] Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research,* 3 (Mar): 1157-1182.

[119] Liu, J., Ranka, S., & Kahveci, T. (2008). Classification and feature selection algorithms for multi-class CGH data. In Journal of *Bioinformatics.* 24(13): i86–i95.

[120] Brank, J., Grobelnik, M., Frayling, N. M., & Mladenic, D. (2002). Feature Selection Using Linear Support Vector Machines. In *Proceedings of the 3rd Int. Conf. on Data Mining Methods and Databases for Engineering, Finance, and Other Fields*, Bologna, Italy, September.

[121] Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis,* 1(1–4): 131-156.

[122] Zhang, T. (2011). Adaptive Forward-Backward Greedy Algorithm for Learning Sparse Representations. *IEEE Trans. Inf. Theor.* 57(7): 4689-4708.

[123] Skurichina, M., & Duin, R. P.W. (2005). Combining Feature Subsets in Feature Selection. Multiple Classifier Systems, 6th International Workshop, MCS, Seaside, CA, USA, Proceedings. Pages: 165-176.

[124] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence* (IJCAI), Montreal, Quebec, Canada, August, Pages: 1137-1145.

[125] Ruta, D., & Gabrys, B. (2005). Classifier Selection for Majority Voting. *In Journal of Information Fusion* 6(1): 63-81.

[126] Lam, L., & Suen, C. Y., (1997). Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. *IEEE Transactions on Systems Man and Cybernetics* - part A: Systems and Humans, 27(5): 553-568.

[127] Zhu, X., Wu X., & Yang, Y. (2004). Dynamic Classifier Selection for Effective Mining from Noisy Data Streams. *In The Fourth IEEE International Conference on Data Mining (ICDM)*, Brighton, UK, November, Pages: 305-312.

[128] Chang, C. C., & Lin, C. J. (2011). LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology,* 2:27:1--27:27.

APPENDIX

Appendix A: Statistics Measures

Different statistics measures such as Recall, Precession and F-score are used to measure the performance of system. Confusion matrix is composed of 4 terms such as TP, FP, TN, and FN. True Positive (TP) refers to number of positive samples which are classified correctly. True Negative (TN) is number of negative examples which are identified correctly. False Positive (FP) denotes number of negative examples which are classified incorrectly as positive examples and finally False Negative (FN) indicates number of positive examples which are identified incorrectly as negative examples.

- Recall or sensitivity is the proportional of correctly classified positive examples.

$$Recall = \frac{TP}{TP+FN} \qquad \text{Eq. 1.1}$$

- Precision or positive predictive value (PPV) is the proportion of examples classified to be positive that were correct.

$$Precision = \frac{TP}{TP+FP} \qquad \text{Eq. 1.2}$$

- F-score is the harmonic average of two other well-known performance measures that are referred as recall and precision and is usually used for measuring the overall performance of NER tasks.

$$F\text{-}Score = \frac{2*Precision*Recall}{Precision+ Recall} \qquad \text{Eq. 1.3}$$

In order to calculate overall F-score among all classes in multi-class NER tasks, there is a need to compute the number of TP, TN, FP, and FN for each class. There are two ways to determine the overall F-score: 1) by computing the average of the individual F-scores which is named as Macro-average F-score 2) by counting the total TP, FP, FN and TN for all NEs in the data set which is named as Micro-average F-score. In this study Micro-average F-score is used.