Chemical Named Entity Recognition using Undersampling and Classifier Ensembles

Abbas Akkasi

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

Doctor of Philosophy
in
Computer Engineering

Eastern Mediterranean University
May 2016
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Cem Tanova
Acting Director

I certify that this thesis satisfies the requirements of thesis for the degree of Doctor of Philosophy in Computer Engineering.

Prof. Dr. H. 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 Doctor of Philosphy in Computer Engineering.

Assoc. Prof. Dr. Ekrem Varoğlu
Supervisor

Asst. Prof. Dr. Nazife Dimililer
Co-supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay
2. Prof. Dr. Nizamettin Aydın
3. Prof. Dr. Tolga Çiloğlu
4. Prof. Dr. Hasan Demirel
5. Assoc. Prof. Dr. Ekrem Varoğlu

ABSTRACT

Chemical Named Entity Recognition (ChemNER) is the first step for a large number of consequent Information Extraction (IE) tasks in the chemistry related sciences and drug development domains. Extraction of drug-drug interactions, chemical compounds' resolution, and creation of question answering systems are examples of such applications. Any improvement in the quality of NER process in this context may affect the performance of subsequent tasks which shows the importance of this preliminary step in IE applications. In this thesis we studied this problem by proposing a modular architecture to improve the performance of ChemNER systems. This thesis has three main contributions to the overall task. The first contribution is the design of a new rule based tokenizer which improves the quality of data preprocessing phase. Due to the highly imbalanced nature of the data used in the NER task, overall performance of the classifiers used is usually not as good as those used in some other common classification tasks. Hence, a new sentence based undersampling approach specifically to be used for the NER problems is proposed as the second contribution for the given problem. The proposed undersampling approach tries to remove the insignificant samples from the training data aiming at preserving the structure of the given sentences as much as possible. We name it as Balance Undersampling (BUS) approach since it tries to keep almost an equal number of negative samples surrounding the positives. The third contribution of this thesis is to use the Particle Swarm Optimization algorithm as a heuristic classifier selection method together with the Naïve Bayesian combination approach to form a classifier ensemble from a large pool of classifiers created using undersampled data with different sampling ratios and various feature sets. All experiments during this

study are conducted using the BioCreative IV ChemDNER corpus which is the most comprehensive data set in the domain.

Keywords: Chemical Named Entity Recognition, Tokenization, Undersampling, Classification, Classifier Ensemble, Particle Swarm Optimization.

ÖZ

Kimsayal Adlandırılmış Varlık Tanıma (KAVT) kimya ve eczacılık ile ilgili alanlarda bilgi çıkarımı öncesi yapılması gereken ilk işlemlerden biridir. İlaçlar arası etkileşimlerin çıkarılması, kimyasal bileşenlerin çözünürlüğünün ortaya çıkarılması ve otomatik soru-cevap sistemlerinin yapımı bu işlemlerden bazılarıdır. Bu sebepten dolatyı KAVT basamağında yapılacak tüm iyileştirmeler, takip eden sistemlerinin başarısını büyük ölçüde etkilemektedir. Bu tezde KAVT problemi ele alınmış ve KAVT sistemlerinin başarımını artırmak için birimsel bir mimari önerilmiştir. Bu anlamda tezin literatüre üç temel katkısı vardır. Birinci katkı olarak metin önişleme işlemleri sırasında performamsı artırmak için yeni bir kural-tabanlı alıntı ayırıcı önerilmiştir. KAVT işleminde kullanılan verinin doğal nedenlerle sınıflar arası dengesiz olmasından dolayı, sınıflandırıcıların başarımı genellikle yüksek olmamaktadır. Bu nedenle, ikinci katkı olarak cümle-tabanlı yeni bir alt-örnekleme yöntemi önerilmiştir. Önerilen yöntem, eğitme veri kümesinde bulunan önemsiz örnekleri cümlenin yapısını en az bozacak şekilde çalışmaktadır. Tüm olumlu örneklerin sağ ve sol taraflarından eşit miktarda olumsuz örneği eğitme veri kümesinden çıkardığı için önerilen yönteme Dengeli Alt-örnekleme (DAÖ) ismi verlimiştir. Üçüncü katkı ise, çoklu sınıflandırcı yöntemi kullanılmasıdır. Bu yöntemin kullanılmasında Parçacık Apaçık Eniyileme yöntemi algoritması sınıflandırıcı seçimi için kullanılmış, seçilen sınıflandırıcılar ise Bayesçi Birleştirme yöntemi ile birleştirilerek alt-örneklenmiş örnekler kullanılarak eğitilmiş büyük bir sınıflandırıcı topluluğu elde edilmiştir. Bu çalışmada, ilgili alanda en büyük bütünce olarak bilinen BioCreative IV ChemDNER bütüncesi kullanılmıştır.

Anahtar Kelimeler: Kimsayal Adlandırılmış Varlık Tanıma, Alıntı Ayırıcı, Alt-örnekleme, Sınıflandırma, Sınıflandırcı Topluluğu, Parçacık Apaçık Eniyileme.

This Thesis is dedicated to my Family.

For their endless love, supports and encouragement.

ACKNOWLEDGMENT

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

Asst. Prof. Dr. Nazife Dimililer, my co-supervisor, helped me with various issues during the thesis and I am grateful to her. I am also obliged to Prof. Dr. Hakan Altınçay and Prof. Dr. Hasan Demirel, thesis monitoring jury members, for their help during my thesis.

I owe quite a lot to my family who allowed me to travel all the way from Iran to Cyprus and supported me all throughout my studies. I would like to dedicate this study to them as an indication of their significance in this study as well as in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

x	Input Vector
y_i	i^{th} class label
y_{Si}	Predicted label with classifier E_i
N	Number of class labels
N_j	Number of samples belong to y_j
L	Number of classifiers
BioCreative	Critical Assessment of Information Extraction Systems in Biology
ChemDNER	Chemical/Drug Named Entity Recognition
CRFs	Conditional Random Fields
E_j	j^{th} classifier
F_i	i^{th} feature set
CM^i	Confusion matrix for i^{th} classifier
μ_k	Computed score with Naïve Bayesian combine for class y_k
B	Titterington constant
NER	Named Entity Recognition
CIP	Class Imbalance Problem
IR	Imbalance Ratio
N_{maj}	Number of samples from majority class
N_{min}	Number of samples from minor class
BUS	Balanced Undersampling
RUS	Random Undersampling
V_{id}	The velocity of d^{th} entry in particle i
x_{id}	The position of d^{th} entry in particle i

χ	Constriction factor
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
p	Precision
r	Recall
MUC	Message Understanding Conference
NE	Named Entity
MCS	Multiple Classifier System
SWF	Stop Word Filtering
PSO	Particle Swarm Optimization
CFM	Constriction Factor Method
BioTM	Biomedical Text Mining
NLP	Natural Language Processing
JNLPBA	Joint workshop on Natural Language Processing in Biomedicine and its Applications
TREC	Text Retrieval Conference
SMILES	Simplified Molecular Input Line Entry System
IUPAC	International Union of Pure and Applied Chemistry
InChi	International Chemical Identifier
UMLS	Unified Medical Language System
SL	Supervised Learning
UL	Unsupervised Learning
SSL	Semi-Supervised Learning

HMM	Hidden Markov Model
MEMM	Maximum Entropy Markov Model
SSVM	Structure Support Vector Machines
SCS	Static Classifier Selection
DCS	Dynamic Classifier Selection
FS	Forward Selection
BE	Backward Elimination
MV	Majority Voting
SMOTE	Synthetic Minority Oversampling Technique
NT	Number of Tokens
ATL	Average Token Length
NISE	Number of Incorrectly Segmented Entities
BPSO	Bayesian method for combination and PSO for Selection
MVPSO	Majority Voting for combination and PSO for Selection
BFS	Bayesian Method for Combination and Forward Selection as selection method
MVFS	Majority Voting for combination and Forward Selection as selection method
BBE	Bayesian Method for Combination and Backward Selection as selection method
MVBE	Majority Voting for combination and Backward Selection as selection method
BFULL	Bayesian Method for Combination of All Classifier in the pool
MVFULL	Majority Voting for combination of All Classifier in the pool
SB	Single Best classifier

MVAWOS	Majority Voting for combination of Base line classifiers trained with Original train data
MVAWS	Majority Voting for combination of Base line classifiers trained with sampled train data
BAWOS	Bayesian Method for Combination of Base line classifiers trained with Original train data
BAWS	Bayesian Method for Combination of Base line classifiers trained with sampled train data

Chapter 1

INTRODUCTION

1.1 Motivation

The main aim of Natural Language Processing (NLP) is to design and implement software that can process, comprehend and generate natural language text. Even though natural language understanding remains an important challenge, text mining which emerged as an important research field in NLP, focuses on discovering hidden information from unstructured textual documents. Many practical text mining applications including Information Retrieval (IR), Information Extraction (IE), and Question Answering (QA) systems have been developed in the past few decades. IE is one of the basic and important applications of text mining that involves extraction of desired information by transforming facts in texts into structured representation [1]. Recent progress in scientific research and practice in pharmaceutical and chemical fields have caused proliferation of information in unstructured textual format [2], [3]. Scientific ideas, hypothesis, facts, and conclusions derived from scientific experiments, as well as academic or industrial conclusions are published in the form of unstructured documents. In recent years the chemical domain has been facing a large amount of textual data published daily. The accumulation of vast amounts of scientific text in chemical domain triggered an urgent requirement for the development of text mining techniques to extract valuable information from this huge volume of literature [4], [5]. Text mining in the chemical domain may enable

and support drug discovery and development process by assisting the scientists to quickly screen through millions of documents and discover novel insights.

Due to the abundance and continuous accumulation of unstructured scientific text, chemical domain has become one of the most active domains of text mining. The high production rate of literature in this domain is the main obstacle to timely processing of text by human experts. Therefore, the use of text mining techniques to extract meaningful and useful knowledge within a reasonable frame has become mandatory.

IE as one of the main subtasks of text mining, aims to automatically extract structured information from unstructured or semi structured text. Information extraction encompasses a number of subtasks including question answering, relation extraction, event detection, text summarization, and co-reference resolution. Most of these tasks have been introduced by the Message Understanding Conference (MUC) and financed by Defense Advanced Research Project Agency (DARPA) to encourage the development of new and better methods of IE [6]. The fundamental step of IE, affecting the performance of all mentioned subtasks is Named Entity Recognition (NER) which aims to identify and categorize existing priori specified named entities in a given text. The "Named Entity" (NE) task appeared for the first time in the Sixth MUC conference [7]. The list of class types in NER tasks are generally predefined and the task can be defined as classifying a portion of text as a NE mention and associating the NE with one of the predefined class types. For example, consider the text "*Michel took an Acetaminophen. He had headache because of too much alcohol that he drunk last night.*" In this text there are four entities with different class types as shown in Table 1.1.

Table 1.1: A sample of named entity classification

Named Entity	Class Type
Michel	Person
Acetaminophen	Drug
alcohol	Chemical
Last night	Time

NER as a classification task, borrows some algorithmic techniques from the machine learning domain as well as NLP. Moreover, considering it as a kind of sequence labeling task [8], NER suffers from common challenging issues in this field such as lack of standard feature sets, class imbalance problem in machine learning approach, difficulties in defining regular expressions, and creating comprehensive repository of named entities.

Quality of the output of NER systems has direct impact on the quality of subsequent tasks since they make use of the NEs. For instance, final results of extraction of pathways, metabolic reaction relation, drug-protein interactions in biochemical domain are greatly affected by outcomes of NER process. Hence efficient detection of named entities in given text is essential for the majority of text mining applications in all domains and especially in the chemical domain.

The work described in this thesis focuses on NER in the chemical domain in the context of supervised machine learning approach. Chemical NER is concerned with the identification of chemical entities such as chemical descriptors, CAS registry numbers brand names and drug names [9], [10], [11]. Chemical NEs extracted from text are used in many processes including drug discovery, chemical research and

manufacturing processes and thus are of immense value for the pharmaceutical and drug industries. [12]. However, the high rate of growth in chemical literature has made it increasingly difficult to get acceptable results in a reasonable time frame. Initial research on chemical NER aimed at designing dictionary or rule based systems. However, the performance of such systems has been affected by comprehensiveness of dictionaries or generality of extracted rules. Therefore, subsequent work focused on constructing systems using machine learning approaches by exploiting wide variety of features and hybrid methods combining different strategies. These systems mostly try to maximize recognition performance by computing discriminative set of features or enhancing the outcomes of existing NER systems [13-20]. An alternative to finding the best performing classification system is to combine sufficiently efficient classifiers, weak learners, in a multi classifier system (MCS) or classifier ensemble [21], [22], [23].

Even though NER systems in the newswire domain have achieved high performances, F-score around 96% [24], due to the special intricacies of the literature in the chemical domain, performance of NER systems in this domain, is still far from satisfactory (F-score of around 87% [25]). The relatively poor performances in this domain mainly are generally attributed to several reasons: i) Diversity in chemical nomenclatures; chemical entity mentions within literature can be found in different forms such as: systematic or semi systematic names, brand name, formula [12], ii) Extensive use of abbreviations, ambiguous names, homonyms, and existence of non-usual characters and symbols inside entity names, iii) Inconsistent use of white spaces and special characters such as punctuation marks caused to the existence of different forms of tokenization for the same names, iv) Continuous generation of domain specific names some of which are used only for short periods, v) Chaining of

NEs with conjunctions and disjunctions in the sentence, vi) Scarcity of freely available, comprehensive and well annotated dataset with complete annotation guidelines.

In addition to these problems a chemical NER system which uses machine learning approaches usually suffers from the class imbalance problem [26]. Observation on the available data sets reveals that the number of named entities of interest, which are considered as positive samples, are drastically lower than the other segments of texts that are called negative samples.

This thesis proposes a novel framework for chemical NER that identifies the chemical entities in a given unstructured natural language text. The underlying classification architecture utilizes Conditional Random Fields (CRFs) [27] which is a machine learning algorithm. The first stage of the framework is a novel tokenizer called ChemTok [28] that accepts unstructured text and produces a list of tokens. ChemTok is designed to handle the peculiarity of the language used in chemical/drug domain. Feature extraction stage augments the tokenized text with features that are widely used in NER systems. In order to overcome the class imbalance problem, a number of classifiers are trained using undersampled data. Due to the special nature of NER as a sequence labeling problem, we propose a novel undersampling algorithm called Balanced Undersampling (BUS) for this stage.

1.2 Methodology

In this study we describe a framework to recognize chemical named entities in unstructured text. ChemDNER dataset [29] released by BioCreative IV [30] is utilized for training the classifiers used since the aforementioned dataset is the most

comprehensive and standard dataset available in the chemical domain. ChemDNER corpus includes three datasets: training, development and test set. Preliminary experiments on the dataset revealed the tokenization problems when standard tokenizers are used in the chemical domain [31]. Therefore, we proposed and implemented a more effective tokenizer, ChemTok [28] that can handle the special notations used in chemical/drug domain. ChemTok, employs a set of rules extracted from the ChemDNER training data. We tested and showed the performance of ChemTok on different data sets in the same domain.

Another novelty in our framework is the undersampling method we used for alleviating the class imbalance which is an inherent characteristic of NER in all domains and particularly in the chemical domain. A new undersampling method namely balanced under sampling which strives to keep the syntactic structures of training samples intact as much as possible while balancing the negative/positive ratio in the dataset is proposed. The output of BUS is a new training data set based on the desired ratio between negative and positive samples.

In the proposed framework, we train a large number of CRF classifiers using different combinations of well-known features and undersampled data. To use the strengths of different classifiers together, a newly designed classifier ensemble system using Particle Swarm Optimization (PSO) for classifier selection and Naïve Bayesian approach to combine classifiers, is applied to combine the outputs of predictors. Results show that both the proposed tokenization algorithm and the balanced undersampling method have positive impact on the classification performance of individual classifiers. Moreover, the proposed ensemble method further improves the performance.

1.3 Summary of Thesis Contributions

Developed framework in this study makes several contributions to the NER field in general and specifically to the chemical NER problem. These can be summarized as follows:

- A new tokenization method applicable for both chemical and biomedical context is devised. Experiments on the effect of tokenization on NER tasks show that it is more efficient than the commonly used tokenizer in this field.
- To deal with class imbalance problem in sequenced data used in the pattern recognition field, a new undersampling approach that has improved NER performance of classifiers is devised.
- Constriction Factor Method (CFM) as a kind of particle swarm optimization algorithm [32] is used in classifier selection phase of MCS in order to statically select experts.
- Naïve Bayesian combination method [33] is applied individually and also along with an evolutionary algorithm in classifier combination phase of the MCS for the NER task.
- The number of diverse classifiers used as members of the classifier repository for the final MCS is very high compared to the MCSs previously used for this problem [25], [34-36].

1.4 Research Objectives

The main objectives of this study are summarized as follows:

- To investigate the effects of tokenization on overall performance of NER systems and to develop a more efficient and domain-appropriate tokenizer for chemical domain.

- To investigate the effects of class imbalance phenomenon on the performance of chemical NER systems and propose a novel method for undersampling in NER.
- To develop a framework in order to identify chemical NEs in an efficient way by means of MCSs.
- To investigate current tools and available systems for chemical NER task.

1.5 Thesis Outline

The remaining of this dissertation is organized as follows: In Chapter 2 a brief explanation on biomedical text mining and its applications is followed by a discussion on chemical NER problem and existing strategies used to resolve this type of problems. Moreover, an in-depth literature review on Chemical NER is presented in the same chapter. Chapter 3 presents an overview of multiple classifier systems and its main components including classifier selection methods and combination approaches. Chapter 4 provides the background knowledge on the class imbalanced problem in different contexts. The strategies and algorithms to decrease the adverse effect of class imbalance on the performance of classifiers are presented in detail in the same chapter. The architecture of the proposed framework is presented and explained in Chapter 5. Additionally Chapter 5 contains a general discussion on different parts of the proposed system, extracted features and prototype of individual classifiers. In chapter 6 the results of employing the proposed system is provided. Finally, in Chapter 7, a summary of the discussion on the results and future work direction in this area are presented. Explanation of classifier evaluation metrics, details of CRFs' algorithm, and individual classifiers performances are given in appendices.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Introduction

Recent developments in life sciences and especially in biomedical/chemical fields have triggered the explosive growth of literature in computer readable unstructured textual format. Processing of such voluminous information in turn, necessitated natural language processing and text mining techniques to automatically extract hidden information in order to make desired knowledge readily available to the experts in the field. The most important and challenging aspect of processing unstructured text, or text mining, is extracting specific facts, objects, events, and relations. Named entity recognition is generally a prerequisite to other text mining subtasks such as relation and event extraction, summarization and question answering. This chapter reviews the biomedical text mining research and its application on chemical literature in section 2.2. NER in chemical domain and the challenges faced in this research field are discussed in Section 2.3. Section 2.4 presents current strategies used in NER systems. A detailed literature review on chemical NER is provided in Section 2.5.

2.2 Biomedical and Chemical Text Mining

Text mining attempts to discover or extract implicit knowledge hidden within unstructured text [37]. Research on text mining has dramatically increased in life sciences especially in biomedical and chemical domains, where journal articles, books, reports, patents etc. are being produced in an increasingly higher pace in the

past few years. The rapid production of knowledge makes it difficult for scientists to keep up to date [38], thus, there is an immediate demand to enable access to the useful desired information. Biomedical Text Mining (BioTM) refers to the text mining process applied on the biomedical, chemical and drug literature. It is a new research field spanning a number of research fields such as NLP, text mining, bioinformatics, cheminformatics, medicine and drug development and computational linguistics. The basic goal of BioTM is to allow experts in field to extract knowledge from relevant documents thus facilitating new discoveries in more efficient manner [39], [40]. The main developments, in this area have been focused on the identification of biological or chemical entities such as drugs, genes, proteins, chemical compounds etc. within the given free text [41]. Text mining and information extraction methods have also been applied to extract the information related to biological and chemical processes, events, and relationships. However since these applications require NER as a preliminary task, it is crucial to improve the NER process.

A large number of scientific events such as shared tasks or competitions, which have been conducted on different applications of BioTM in recent years, show the increased interest and requirement in these fields. Text Retrieval Conference (TREC) chemical track 2011 [42], Joint workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), Bio-Entity recognition challenge [10], BioNLP shared task 2013 [43], Critical Assessment of Information Extraction systems in Biology (BioCreative) IV and V (2013 and 2015 respectively) [30], [44], [45], Linking literature, Information and Knowledge for Biology (BioLINK SIG 2013) [46] are examples of such shared tasks. The main aim of all aforementioned

events was to find efficient methods to extract useful information from the unstructured documents in the biomedical, chemical and drug related fields.

The chemical track of TREC 2011 focused on evaluation of search technologies for retrieval and knowledge discovery of digitally stored information on chemical patents and academic journal papers. The aim of Bio-Entity recognition task at JNLPBA program was to identify entities in the domain of molecular biology that corresponded to the instances of concepts that are of interest to scientists. The BioLINK SIG has been regularly held since 2001 and its main focus is on the development of tools for biomedical text mining. BioCreative IV and V challenges included various tasks in biomedical fields. Both of them have organized special tracks on information extraction from chemical texts. These tracks were divided into two parts: chemical named entity recognition and chemical document classification.

Figure 2.1 illustrates an overview of IE task in biomedical field and clearly shows the importance of NER in this framework. The first step in the general IE framework involves selecting the required documents that will be used from the vast amounts of documents available to the public. The selected documents are then normalized and annotated with mentions of interests. In the next step NER is applied to the normalized documents. Methods used for NER are discussed in detail in subsequent sections of this chapter. Named entities recognized at the NER step can then be utilized for populating ontologies or as input for other tasks such as relation discovery, summarization and question answering.

2.3 Chemical Named Entity Recognition

A named entity is a phrase that clearly identifies one item from a set of others which have similar attributes. For instance persons, dates, geographic locations and organization names are examples of named entities in newswire domain.

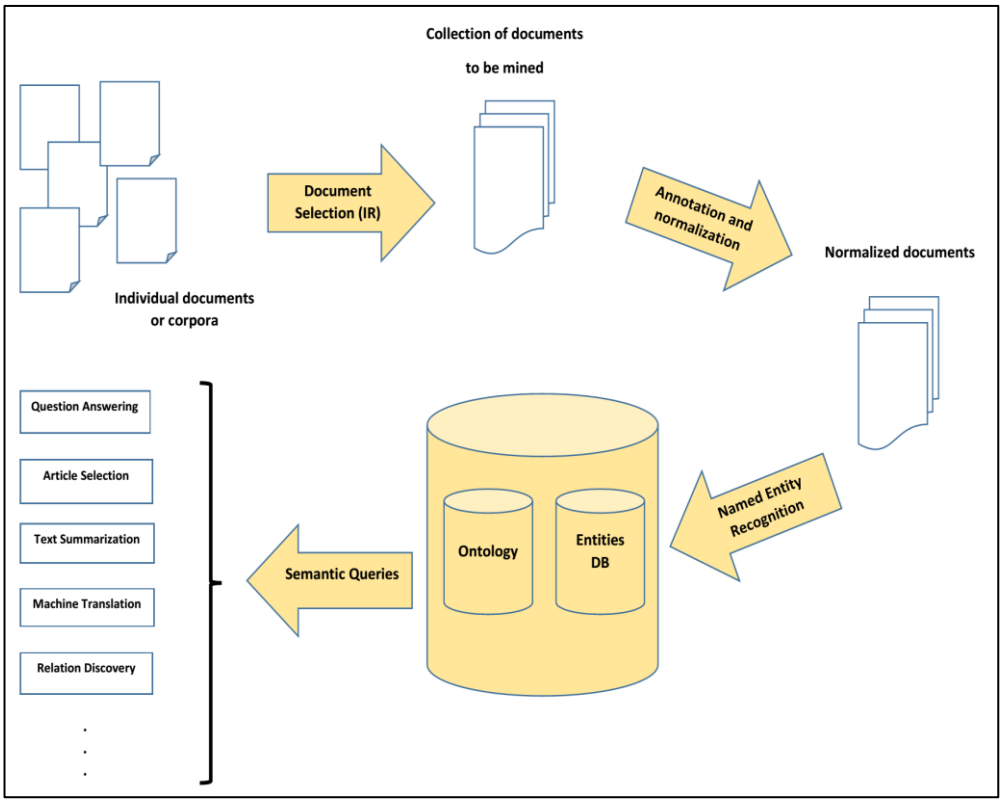

Figure 2.1: Overview of IE task in biomedical domain

In the chemical context a named entity can refer to drug names, chemical compounds, formulas, abbreviations etc. that appear in given document possibly in different formats. In chemical literature, locating such entities is crucial for many tasks such as identification of relationships or interactions between the entities and the retrieval of documents of interest. The process of recognition of chemical entity mentions from unstructured text and assigning the pre-determined class labels to them is known as "Chemical Named Entity Recognition "*ChemNER*" or "*Chemical Semantic Tagging*". Use of text mining approaches in drug discovery and chemical

research has been an active area of research interest in recent years [47]. Class labels for chemical entity mentions can be categorized by their structures such as: abbreviation, systematic, semi-systematic [45]. The majority of related work in this field has been done on the detection of genes and protein names in biomedical texts and very few studies focused on the chemical compounds or drug related terms until recently [48].

2.3.1 Difficulties Appear in Chemical NER Process

As mentioned in Chapter 1, due to several reasons such as ambiguity, different nomenclature, writing style etc. the performance of named entity recognition systems in biomedical and especially chemical context achieved less success than newswire domain. Some of the main causes of the difficulties in chemical literature are described in more detail below.

- **Lack of a universal standard for chemical entity representation**: Usually chemical entities are referenced in documents in different forms including common names (trade name), data base identifiers, systematic nomenclature, CAS registry numbers, International Chemical Identifiers (InChI) [49], Simplified Molecular-Input Line-Entry System (SMILES) codes [50], or schematic structures and images. Different coding and identification approaches have different word formation characteristics described by their own guidelines which makes it difficult to recognize the chemical NEs easily. Figure 2.2 depicts an example of various naming methods that can be used in literature to represent the same entity.

  In general, naming approaches can be divided into two groups: systematic and non-systematic. Systematic nomenclature uses a set of rules to name chemical compounds. Even though the most widely used systematic

method is the one created by International Union of Pure and Applied Chemistry (IUPAC) [51], many other systematic naming approaches such as CAS Index Names, InChI, and SMILES, may be utilized by the researchers in this field.

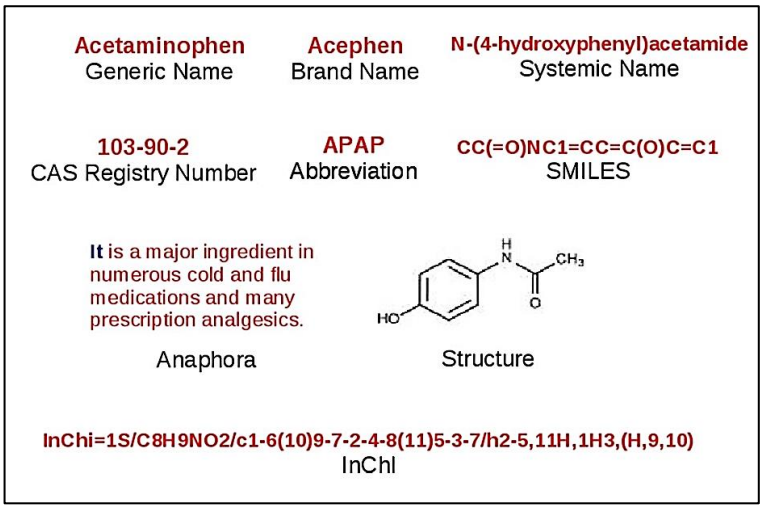

Figure 2.2: Diversity in the representation of chemicals

In addition to the systematic nomenclature, there is a widespread use of generic and trade names in the texts due to their popularity or simplicity. For instance an entity with IUPAC name "*3,7-dihydro-1,3,7-trimethyl-iH-purine-2,6-dione*" is commonly used with the name "*coffein*". The ambiguity of chemical names especially in their common or trivial forms is another cause of difficulty in the recognition of chemical information given in texts.

Different chemicals which have different physio-chemical properties can be referred using their trivial name [52]. For example "*acetylacetone"* may refer to either one of its two tautomeric forms, "*keto*" or "*enol*". In that case recognizing and identifying the chemical unambiguously becomes very difficult.

Use of semi-systematic naming method provides another unique challenge in chemical NER. Entities that are named semi-systematically usually contain a mixture of both systematic name and nonsystematic name fragments. For example in the name *"3',5'-dichloromethothotrexate"* , the chunks "*di"* and "*chloro"* are generated using systematic naming method whereas *methoraxate* is trivial drug name.

- **Presentation of Chemical information in image format**: Patent documents are usually available as images of texts documents (e.g. PDF or TIFF). Such documents are often converted from those file formats to text by means of Optical Character Recognition (OCR). OCR documents usually have interpretation errors or loss of graphical images that may contain chemical structure diagrams. For example "*EXAMPLE 22. Amino-3,4*" may be converted to "*EXAMPLE 22- Amino-3 4.*" [3].
- **Difficulties in mining patents**: Patent documents are often written by patent agents or attorneys who are not familiar with scientific writing standards [53]. To formulate the claims in patents, usually a narrative style is used. For instance the patent writers may express a claim in the broadest way possible, making formulation ambiguous and prone to misinterpretation.
- **Widespread and inconsistent use of abbreviations**: Despite the widespread use of abbreviations in chemical texts, the lack of a standard and unique procedure for abbreviation construction makes their detection very difficult. The position of the first mention of abbreviations may also differ. In some texts, abbreviations appear after the entity names whereas in others they appear before the actual entity name. Furthermore, the abbreviations may be introduced by a complete sentence or a phrase or it may be separated from

rest of the text with parentheses, comma, or dashes. For example, some abbreviations are produced from the first letter of the components of a multi token entity mention such as AAAD for "*Aromatic Amino Acid Decarboxylase*". On the other hand some abbreviations are made of initials of the syllable. For example 5-HMF for *"5-HydroxyMethyl-Furfural"* [54].

- **Nested named entities**: In the chemical NER domain it is very common to use an entity name inside another entity name. This phenomenon is known as nested named entities. The nested named entity problem makes recognition of the entities difficult and is often ignored in NER studies and only the outermost entities commonly are taken into account [55].
- **Continuous addition of new names**: Biomedical and chemical domains are rapidly developing research fields and thus vast amounts of publications are being produced as outcomes of new discoveries and research. Hence the rate of newly added named entities to the literature is high and it makes dictionary-based NER systems inefficient.

2.4 Approaches to Implement Chemical NER Systems

The approaches used for creating NER systems can be categorized into three groups: dictionary based approaches, context or learning based approaches and rule or morphology based approaches. Furthermore any combination of these three methods, known as hybrid approach [56] can also be used. The following subsections describe the different methods focusing on chemical literature and provide information on their characteristics.

2.4.1 Dictionary Based Methods

Dictionary based methods refer to a family of techniques that discover entity mentions in text by looking up the existence of the entities in a predefined repository

or dictionary. Hence, constructing dictionaries of good quality and implementation of efficient search or look up algorithms are mandatory for dictionary based methods. A critical aspect for success of dictionary based methods is to create dictionaries that are as comprehensive as possible. Dictionaries can be generated manually or automatically from related resources such as public chemical databases which usually contain lists of words that are grouped together based on their semantic similarities. A commonly used resource is Unified Medical Language Systems (UMLS) [57].

Even though it might seem like an advantage to combine a number of dictionaries together, size of combined dictionaries that may contain several millions of entries is usually much larger than a typical dictionary. For example, *Joint Chemical Dictionary* (JoChem) [18] consists of more than 2 million synonyms, while typical dictionaries containing gene names contains tens of thousands of entries. The most comprehensive dictionary for drugs and chemical compounds is the JoChem, which is created by merging several lexical resources such as PubChem [58], DrugBank [59], and Mesh terms [60]. Another example for chemical dictionary is ChemSpider [61] which in comparison to JoChem, it has fewer but higher quality entries.

A drawback of dictionary based methods is the need for extensive manual curation to maintain the dictionaries, add new entries and eliminate redundant entries. Another drawback is that dictionaries are not very effective in looking up incorrectly or differently spelled words; it is necessary to enhance either the dictionaries or the look up algorithms to allow the potential orthographic or spelling variations. Usually a string comparison metric such as Levenshtain distance method [62], which produces

an overhead on the lookup function of the dictionary based methods, is utilized to find matches even when there is spelling variations in strings.

2.4.2 Learning Based Methods

In Machine Learning-based NER systems, the purpose of NER approach is to convert the identification problem into a classification task and employ a classification model to solve it. In this approach, the system looks for patterns and relationships in text to make a model using statistical models and machine learning algorithms. [20].

The main idea behind learning based methods is to infer general patterns or models from sample instances that can be used subsequently to make predictions or classify unseen data; thus they require data to learn from. Learning process can be performed in three ways: Supervised learning (SL), Semi Supervised learning (SSL), and Unsupervised Learning (USL).

Almost all variants of SL approach typically consist of learning or deducing a "model" from a large set of annotated data known as train data that is usually enhanced by addition of discriminative features. The model created is then used to label or recognize entity mention in unlabeled data.

Unsupervised learning (UL) approaches make deductions using unlabeled input data. The most commonly employed UL approach is clustering where the unlabeled train data is separated into a number of clusters using distance or similarity metrics. After the clusters are formed using the input data, new data is easily categorized by computing its distance from or similarity to each of the clusters. UL techniques

typically rely on lexical resources such as MeSh, and UMLS, lexical patterns, and statistics computed on large unannotated data sets [63].

Semi-Supervised learning (SSL) or weakly supervised methods are combination of supervised and unsupervised approaches where a small set of annotated data is utilized to start learning process in addition to larger amount of unannotated data.

The most frequently used approach to create NER systems is the supervised learning method. CRFs [27] introduced in 2001, has been extensively used for NER and similar tasks ever since. CRFs are described in detail in Appendix B. Hidden Markov Models (HMM) [64], Maximum Entropy Markov Models (MEMM) [65], Structured Support Vector Machine (SSVM) [66] are other supervised machine learning algorithms that have been employed in this area. One of the difficulties in supervised machine learning approaches is the need for labeled or annotated training data, where the quality of the annotation has significant effect on the success of the approach.

2.4.3 Rule Based Methods

In rule based approaches a set of usually hand crafted rules are used to identify the entity mentions [67]. Manually hand crafted rule sets include syntactic and grammatical rules. In some cases rules are used in combination with dictionaries. In general two types of rules can be used in this approach: i) Context based rules that rely on the context of the words in the text [14] [68], ii) Pattern based rules that depend on the morphological or orthographic patterns of the words. [69]

If the experts are provided with the adequate resources and may derive comprehensive rule sets, rule based approaches may perform well, but if data is changed even slightly the cost of maintaining the rules may be quite high.

2.5 Previous Work on Chemical NER

Despite the importance of chemical NER, only a few of the chemical NER systems have been made publicly accessible [20]. Nevertheless, a considerable number of strategies and approaches for the recognition of chemicals in text have been proposed. There are some bottlenecks in implementation and comparison of the performances of such systems including: i) Lack of comprehensive train/test data set, ii) Difficulty in defining annotation guidelines of what actually forms a chemical entity name, iii) Diversity in terms of textual data sources and scopes used for data set creation. In this section a literature review on chemical NER is given. The corpora available in this research field are presented in the following subsections and the evaluation metrics used in this context are presented in Appendix A.

2.5.1 Chemical Corpora for NER Task

Current work in chemical text mining increasingly focused on the use of supervised machine learning approaches for NER problems [70]. Availability of a large manually annotated text corpus is necessary to develop such systems.

There are only few chemical corpora with manually labeled entities to use in text mining tasks unlike many other domains including biomedical domain. There are more than 36 corpora in biological area [71], a few of which contain chemical entities besides other types of entities. In addition to biological corpora, some other corpora have been developed specifically for chemical domain. Information about existing corpora is summarized in Table 2.1.

As shown in Table 2.1, ChemDNER is the largest and most comprehensive corpus in terms of the number of articles used in the chemical and drug domain. This corpus is

constructed using PubMed articles from different branches of chemistry and pharmacy, such as applied chemistry, pure chemistry, physical chemistry, organic chemistry etc. All experiments in this thesis are conducted using the ChemDNER corpus.

Table 2.1: Description of available chemical corpora

Corpus	**Main Focus**	**No. Of used articles**	**Availability**
GENIA [72]	Biological besides some chemicals	1999	http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA
CRAFT [73]	Biology	97	http://bionlp-corpora.sourceforge.net/CRAFT/
PennBioIE CYP 1.0. [74]	Biology	1100	https://catalog.ldc.upenn.edu/LDC2008T20
EU-ADR [75]	Biology	300	http://euadr.erasmusmc.nl/sda/euadr_corpus.tgz
ADE [76]	Biology	3000	Not Available
DDI [77]	Drug	700	https://www.cs.york.ac.uk/semeval-2013/task9/index.php%3Fid=data.html
EDGAR [78]	Biomedical	103	Not Available
Metabolites and Enzymes [79]	Metabolic	296	http://www.nactem.ac.uk/metabolite-corpus/metabolite-corpus-09012013.zip
IUPAC training [15]	Chemical (IUPAC names)	463	http://www.scai.fraunhofer.de/chem-corpora.html
SCAI [80]	All Chemical Names	100	http://www.scai.fraunhofer.de/chem-corpora.html
PubMed [81]	Compounds, reagents, chemical adjectives enzymes and prefix	42	Not Available
Sciborg [81]	All chemical names	42	Not Available
European Patent Office and the CheBI [17]	All chemical names	40	http://chebi.cvs.sourceforge.net/viewvc/chebi/chapati/patentsGoldStandard
ChemDNER [11]	Chemical compounds and drugs	10000	http://www.biocreative.org/tasks/biocreative-iv/chemdner/

2.5.2 Literature Review

NER in the biological domain has mainly focused on identifying gene or protein names, where a number of effective systems have been developed during the past few years [82],[83] such as BANNER [84], ProMiner [85], tmVar [86] and GNAT [87]. In contrast, chemical NER has received less attention. The earliest work on recognition of chemicals was performed in the late nineties. Heym et al. [88]

presented an algorithm to recognize and segment chemical words by matching the strings of characters with some stored words, similar to dictionary based method. Their work can be considered as the starting point for Chemical NER problem. Kemp and Lynch [89] developed a statistical method to detect chemical compound names in Standard Generalization Markup Language (SGML) patent texts. Wilbur et al. [90] implemented a system using both dictionary and learning approaches. To implement their dictionary based method, they created a list of chemical morpheme segments using the algorithm presented by *Registry File Basic Name Segment* dictionary [91]. The algorithm matches the longest left-most segment with character strings given in text. Furthermore they employed the naïve Bayesian algorithm in machine learning approach. The Open Source Chemistry Analysis Routines (OSCAR 3) developed by Corbett et al. [92] in 2008 to identify chemical entities is based on Maximum Entropy Markov Models (MEMM) [65]. It is tested on SCAI Corpus and PubMed Corpus, none of which are freely available. Jessop et al. [93] implemented OSCAR 4 by refactoring OSCAR 3. Rocktäschel et al. [94] reports that OSCAR 4 yielded a minor increase in performance compared to OSCAR 3. Klinger et al. [15] created a chemical NER system to detect IUPAC and IUPAC like entities using CRFs [64] algorithm. The implemented system is not freely available and does not cover trivial or drug names. Hetten et al. [18] implemented a combined dictionary for drug names, abbreviations, and small molecules using names extracted from the UMLS, MeSH, CheBI, DrugBank, HmdB, KEGG, and ChemIDplus. In 2008 Segura-Bedmar et al. [95], developed DrugNER system for recognition of drug names. They combined the UMLS MetaMAp Transfer (MMTx) program [96] and rules of nomenclature by the World Health Organization International Nonproprietary Names (HOINNs) program [97]. To evaluate the system, they used

their own DrugNER corpus, and reported a very high performance. ChemSpot is another state of the art chemical NER system created by Rocktäschel et al. [94]. It is implemented using a hybrid approach combining CRFs to identify systematic named entities and an exhaustive dictionary to detect other names such as brands, drugs, or small molecules.

Due to the sparsity of annotated corpora for training, failure in covering all types of chemical entities, and lack of publicly available annotating guidelines, it was not possible to evaluate efficiency of the proposed chemical NER systems until 2013. BioCreative IV organized a track on chemical/drug NER (ChemDNER), and invited researchers to develop their systems using presented corpus in 2013. 26 research teams have participated in task. Common characteristics of all teams were the use of the corpus to train systems or to adapt and fine-tune previously created systems. All participants employed the official evaluation library presented by BioCreative to evaluate and improve their systems during the development phase. Summary of techniques used for implementation of systems, subtasks of NLP attempted by the participants, and types of post processing employed are shown in Table 2.2. The first row of the table shows the reference number of the articles, which discuss the work and the rank of the system proposed for the ChemDNER task in terms of the achieved F-scores. Additionally, Table 2.3 summarizes the features used by the participating systems.

Table 2.2: Overview of the methods used for ChemDNER in BioCreative IV

	1[25]	2[98]	3[36]	4[99]	5[100]	6[101]	7[102]	8[103]	9[104]	10[105]	11[106]	12[21]	13[107]	14[108]	15[109]	16[22]	17[110]	18[34]	19[23]	20[111]	21[112]	22[113]	23[114]	24[35]	25[41]	26[19]
Techniques																										
Machine Learning																										
CRFs	×	×		×	×	×	×	×		×			×		×	×	×	×	×	×	×	×		×	×	
SVM	×						×			×					×				×		×					
Log. Regression																			×							
Max. Entropy													×													
Random Forests																				×						
Rule Based	×		×						×							×										×
Dictionary																										
Dictionary	×	×	×	×	×	×	×		×	×	×		×	×		×		×	×	×	×	×	×	×	×	×
Only Dictionary											×	×											×			
NLP																										
Tokenization	×			×	×	×	×	×	×	×	×	×	×		×	×	×	×		×	×		×	×	×	×
Sentence Splitting	×			×	×	×	×	×	×	×	×	×	×		×			×			×				×	×
POS tagger	×			×	×	×	×	×	×	×		×				×					×					×
Nomenclature rules	×		×	×	×				×		×			×							×			×		
Lemmatization	×			×	×	×	×	×								×			×							
Stemming					×		×						×		×	×		×		×	×					
Shallow parsing						×			×																	
External CNER	×	×	×	×	×				×		×	×		×		×					×			×	×	×
Post Processing	×	×	×	×	×	×	×		×		×	×	×	×		×		×	×		×	×	×	×	×	×

Table 2.2 shows that most of the presented systems are hybrid systems making use of dictionaries and machine learning approaches. It also depicts that CRFs were used by the majority of the participants. Out of the 26 participating systems, only 6 used SVM as the learning approach. Log regression and Max Entropy are used only by one system. Only two systems [36], [104] used solely rule based approaches, which lead them into third and ninth position in chemical NER rank. Two systems [113], [114] over three, which used only dictionary lookup approach using considerable databases and terminologies, could achieve satisfactory results (rank 11 and 12). Moreover all participating teams applied at least one of preprocessing tasks from NLP domain in their systems. Except for six systems (2, 8, 10, 15, 17, 20) all others, used post processing to improve the outcomes of NER systems. The following discussion presents the methodologies for most of the systems mentioned above.

Leaman et al. [25] implemented tmChem which achieved the highest performance. They employed a model combination approach to combine two different created models. The differences of their models are on the tokenization methods, feature sets, CRFs parameters, and post processing approaches. The first model is an adaptation of BANNER [84]. They used a finer tokenization method than BANNER's default that was tuned for gene or disease detection. CRF with order of 1 is used to train first model. To create the second model, they used CRF++ library [115] by repurposing a part of tmVar system for identifying genetic variants [86]. The order of CRF in second model is set to 2. After model creation phase, they combined models to get a final chemical NER system.

Table 2.3: Overview of used features by participating teams in ChemDNER task of BioCreative IV

	1[25]	2[98]	3[36]	4[99]	5[100]	6[101]	7[102]	8[103]	9[104]	10[105]	11[106]	12[21]	13[107]	14[108]	15[109]	16[22]	17[110]	18[34]	19[23]	20[111]	21[112]	22[113]	23[114]	24[35]	25[41]	26[19]
Word Level Features																										
Numerical/Digit	×	×		×	×	×	×	×					×				×	×		×	×			×		
Word Punctuation	×	×		×	×	×	×	×							×		×	×			×			×		×
Word case	×	×		×	×		×	×		×			×		×			×			×			×		×
N- gram	×	×		×	×	×		×					×		×			×			×			×		×
Word Morphology	×			×	×	×	×	×		×							×	×		×	×					
Word Patterns	×			×	×	×							×		×		×	×			×			×		
Word Length				×	×	×							×		×			×			×		×	×		×
POS	×			×	×	×	×	×		×						×					×					×
Special Character	×					×	×						×		×			×			×			×		
Whitespace	×	×			×			×							×			×								×
Other		×														×										
Lookup Features																										
Chemical lexicons	×			×	×	×	×											×	×		×		×			×
Stop Words																				×	×		×	×		×
Other		×											×										×	×		
Document Features																										
Mentions in training				×	×		×						×				×	×	×	×	×					
Multiple mentions			×																		×					
Other		×						×										×			×					

Lu et al. [98] implemented a system using CRFs model and word clustering features. To create the CRF model, they mixed word level and character level CRFs models. They also created clustering features using PubMed articles based on the one-level or multi-level clusters. Lowe et al. [36] implemented LeadMine as another NER system by combining the rule based and dictionary based approaches together. Most of the dictionaries used by LeadMine are automatically derived from publically available resources to identify trivial names. Also it encoded expertly curated rules to describe systematically named entities. Batista-Navaro et al. [99] developed ChER as chemical NER system by incorporating specialized preprocessing analytics and rich feature sets for machine learning in addition to post processing for abbreviation detection. Huber et al. [100] retrained ChemSpot [94] using other features derived from the output of individual components used in ChemSpot plus other chemical resources. Moreover they used outputs of OCSAR 4 [93] as input features. Campos et al. [101] developed a supervised learning based method to extract chemical compounds from given documents. Their proposed system uses a rich feature set such as linguistics, orthographical, morphologic, and dictionary matching features. They developed a system using two frameworks: Gimli [116] for feature extraction and machine training and Neji [117] system for post processing. Tang et al. [103] implemented another machine learning based system using CRFs and SSVM [66] and different sets of features including orthographic, morphologic and domain knowledge features. Furthermore, they used word representation features including Brown clustering [1118], random indexing [119], and skip-gram [120]. Another chemical entity recognition system is created by Munkhdalai et al. [103]. It incorporates domain knowledge from chemical and biomedical context with word representations. They extended BANNER along with presentation of semi supervised

learning method that efficiently exploits unlabeled data for entity recognition. The key feature of this method is learning of word representations from a vast amount of textual data for feature extraction. Cocoa [121] is an existing entity recognizer for the biological domain. Ramanan and Nathan [104] have adapted the output of Cocoa to detect chemical entities. At first, they trimmed the generic entity terms which were irrelevant to the chemical context and excluded them. Then they added dictionary entries to handle unusual entity names in the given abstracts. Zitnik and Bajec [105] proposed a novel NER system using different types of CRFs whose outputs are input to SVM classifier to combine. Irmer et al. [106] presented a system using a modular text processing pipeline. They integrated it with a number of modules into the OCMiner which is a pipeline for unstructured information processing based on the Apache UIMA framework [122]. Additionally, they made use of a kind of dictionary based method for the annotation of chemicals. Another hybrid system which combines dictionaries with a rule based approach is developed by Akhondi et al. [21]. Different number of available dictionaries including ChEBI [17], ChEMBL [122], ChemSpider [61], DrugBank [123], HmDB [124], NPC [125], TTD [125], PubChem [126], JoChem [18], and UMLS [57] are employed by this system to extract nonsystematic chemical entities. Xu et al. [115] designed a three step pipeline consisting of a preprocessing module, a recognition module, and post processing module. For the learning part of the recognition module they employed features frequently used in NER systems such as linguistics, character features, word shape, contextual features, and word representation features. Kumar et al. [109] developed a domain independent model creating three systems using CRFs and one using SVM. Then they combined the results of those systems. In the training phase they used domain independent feature sets without considering external resources related to the

context. Yoshioka and Dieb [22] implemented a classifier using the outputs of well-known chemical NER systems e.g. OSCAR 4 and ChemSpot along with some linguistic features such as POS. They showed that ChemSpot by itself is good at precision and in contrast OSCAR 4 is good at recall. Thus to take advantage of these two systems they fed the output of these classifiers as input feature to a CRF and created a new classifier. Named Entity Recognizer of Chemicals (NEROC) [110] is another NER system for chemical context. Its basic architecture is exactly the same as the system introduced in [22]. The only difference between two systems is the feature sets employed and the toolboxes utilized to create final systems; NEORC made use of more features compared to the system proposed in [22]; NEORC uses Mallet toolkit [127] whereas other one uses CRF++ [115]. Another ensemble approach is introduced by Khabsa and Giles [34] which is based on employing multiple classifiers and output probabilities that represent the confidence score for each entity. They used a modified version of ChemXSeer [128] along with ChemSpot and OSCAR 4 for the implementation of their approach. Ravikumar et al. [23] extended BioTagger-GM [129], a system for gene names detection, and MedTagger [130] a clinical related entity recognizer. They used three machine learning algorithms; CRFs, SVM, and logistic regression [131]. Then they combined the results of different systems and did some post processing for parenthetical alignment errors and removing false positives appearing in the train data. Li et al. [112] developed another kind of hybrid system combining the machine learning approach with hand crafted dictionary extracted from training data. They used CRFs with common orthographic and morphological features. In the dictionary based phase they tried to find entities from test data, which have been seen before in training data. Moreover, they did some post processing such as removal of wrapping brackets and

symbols appearing at the end. Shu et al. [113] participated in the task by implementing a system using CRFs as learning algorithm and common orthographical and morphological feature set. Additionally they used some linguistic features such as part of speech tags. Another dictionary based solution to ChemNER problem was DBCHEM presented by Ata and Can [114]. It is based on database queries for chemical/drug identification. They prepared a database with 145 million entries including chemical compound and drug names, their synonyms, and molecular formulas. They utilized PubChem Power User Gateway (PUG) [126] to create a database. Usie et al. [35] implemented CheNER as hybrid system. It uses CRFs with different types of features in addition to dictionary features extracted from dictionaries such as JoChem, PubChem, and ChEBI. Additionally it makes use of outputs of OSCAR 4 and ChemSpot for combination. Lana-Serrano et al. [19] proposed a rule based approach using semantic information for Chemical/drug entity detection by means of ChEBI ontology and the MeSH Meta thesaurus [60] to extract semantic information. Also they integrated MetaMap tool [96], ANNIE POS tagger and pharmacological databases such as DrugBank. Dai et al. employed machine learning approach with representative tag scheme and fine grained tokenization approach [123]. Most commonly used tag representation scheme for NER task is IOB2 [132], but they used IOBES scheme with combination of fine grained tokenization results. They implemented two types of tokenization in their task; Coarse grained tokenization where the standard Penn Treebank tokenization rules [124] are used, and fine grained tokenization where firstly coarse tokenization is applied on data then generated tokens are tokenized again through following two steps. First, insertion of separations before and after symbols like hyphens and dashes, second, separation at the locations between letters and digits, as well as at

character locations where a lower case letter is followed by an uppercase letters. Finally they combined that scheme with the fine grained tokenizer to use during their machine learning phase.

2.5.5 Publicly Available Chemical NER Systems

Recent research in text mining applications especially NER in chemical context resulted in a number of commercial or freely available tools and products for the task. ProMiner [125] is a commercial NER system that was originally developed for identification of genes and protein names in biological texts. It has later been optimized for chemical NER purpose. InfoChem developed a system namely ICANNOTATOR [126] which is able to extract chemical entities including trade, trivial and systematic names. Moreover it can detect standard chemical identifiers such as CAS register numbers or InChi. ChemAXon implemented D2S (Document to structure) [108], to identify chemical named entities in the documents with different formats such as PPT, PDF, DOC, and ODT. It can also subsequently map the recognized mentions to their structures. Peregrine [18] is a dictionary based publicly available tool applicable for chemical/drug NER task. Chemical Entities Recognition Skill Cartridge (CER) from the TEMIS (TextMIning Solutions) [133] allows users to load and use precompiled dictionaries for their chemical/drug NER tasks. In contrast to CER, Whatisit [134], an online text processing system which does not allow users to replace or modify underlying dictionaries. EBI developed another publically available web based system, EbiMed [134], to recognize the drugs and chemical named entities in Medline repository. GATE [135] (General Architecture for Text Engineering) is an open source text mining platform that provides customizable and re-trainable algorithms which can be used for chemical/drug entity mention recognition. SIIP [136] (Strategic IP Insight Platform)

gives an interactive platform for patent literature processing. It provides chemical annotation services by making use of combination of rules and dictionaries. In particular, it uses negative dictionary (i.e. a dictionary of terms that should not be identified as chemicals) to filter out the identified potential mentions. ChemFrag [137] is another rule based NER system from IBM which employs some rules for identifying organic chemical names. ChemBrowser [138] is a chemistry specific NER engine which has been implemented using hybrid approach, allows users to quickly merge different NER strategies together for a given solution. ChemicalTagger [139] is an open source NER system which uses OSCAR for recognition of chemical and drug compounds. ChemEx [140] is another open source system to facilitate the chemical data curation by extracting chemicals, organisms and assays from large collection of texts. It can also handle both text and image in documents. The text detective module uses some rules and dictionaries to recognize chemicals, genes, diseases, and organisms in documents. CYP34A introduced by Feng et al. [141], is a text mining engine for information extraction on chemical interactions. It is created based on dictionary approach in order to recognize chemicals.

However, no evaluation on the performance of mentioned systems has been made publicly available.

Chapter 3

MULTIPLE CLASSIFIER SYSTEMS

3.1 Introduction

One solution to increase the performance of the classification tasks is to create one classifier with the highest possible performance. In this case choosing the most suitable classification algorithm and fine tuning its parameters along with extracting the most discriminative and useful features are necessary to achieve the highest possible performance. However, because of the large number of classification algorithms and possible parameters, choosing the most suitable algorithm and finding the optimum values of the parameters is not always trivial. Moreover, selection of the best subset of features among very large number of features further complicates the process of finding the best classifier.

Another solution is to design a Multiple Classifier System (MCS) using a set of classifiers with relatively good performances. In this case, each input is classified by making a joint decision using all or a subset of the classifiers in the MCS. It has been shown that MCS or classifier ensemble is usually more accurate than the individual members of the ensemble in many applications [142], [143].

To create an MCS, each individual classifier is trained using training samples labeled with corresponding class labels. Training data is often enriched with discriminative features extracted from the given data itself or using external resources. Each

classifier maps the input vector **X** to a specific class, Y_i, among N possible class labels. Outputs of a classifier can be of the following types [144]:

- **Abstract**: the classifier output is a single label from the set of labels. This type of output is also known as hard or crisp outputs.
- **Rank:** each classifier produces a list of all possible classes ordered from the most likely class to the least likely. For example in the case where there are 3 possible class labels, for a given input sample the output would be $y = \{y_3, y_1, y_2\}$, where the class label in the first position is ranked as 1, the class label in position two is ranked as 2 etc.
- **Measurement**: In this case, output for each input data is a list of all possible classes with their corresponding confidence score computed by the classification algorithm. As an example consider the 3 class problem; for a specific input sample the output of the classifier may be y= {0.24, 0.44, 0.22}, which shows that the probability of the given sample being from class y_1 is 0.24, from y_2 is 0.44 and finally for being from y_3 is 0.22.

Joint decision of an ensemble of individual classifiers is computed using the types of outputs according to the combination scheme employed. The prevalent combination schemes used in MCSs are explained in the next section.

One of the most crucial decisions in MCS framework is classifier selection for the ensemble which refers to deciding which classifiers should be employed in the ensemble making the joint decision. Clearly classifiers with very poor classification performance are not expected to be very useful in making aggregate decision. In the same way, combining a number of similar classifiers is not expected to bring any benefit. The classifiers that are to be included in an MCS should not be identical. In

other words they should not make similar classification errors. This property is called diversity. Diversity in the regions where classifiers make mistakes is an intuitive criterion that can be used to choose which classifiers will be added to the ensemble to improve the classification performance. There are several approaches to achieve diverse classifiers:

i. Using different classification algorithm such as HMMs, SVM, CRFs, Decision Tree, etc.
ii. Using the same classification algorithm with different values for parameters.
iii. Training various classifiers using different training data or subsets of training data instead of whole data set.
iv. Using different types or combinations of features.

Designing an MCS involves decisions on the architecture of the base classifiers; the type of outputs that will be combined, the selection criteria used to choose the base classifiers, D_i, from the repository and the fusion function f such that $D(\mathbf{x}) = f(D_1(\mathbf{x}), D_2(\mathbf{x}), \ldots, D_L(\mathbf{x}))$, where, $D_i(\mathbf{x})$ is the prediction of i^{th} classifier given input $\mathbf{x}$.

3.2 Criteria Used for Classifier Selection

Selection of the most suitable classifiers from a pool of all candidates is one of the most important steps in designing an MCS. In general, classifier selection methods can be categorized under two groups: Static Classifier Selection (SCS) and Dynamic Classifier Selection (DCS). In SCS, the same set of classifiers is used to predict all unlabeled samples. On the other hand, in DCS, a different set of classifiers may be used to produce the joint decision for each unseen pattern. The objective of both strategies is to achieve the highest possible classification performance. In SCS

approach to find the optimum subset of classifiers, after training the base classifiers on training data, combination results using development data are used to select the members of the ensemble. The ensemble with the highest performance on the development data is used to classify unseen data. In the cases that there is no development data, n-fold cross validation [145] can be employed as an alternative. In DCS, the classifiers that will participate in the joint decision is determined dynamically based on the performance of the classifier on the similar input values in training data [144].

The classifier selection criteria employed has great impact on the final performance of MCS. The simplest, most intuitive approach is to consider the individual performance of the base classifiers such that *k* top performing classifiers are selected for the ensemble. However as mentioned earlier, if selected classifiers make the same classification mistakes, combining them will not improve the overall performance. Therefore performance of individual classifiers alone cannot be a good selection criteria.

An ideal ensemble includes highly performing classifiers which disagree with each other as much as possible [146]. Disagreement among classifiers refers to making mistake in different regions of input data and it is known as diversity [144], [147]. A number of diversity measures have been proposed as classification selection criteria in [144] including: Q statistics, Correlation, Disagreement measure, generalized diversity, and double fault measure.

However most of the studies have shown that there is not a strong correlation between the performances of combined classifiers in an ensemble and diversity

measures [148], [149], and [150]. Another intuitive approach to choose diverse classifiers is to include classifiers with different success on various cases.

The performance of the ensemble can also be used as a selection criterion. The problem with this case is the cost of exploring the solution space containing all possible subsets of candidate classifiers, especially when the number of classifiers is large. Another difficulty in this case is the need for development data in addition to unlabeled test data to evaluate the ensemble's outcome. Using n-fold cross validation or reserving a predefined portion of training data as development data are two possible solutions to the problem.

Besides the methods mentioned previously as selection criteria, there are two popular methods, namely Bagging [151] and Boosting [152] which are also employed to improve diversity among classifiers by altering the training data seen by each individual classifier. Bagging approach trains each individual classifier of the ensemble using bootstrapped samples of training data with replacement. On the other hand Boosting methods adapt the selection probability of samples over the time, based on performance of the most previously created classifiers such that the samples misclassified by the previous classifiers are more likely to be selected for training new classifiers.

3.3 Search Algorithms used for Classifier Selection in MCS

In order to choose the most optimum ensemble of classifiers which produce the highest classification accuracy, the search space containing all the possible combination of individual classifiers must be explored. Different search methods are available for this purpose. Greedy search solutions attempt to systematically find the

optimal ensemble by considering the effect of adding specific classifier into or removing it from the ensemble. Evolutionary search strategies on the other hand, form a population of ensembles and perform optimal search by evolving the population and selecting the ensemble with the highest performance. The most frequently used approaches to find an optimal ensemble are presented in the following subsections.

3.3.1 Single Best (SB)

The simplest and most straightforward approach to classifier selection is to select the highest performing classifier among all candidates. In MCS, usually single best classifier is used as a benchmark to compare the performance of the ensemble produced.

3.3.2 N Best (NB)

In this case *N* classifiers that have highest performance are selected from the classifier pool. This method is also computationally cheap and it merely involves calculating the performances of individuals and sorting them in ascending order. But it does not take diversity and joint performance into account.

3.3.3 Forward Selection (FS)

Forward Selection [153] is a greedy search algorithm that produces an ensemble by recursively adding more classifier to the initial ensemble that usually contains only one classifier, the single best individual classifier. At each iteration, a new classifier is selected from the pool and added to the candidate ensemble. If the performance of the new ensemble is superior to that of the former ensemble, new classifier is added to the ensemble otherwise it is discarded. The process continues until all classifiers in pool are considered for addition to the ensemble.

3.3.4 Backward Elimination (BE)

Backward Elimination [153] is another type of greedy search algorithm. In contrast to the FS algorithm, it starts with a target set containing all members of the classifier pool. Then in each iteration, the weakest performing classifier is removed from the set with the aim of improving the performance of the ensemble. If removing a classifier results in any improvement on overall performance of ensemble, the classifier is eliminated, else it remains in the ensemble. The elimination process is continued until all classifiers are examined.

3.3.5 Evolutionary Algorithms

Given a large number of classifiers in repository, an intelligent or evolutionary classifier selection process, rather than a greedy approach, becomes a crucial issue in MCS design process due to the size of search space. Successes of evolutionary algorithms in classifier selection process have been shown through several studies on different applications [154].

Population based evolutionary algorithms have been more prevalent in classifier selection process. Genetic algorithms received more attention among all others in this area [154], [155]. There are also other algorithms that can be used beside the genetic algorithm such as Particle Swarm Optimization (PSO) [156], Tabu Search [157], and Bee Colony [158]. In this thesis PSO is employed as part of the designed ensemble approach. PSO is an evolutionary algorithm that is applicable to vast amount of problems. In addition to being computationally simple, PSO has powerful search mechanism based on bird flocks principle [156]. In this algorithm, a population of individuals is evolved toward the solution space of an optimization problem by means of the evolutionary operators, which produce new solutions from the current populations. Through the evolution process, at the end of each iteration,

candidate solutions are evaluated. If the population does not contain the optimal solution, evolution continues until an optimal solution is found. Even though PSO in its original version was proposed to solve real-valued problems, it is also applicable for discrete issues. The Constriction Factor Method (CFM) which is a modified version of the basic PSO, shows better convergence properties in comparison to the basic algorithm [32], [159], [160, [161], [162]. Hence, CFM version of PSO was employed in this thesis.

3.4 Combination Methods used in MCS

Classifiers selected for ensemble should be combined in some way in order to produce a final joint decision. Different combination approaches are applicable at the aggregation stage of a MCS, depending on the output type of individual classifiers. Some of the most commonly used combination methods are described in the following discussion.

3.4.1 Majority Voting Method

Majority Voting [144] is the most commonly used method in general voting category. It takes the abstract outputs of all classifiers in an ensemble as input and determines the label which received the majority of votes.

Weighted majority voting approach is a type of voting method where a weight that reflects accuracy or reliability of each classifier for its predictions is utilized. In this case weights are considered during aggregation of votes by increasing or decreasing the impact of predictions of individual classifiers. In the weighted majority approach, the weights used can be taken as a constant weight for each classifier or each individual classifier can have a different weight for each possible class based on the

strength of that classifier for predicting samples from each class. The latter is then referred to as class-based weighted majority voting.

In addition to the majority strategy, voting approaches can also be applied in unanimity or plurality forms [144]. In the unanimity voting approach, the ensemble's output for each pattern is labeled as a specific class, if almost all members of the ensemble agree on that label. On the other hand in the plurality mode, only the half number of members plus one need to agree on a class label.

3.4.2 Algebraic Combination Methods

If the outputs of classifiers are from measurement type, algebraic combination schemes including Sum Rule, Product Rule, Max Rule, Median Rule, and Mid Rule can be employed to combine the members of the selected ensemble [144]. These rules are easy to implement. For example in the case of Sum Rule, summation of scores of all classifiers for each class label should be calculated for a given input pattern. Then the class label with the highest score is denoted as the decision of the ensemble. For the product rule the approach is similar to Sum Rule; but instead of summation, product of scores is used to calculate final decision. For Max, Median, and Mid Rules the given input pattern gets a label with the maximum measurement achieved using maximum, median, and mid of all scores of classifiers in ensemble respectively per each class label.

3.4.3 Naïve Bayesian Combination Method

Using Naïve Bayesian approach for uncertain combinations helps in understanding the differences between individual performances and in the case of limited training data, incorporates some sort of existent prior knowledge about their abilities [163]. The assumption is that for a given class label, classifiers are mutually independent

(conditional independence). Here, $P(S_i)$ denotes the probability that sample **x** is labeled by classifier E_i as belonging to class $y_{S_i} \in$ class list (predicted label for sample **x** with E_i is y_{S_i}). The result of conditional independence is shown in equation (3.1).

$$P(\overline{\boldsymbol{S}}|y_k) = P(S_1, S_{k2}, \dots, S_L|y_k) = \prod_{i=1}^{L} P(S_i|y_k) \quad (3.1)$$

where L shows the number of classifiers and y_k represents the class k from the label set. Then, the required posterior probability for labeling **x** is shown as equation (3.2).

$$P(y_k|\overline{\boldsymbol{S}}) = \frac{P(y_k)P(\overline{\boldsymbol{S}}|y_k)}{P(\overline{\boldsymbol{S}})} = \frac{P(y_k)\prod_{i=1}^{L} P(S_i|y_k)}{P(\overline{\boldsymbol{S}})}, \text{ k=1, 2, … , N} \quad (3.2)$$

Since the denominator is not based on the y_k, this part is ignored; as such, calculation of the support for class y_k is done in the following manner, shown in Formula (3.3):

$$\mu_k(x) \propto P(y_k)\prod_{i=1}^{L} P(S_i|y_k) \quad (3.3)$$

To calculate a $N \times N$ confusion matrix CM^i for each classifier, E_i, the results of applying it on development data set should be observed. Here N represents the number of the classes. The $CM^i\ [k][\ S_i]$ of this matrix, CM^i_{k,S_i}, shows the number of data elements which originally had the class label of y_k, and were assigned to class y_{S_i} through E_i. N_j Determines the total number of class y_j samples. Taking $\frac{CM^i_{k,S_i}}{N_k}$ as the probability estimate $P(S_i|y_k)$, and $\frac{N_k}{n}$, n shows the total number of samples, as a prior probability for class y_j, equation (3.3) can be modified as equation (3.4).

$$\mu_k(\boldsymbol{x}) \propto \frac{1}{N_k^{L-1}}\prod_{i=1}^{L} CM^i_{k,S_i} \quad (3.4)$$

In equation (3.3), if zero is taken as the estimate of $(S_i|y_k)$, it automatically nullifies $\mu_k(\boldsymbol{x})$, and the other estimate values are not taken into account. Titterington et al. [166] proposed a novel formula for Naïve Bayesian combination as a solution for this problem which is given in equation (3.5):

$$P(\overline{\boldsymbol{S}}|y_k) \propto \{\prod_{i=1}^{L} \frac{CM_{k,S_i}^{i} + \frac{1}{N}}{N_k + 1}\}^{B} \quad (3.5)$$

Where *B* is a constant, which should be determined individually for various classification tasks. Titterington has proposed values 1, 0.8, or 0.5 for *B*. Formula (3.5) has been used as the combination strategy for the selected classifiers in this study.

Chapter 4

CLASS IMBALANCE PROBLEM

4.1 Introduction

Classification is a popular and important task of pattern recognition which aims to map a given input data into one of predefined class labels. There are many algorithms developed for different classification problems. Generalization ability of a classifier is the judgment measure for its performance which is usually demonstrated by error rate or overall accuracy.

Classification algorithms mostly assume that misclassification rates for different classes have the same cost and treat them equally during the learning phase. In these cases the learning process is done by aiming at achieving maximum overall accuracy [165], [166]. However this is not the case for all real world applications. There are plenty of problems that have unequal cost of misclassification for individual classes, such as fraud detection in banking transactions, telecommunication risk management systems, fault prediction in software engineering [167] etc.

Named entity recognition in biological and chemical context also suffers from equal error rate phenomenon discussed above since the classification algorithms treat the named entity samples and non-entities equally Thus they are favored in the recognition of samples from classes which are in the majority class. For example, in the chemical NER problem an entity of interest is much less likely to occur in text

than non-entities. On the other hand rare samples from specific classes are usually costly and more important.

The Class Imbalance Problem (CIP) is when number of samples from some classes are much more than the number of samples from other classes. . The direct impact of this problem on the chemical NER task has caused to dedicate an individual chapter to the CIP and involving strategies. Additionally, related studies which tackled the CIP in NER context are represented at the end of the current chapter.

4.2 What is the Class Imbalance Problem (CIP)?

Class imbalance problem refers to the category of classification problems in which the number of samples from some classes is much more than samples belonging to other classes i.e. skewed class distribution. In such conditions standard classifiers usually tend to be favored by the classes related to majority of samples and ignore others. Since classifiers mostly focus on overall accuracy, this leads to lower performances resulted from ignoring the different kinds of classification errors [165]. For instance, consider a classifier that wants to learn from a data set containing 95 samples with a specific class and just 5 samples from the second class. If all data are labelled as the first class, overall accuracy would still be 95%. Hence, here accuracy will not be an appropriate evaluation metric when CIP is an innate characteristic of data.

In practice CIP is addressed with binary classification problems where multi class problems are usually translated to a sort of two-class problems. As the samples from rare class are of greater interest than those samples that belong to the other, the minor objects are referred as positives and majority samples are known as negatives.

Another important concept related to the CIP is the ratio between the negative samples (N_{maj}) and positives (N_{min}) which is known as Imbalance Ratio (*IR*), Imbalance Rate (*IR*), or Imbalanced Degree (*ID*). Normally *IR* is computable according to equation (4.1):

$$IR = \frac{N_{maj}}{N_{min}} \tag{4.1}$$

There is no standard or agreement about the exact value for the imbalanced degree required for a data set to be considered as "*imbalanced*". However, most professionals agree that a data set with *IR* value around 10 would be modestly imbalanced, and a data set with *IR* value above 100 is extremely imbalanced [168]. There are two more subtle points about *IR*. First, class imbalance should be defined with respect to a specific data set and since the class labels for test data are not known, imbalance ratio is calculated based on the distribution of samples within the training data. The second point is concerned about the actual size of training data used for *IR* calculation. It is important to know that the CIP for a data set with 20,000 positive samples and 2,000,000 negatives is quite different from a data set with 20 positives and 2000 negative samples, even though the IRs are same.

CIP can be categorized into two types: '*between classes*' imbalance and '*within classes*' imbalance. Between classes imbalance points outs to the class distribution only, such that some resampling techniques can help to decrease its effect on the classification performance [169]. In the within classes case, samples from the minor class are included in some sub clusters, where samples that belong to some of the clusters are very rare in comparison to the members of other clusters [170]. Between

classes imbalance is also known as the *rare class* problem, while within class is referred as the *rare case* problem [171].

Many solutions have been proposed to deal with the CIP at both data and algorithmic levels. Data level approaches include some preprocessing tasks on training data aiming to achieve a balanced set as much as possible. Resampling techniques including oversampling and undersampling belong to this category. On the other hand, algorithmic solutions try to enforce the learning process with respect to rare classes. One class learning and cost sensitive classification are two mostly used solutions at this level. Another strategy is the ensemble learning strategy which has shown great success in general classification tasks. Detailed information on each strategy is given in the next section.

4.3 Solutions to CIP

Generally speaking, many studies have been carried out to deal with the CIP in different application domains. All approaches can be categorized in one of the three categories discussed below: resampling, algorithmic and ensemble methods. In the following subsections the most widely used and popular methods for each category are presented.

4.3.1 Resampling Techniques

Resampling methods are data-level strategies which aim to adjust the distribution of the training data. Resampling can be done either by oversampling or undersampling approaches. Oversampling tries to increase the number of positive samples to balance the majority and minority classes. Likewise, undersampling eliminates some negative samples until the data set becomes relatively balanced. In both approaches the data sets obtained after resampling is composed of almost balanced numbers of

samples belonging to the minor and major classes. However, both mentioned techniques have their own drawbacks. Oversampling by duplicating exact copies of positive patterns can lead to overfitting [172]. By removing negative samples using undersampling, it is possible to lose some potential useful information for recognition purpose which may rely on removed samples [173].

4.3.1.1 Oversampling

The simplest type of oversampling is Random Oversampling (ROS) which replicates randomly selected positive samples in the given training data until a desired degree of class distribution is achieved. Synthetic Minority Oversampling Technique (SMOTE) presented by Chawla et al. [174] creates synthetic positive examples instead of merely copying existing positive samples in the given data set. In this method newly generated samples will be added to the current data. To create a new positive sample, SMOTE first selects a positive sample randomly and finds its K nearest positive neighbors. The distance between selected sample and each one of its neighbors is computed next and the difference vector where each dimension is multiplied by a random value in [0, 1] is added to the selected positive example. Result of this summation is added to the data set as a new synthetically created positive sample. Borderline SMOTE [175] is a modification of SMOTE where only borderline samples are considered to be used in SMOTE. It assumes that the instances near classification boundaries (border line instances) are more likely to be misclassified and thus they are more important. If most nearest neighbors of a positive sample belong to the negative class, then that border line sample is treated as a sample likely to be misclassified. Another development on the basic SMOTE is Adaptive Synthetic sampling (ADASYN) [176]. It creates positive samples adaptively according to their distributions where for difficult positive samples more

synthetic samples is created in comparison to easy ones. Difficulty or easiness of samples determines by the neighborhood of positive samples.

4.3.1.2 Undersampling

Random Undersampling (RUS) is a non-heuristic approach that removes negative samples from the data set randomly to achieve a preset balance ratio between negatives and positives. However, it may also cause the elimination of useful information from the negative samples. Tomek links introduced by Tomek [177] is a commonly used undersampling approach where given two examples *a* and *b* from different classes, *a* from positive class and *b* from negative, the (*a*, *b*) pair is called the Tomek link if there is no such example *c*, such that *d* (*a*, *c*) < *d* (*a*, *b*) or d (*b*, *c*) < *d* (*a*, *b*), where *d* (*a*, *b*) defines the distance between *a* and *b*. In such a case, the negative samples from the link is removed. The main drawback of this method is the high computational cost of finding Tomek links among samples [176]. Condensed nearest neighbor (CNN) [178] and One Sided Selection (OSS) [179] can also be used for undersampling purpose. CNN aims to find samples away from decision boundaries where the main idea is to find a subset of the training data where all samples could be correctly predicted by using 1-KNN in the found subset. One sided selection uses CNN to remove redundant negative samples first. Subsequently the Tomek links method is applied on the obtained set to remove borderline and noisy negative samples. Edited Nearest Neighborhood (ENN) [180] removes samples where label is different than at least is differ than two of its 3 nearest neighbors. Neighborhood cleaning rule (NCR) [181] removes the negative samples that are misclassified by 3-NN. Meanwhile if positive samples are miss-predicted by 3-NN, its negative neighbors are removed.

4.3.1.3 Hybrid Techniques

In order to take advantages of undersampling or oversampling methods together, some techniques, which combine both strategies, have been developed. By means of combined methods it is possible to make data sets balanced without neither losing too much useful information nor suffering from overfitting. For Example SMOTE+ENN and SMOTE+Tomek are two successful hybrid methods presented in [182].

4.3.2 Algorithmic Techniques

These approaches try to adopt classification algorithms to enforce the learning with respect to the rare classes. The main idea is to adjust the bias of the classifiers internally [183]. Being algorithm- specific is the main weakness of these approaches. Most of these methods involve applying search strategies that are well suited for detecting rare samples in data when common patterns are abounded.

4.3.3 Ensemble Learning

Ensemble learning is one of the major approaches to deal with CIP in different applications. It allows individual classifiers to emphasize the positive class regions differently and take advantage of their combination to decrease the risk of overfitting. The main aim of existing ensemble techniques is on how to rebalance the training data for each base classifier and how to create the cost sensitive ensemble. Bagging and AdaBoost [184] are two commonly used ensemble methods in this area. It is important to know that applying the original Bagging algorithm for imbalanced class distribution, is not useful since every bootstrapped subset of training data will still be imbalanced. Furthermore, it could be even more favored on the negative class compared to the original data set [185]. The simplest way is to correct the skewness of data in each subset using sampling methods and build the

base classifiers from data with more balanced class distribution. Some representative ensemble methods for imbalanced data based on Bagging approach are: AB-SVM [186], Bagging Ensemble Variation (BEV) [186], Easy Ensemble and Balance Cascade method [187]. AdaBoost builds classifiers sequentially with subsequent classifiers focusing on misclassified samples from training data by former classifiers. Therefore, it can be considered as an accuracy oriented algorithm since it emphasizes misclassified samples. Also it treats all classes equally such that classifiers with higher accuracies will receive higher weights. For imbalanced data, since rare samples contribute less to the overall classification accuracy, learning strategy will tend to bias towards the majority class. Therefore the original AdaBoost will not work well for recognizing rare cases. For CIP cases, AdaBoost can evolve in two ways; first integration with resampling techniques and second cost sensitive boosting [169]. For the first one some methods such as SMOTEBoost [188], JOUSBoost [189] are developed. For the second approach AdaCost [190] or RareBoost [191] can be listed as developed methods.

4.4 CIP in Named Entity Recognition

Named Entity Recognition, being a classification task also suffers from the class imbalance problem. As mentioned earlier, the main aim of NER is to find mentions of interests in the given text while the rest of the document is not as significant. Thus, entity mentions are considered as positive samples and other text segments as negatives. Since in any given text the number of positives is much less than the number of negatives, CIP poses to be a natural problem for NER. The number of studies carried out to investigate the CIP particularly in the NER context is very few. The simplest way to deal with CIP in the context of NER is the stop word filtering method. Since usually stop words in a language do not carry much useful

information, and due to the fact that usually they are not entities of interest, removing them from the given data the number of negative samples can be decreased. The main drawback of this approach is lowering precision through increasing the number of False Positives. Removing stop words which do not belong to entities and keeping the ones within the entities from the training data causes the classifier to learn and predict every such token in test data as positive thus making a false positive prediction. Massimiliano et al. [192] introduced the instance filtering approach where firstly the usefulness probability of tokens in text is calculated and tokens with lower probability are eliminated. They applied their method on both training and test data to decrease the time needed for both learning and generalization processes by downsizing the data sets. The possibility of losing positive samples as well as useful negative ones in this approach exists. Maragoudakis et al. [193] applied Tomek Link method on training data to decrease the number of negative samples. Tomanek and Hahn [194] used an altered version of active learning method to reduce the number of negative through learning phase.

Chapter 5

PROPOSED FRAMEWORK

5.1 Proposed System Architecture

The presented framework relies on combination of three individual concepts in machine learning which lead to an increase in classification performance. The tokenization strategy is considered firstly as the data preprocessing step. A more effective rule-based tokenizer, ChemTok [28], is designed for this purpose. The second investigated issue is the effect of class imbalance on the overall performance of classifiers for the NER task. A novel undersampling method, balanced undersampling, is implemented for this purpose. The main idea behind the proposed method is to maintain the presence of negative samples around positive ones as much as possible while achieving a more balanced negative-positive ratio to provide more information to the learning algorithms about the neighboring samples. Finally the effect of combination of an ensemble of classifiers for improving the quality of the resultant NER system's performance is considered. A new ensemble scheme using particle swarm optimization for classifier selection and Naïve Bayesian approach as the combination method is proposed in this thesis. The underlying architecture of the proposed approach incorporates all three novelties together and is shown in Figure 5.1.

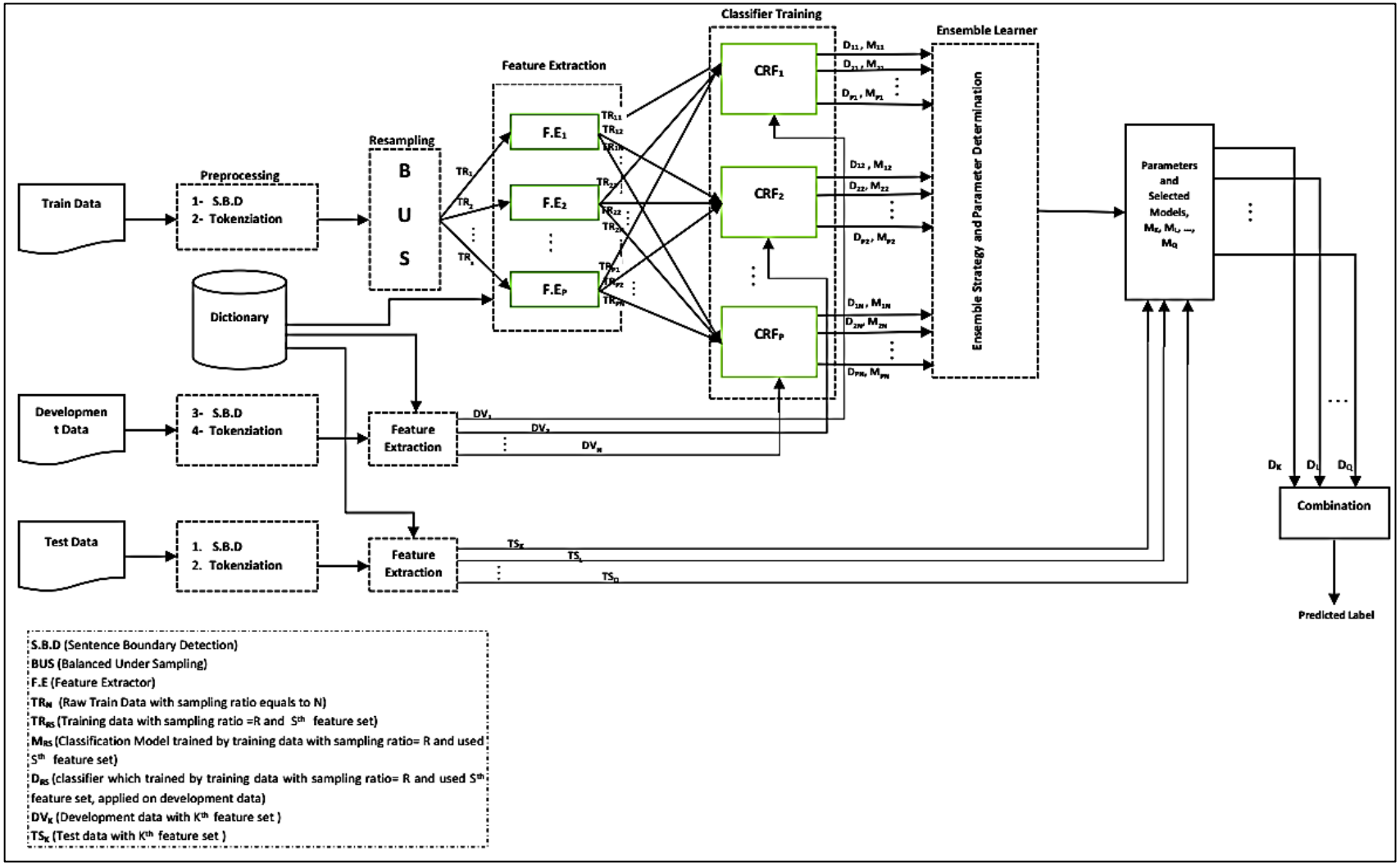

Figure 5.1: Proposed System Architecture

As depicted in Figure 5.1, there are five main modules in the proposed system including: preprocessing, resampling, feature extraction, classifier training and ensemble learning. The preprocessing module is responsible for preparing the given data for the subsequent steps. In this module two main operations are performed on raw data: separation of individual sentences by detection of their boundaries and tokenization using the proposed tokenizer, ChemTok. More details on this module are given in section 5.3. As mentioned in Chapter 4, class imbalance is a natural characteristics of data used in NER problems. Hence, to decrease the effect of the imbalance problem the output of the preprocessing module is passed on to the resampling module. This module resamples the data at various ratios and makes it available for the next step. The proposed BUS utilized in this framework is presented in section 5.4. After undersampling the training data, different number of features is extracted using the feature extraction module. Commonly used feature sets in the NER domain plus some domain specific features using external dictionaries are created using this module. Section 5.5 gives detailed information about all features used. Using different training data with various feature sets from previous stages as well as different classification parameters, CRFs classifiers are trained in the next step. Details of classifier training are given in section 5.6. Subsequently, the proposed heuristic ensemble scheme is utilized using the classifiers trained in the previous step on development data. The ensembling module is designed based on the constriction factor method version of PSO for classifier selection and Naïve Bayesian method for classifier combination. This module is explained in section 5.7. After training the ensembling module using development data and selecting the subset of classifiers, the selected subset is tested using test data.

5.2 Data Used

In this thesis the ChemDNER corpus released for chemical/drug NER task at the BioCreative IV [45] event is used. ChemDNER task of BioCreative IV focused on the detection of mentions of chemical compounds and drugs, in particular those chemical entity mentions that can subsequently be linked to a chemical structure [20]. The entity mentions, which exist in the data sets mentioned, belong to the one of 7 different classes: *"ABREVIATION", "FORMULA", "IDENTIFIER", "SYSTEMATIC", "TRIVIAL", "FAMILY", and "MULTIPLE"*. Hence, other tokens within the text, which do not belong to any of the entity mentions above, are considered to belong to the "OUT" class. In this study we labeled all tokens that do not belong to entities as "OUT" and those tokens belonging to one of mentioned classes are labeled as "CHEMICAL". This conversion also has been applied by some participants in the task, particularly by the first ranked system, tmChem [25].

The ChemDNER corpus is currently the most comprehensive publicly available chemical related data set for the NER task in the chemical domain. The corpus consists of three individual parts; training, development, and test data sets. The train and development sets contain 3500 abstracts each, and the test data set contains 3000 abstracts. Table 5.1 shows the details of each data set. All sets include raw abstracts and annotation files listing each named entity together with its exact position in the corresponding abstract using character offsets. Tag representation scheme used in this thesis is IOB2 [132], which is the commonly used labeling method for the NER in biomedical field. Here, instead of labeling whole token as an entity mention, we make use Beginning (B), Inner (I), and Outside (O) labels to tag the token as follows:

'B' is used for the first token starting a mention, 'I' is used for any further token in the mention, and 'O' is used for all other tokens not part of any mentions. The last two columns of Table 5.1 show the ratio of negative samples (number of tokens from OUT-class) to the number of positives (B-CHEMICAL tokens and I-CHEMICAL) for the training data in the form of average (column C_9) and maximum imbalance ratio (*IR*) (column C_{10}). Considering these ratios, it is clear that the given data is highly imbalanced.

Table 5.1: Statistics of ChemDNER Corpus

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Train	3500	584	30418	3651	8520	899343	102228	41009	24.42	113
Development	3500	593	30445	3701	8677	893180	101724	40129	-	-
Test	3000	522	24655	3514	7563	772847	94871	30820	-	-

[***C_1:*** *Number of total documents for each data set,* ***C_2:*** *Number of documents w/o entities,* ***C_3:*** *Number of all sentences,* ***C_4:*** *Number of sentences w/o entity,* ***C_5:*** *Number of unique chemicals,* ***C_6:*** *Number of tokens excluding sentences w/o entity,* ***C_7:*** *Number non entity tokens for sentences w/o entity,* ***C_8:*** *Number of tokens belonging to an entity,* ***C_9:*** *Average Negative-Positive (imbalance) ratio,* ***C_{10}:*** *Maximum Negative-Positive (imbalance) ratio.*]

5.3 Data Preprocessing

Data used for NER needs to be in proper format for the subsequent classifier training phase. Since the given input samples for NER process are sentences, the first step is sentence boundary detection. Then, all sentences must be tokenized into segments which can be used as samples. Data resampling and feature extraction are also important steps of data preprocessing, but because of the special significance of these tasks in our proposed framework, we studied them in separate sections.

5.3.1 Sentence Boundary Detection

Sentence boundary detection is a NLP related task which decides where sentences begin and end. Most NLP applications require that the input text is divided into

sentences. Named entity recognition task also needs texts to be segmented into sentences. However, often this task is challenging because of the presence of punctuation marks. For instance, even though periods are normally used to show the end of a sentence, they may also be used as a decimal points, and ellipsis or as part of abbreviations and email addresses. Based on statistics, about 47% of the period marks refer to the abbreviations in Wall Street Journal corpus [195]. Therefore considering period marks alone is not sufficient to correctly identify the sentence boundaries. Excessive use of punctuation marks in biological and chemical related texts further exacerbates the problem. We used the sentence detector module from Apache OpenNLP [196] in our study to separate individual sentences. This module uses the following rules to identify the end of sentences: i) if the end character is a period, it ends a sentence, ii) if the preceding token is in the hand compiled list of abbreviations, then it does not end a sentence, iii) if the next word after a period is capitalized, it ends a sentence.

5.3.2 Tokenization with ChemTok

Tokenization is the most important basic step for NER process using machine learning strategies where the given raw text is broken into tokens. Tokenization approaches can vary depending on the context [197]. Breaking text into white space separated segments, known as white space tokenization, can be thought as the simplest tokenization method which may be acceptable in newswire domain. However, in some other contexts such as biological, chemical, or drug development science, segmentation of text only by white spaces is not appropriate due to the inconsistent use of spaces, variety of the nomenclatures utilized in the field, presence of punctuation marks inside named entities, nonstandard orthography, existence of ambiguous punctuation etc. [48],[198]. During this study we developed an effective

rule based tokenizer, ChemTok [28], which utilizes rules extracted from training data. The main innovation of ChemTok is the use of the extracted rules with the aim of merging the tokens formed at previous text breaking steps. Thus it can produce longer and more discriminative tokens. We have shown in our experiments that ChemTok outperforms the performance of two state-of-the-art tokenizers in the domain, tmChem [25], and ChemSpot [94]. Figure 5.2 presents the algorithm of ChemTok. The algorithm simply tokenizes raw text at white spaces in the first step. Then, two lists are utilized in the second step. The first list contains domain specific affixes such as 'Hyper', 'Anti', 'Amino', constructed from external sources listed in [199], [200].

The second list includes all chemical entities from ChemDNER training data. If a given token contains a substring that is found in the first list, then the token is segmented at the corresponding affix boundary. For example the tokens '*Antiherpetic*', '*hyperinsulinaemia*', '*Aminoacid*' are split at this step. These conjoined tokens are separated into two tokens since these tokens can also be used separately as part of NEs. After this step, the second list is used to decide whether a token should be considered for further tokenization or not. If the token is found in this list, it is assumed that the NE boundaries are correctly segmented and no further tokenization is required. The tokens not found in the second list may be further tokenized at different delimiters such as Greek letters, punctuation marks, and case changes of alphabetical characters if they exist within the token. Following this step, recombination rules are applied on the tokens resulting in from previous steps in order to generate longer and thus more discriminative tokens. Table 5.2 illustrates the rules with an example for each.

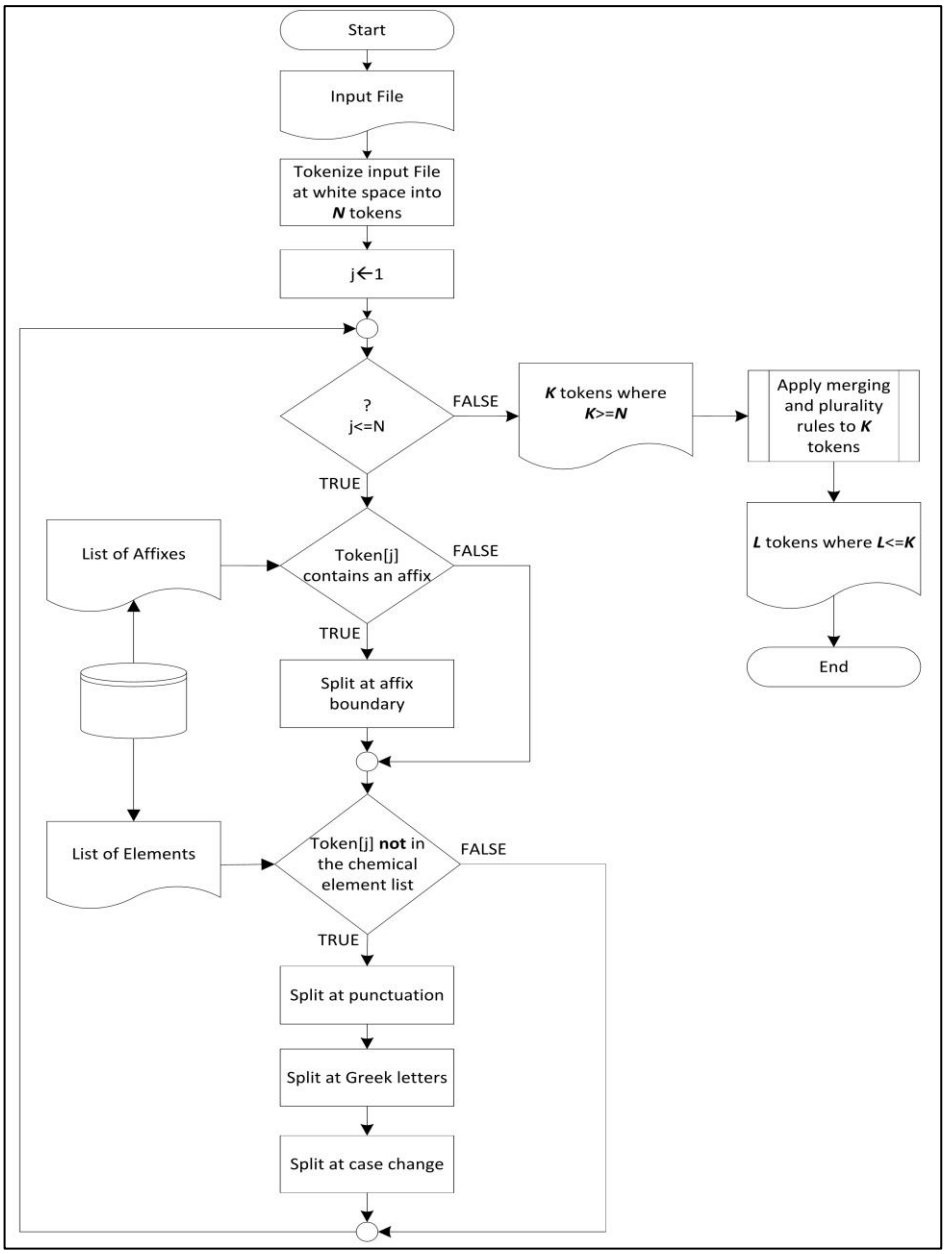

Figure 5.2: ChemTok Algorithm

Rule 1 is for merging the tokens that were split incorrectly at punctuation marks. Rule 2 combines the balanced containers around digits into the token, which is crucial for the recognition of formula entities in chemical domain. Rule 3 is used because previous step splits all words that start with uppercase, followed by sequence of lower cases including the common English words such as the ones which appear as the first word in a sentence. According to Rule 4, the list of known chemical names containing chemical compounds, basic chemical elements, amino acid names

and amino acid chains [199],[201] is used to merge tokens. Then all tokens in plural form are broken into two tokens: one token for the base form of the word and one token for the plurality suffix such as ‘s’, ‘es’ or ‘ies’. Representing the chemical entities in their base form makes recognition task easier.

In addition to the suitability of ChemTok for chemical texts, our experiments showed that it can also be used successfully on other biomedical domains using different data sets [28].

Table 5.2: Rules used in Step 3 of the ChemTok Algorithm

Rule no.	**Rule Explanation**	Example	
		Tokens after Step 2	**Merged Token**
1	Numeric tokens which are separated by ‘.’ or ‘,’ or ‘/’ or ‘-‘ or ‘_’are integrated into a single token.	125 , 12 , 12	125,12,12
2	If concatenated tokens from Rule 1 are surrounded by balanced containers such as parentheses, braces, and brackets, both container tokens are conjoined into the token.	(1-3)	(1-3)
3	Single uppercase tokens which are followed by sequence of lowercase letters as the next token are re-combined to a single token.	C ommon	Common
4	If the concatenation of consecutive tokens is found in the list of known chemical names, they are merged into one token.	Na CL	NaCL

In general, the advantages of using ChemTok can be listed as follows: generation of longer discriminative tokens, decrease in the number of incorrectly segmented NEs, improvement in the performance of consequent NER classifiers which uses such tokenized data, decrease in the classifier learning time by reducing the number of samples used in classifier training phase. The competitive results obtained from the use of ChemTok compared to other tokenizers of state of the art ChemNER system are presented in Chapter 6.

5.4 Balanced Under Sampling

It can clearly be seen from Table 5.1 that class imbalance problem exists in the given training data when the tokens, which are parts of entity mentions ("CHEMICAL"), are considered as positive samples and others from the "OUT" class as negative ones. Imbalance in the number of samples from different classes is one of essential causes of the increase in the number of positive samples (named entities of interested) that are incorrectly identified as negatives (False Negative). The existence of False Negatives can affect the overall performance of a system by decreasing the recall value and consequently reducing the F-score as the classifier performance measure. Different strategies which deal with CIP in classification problems were discussed previously in Chapter 4. In the context of named entity recognition, the recognition goal is to correctly identify the named entities (positive entities). Negative entities, i.e. tokens which belong to the negative class ("OUT" class) are not of interest to recognize, but they can provide potentially useful information for learning algorithms considering their positions in a given sentence. This is because usually classifiers learn from tokens surrounding positive tokens i.e. tokens which belong to the positive class ("B-CHEMICAL" or "I-CHEMICAL"). Random undersampling [202] a well-known approach used for the CIP eliminates the negative samples randomly from the whole data. However, RUS may remove the negative samples around the positive ones, which are potentially useful in NER. In this thesis, a new undersampling strategy is devised to decrease the number of negatives while preserving the structure of sentences by keeping equal number of tokens which belong to the negative class on both left and right hand sides of each entity in a sentence since the information given by tokens at the vicinity of entities is very useful to improve the classifiers' performance. Thus, the method is named as

balanced undersampling. The BUS algorithm is shown in Figure 5.3. For each sentence given in the training data, BUS is used with an input sampling ratio (R_s). If the input ratio is greater than the imbalance ratio of the original sentence, there is no need to undersample and the sentence is left unchanged. Otherwise, for each named entity in an individual sentence, negative tokens are selected until the desired input sampling ratio is achieved. In the first round the negative token closest to the beginning of an entity from the right side is selected if it exists; and in the second round the negative token closes to the end of the entity is selected if it exists. In fact, unlike other popular methods, instead of removing negative tokens, BUS selects the negative tokens of the sentence that will be included in the undersampled data.

Figure 5.4 illustrates three examples of undersampling using proposed BUS algorithm. All tokens in a sentence belonging to different class types, OUT, B-CHEMICAL, and I-CHEMICAL are considered as an array for the sentence with values O, B, and I respectively. The numbers in the cells in undersampled version of the sentences indicate the order in which each negative token is selected by the algorithm. Negative tokens that are not selected by BUS are shown with ×. In the examples given, the desired input undersampling ratio, R_s is assumed as 3. For case 1 above, since the number of negative tokens needed are more than the existing ones in the sentence, the sentence is left unchanged and all negative tokens are selected. In case 2, the sentence contains 3 entities with 2 tokens each (B-CHEMICAL and I-CHEMICAL) resulting in 6 positive tokens. In order to get the desired ratio of 3, the sentence should contain a maximum of 18 negative tokens. The algorithm treats the sentence as an array of tokens and marks all positive tokens of the array as selected as discussed above. The negative tokens to be kept are selected one by one using the BUS algorithm until the desired ratio is achieved. In case 3, the sentence contains 3

entities with 1 token each. Therefore, only 9 negative tokens should remain in the resultant sentence. As discussed before, BUS considers the first token on the left hand side of each entity to be selected first.

```
N=number of tokens in S
N_p=Total number of tokens with B- and I- tag in S
N_n=Total number of tokens with O tag in S
N_s=Number of selected tokens with O tag in S
K=number of entities in S
start_k=Location of the first token of entity k
end_k=Location of the last token of entity k
R_s= Desired sampling ratio
If N_p=0
   Take the sentence out
    Finish
else:
    Mark all positive tokens in S as selected
    If R_s <= N_n/N_p /*no need for undersampling*/
       mark all O tagged tokens in S as selected and return S
    else /*Choose a subset of the negative tokens starting from the closest neighbors of the entity*/
            N_s=0
        for k=1 ..K
            left_k=start_k -1
            right_k=end_k +1
            end for
            end_0=0
            start_k+1=N
         loop
                for k=1 ..K
                    If left_k>=0 and left_k>end_k-1 and token at left_k is not selected/*S[left_k] is not selected*/
                                    mark token at left_k as selected
                                    N_s = N_s+1
                                    left_k = left_k-1
                            end if
                            if R_s<= N_s/N_p return S
                 end for
                 for k=1 ..K
                    If right_k<=N and right_k<start_k+1 and token at right_k is not selected /*S[left_k] is not selected*/
                                    mark token at right_k as selected
                                    N_s=N_s+1
                                    right_k = right_k+1
                        end if
                        if R_s<= N_s/N_p return S
                 end for
        end loop
       end if
       Undersampled sentence ← All tokens that are marked as selected in S in the same order they appear in S.
```

Figure 5.3: Balanced Undersampling Algorithm applied on each sentence

However, in this example case, the rightmost entity is the last token in the sentence. Therefore, only negative samples from the left side of this entity will be selected.

Figure 5.4: Examples show balanced undersampling

Applying BUS on training data guarantees that there are no sentences in the training data set whose imbalance ratio is greater than R_s. Moreover, using different values for R_s, will lead to the production of training data with different number of tokens which belong to the negative class, hence different imbalance ratios. The efficiency of BUS in this domain compared to the two other common undersampling methods, Random undersampling and stop word filtering is shown by conducting a series of experiments using different feature sets. Details on results achieved are presented in Chapter 6.

5.5 Feature Extraction

Feature extraction is a common task for a large number of disciplines such as machine learning, pattern recognition, data mining and statistics [203]. Feature extraction starts from an initial set of data and derives values or features intended to be informative and discriminative, which facilitate the consequent learning and

generalization tasks. Depending on the task at hand, there are some common features that can be extracted from data. The extracted features are expected to contain the relevant information from the input data that may be used effectively in performing the targeted task. To study the effect of undersampling on the performance of NER system and also to create a set of different classifiers to use in the ensemble learning module in the next step, different features have to be extracted. Features commonly used in other studies in the literature are extracted from data using the feature extraction module. In addition to single features extracted, combination of features, as applied in our studies in the domain, are also used. Each token by itself is considered as a basic feature. Moreover, preceding and following tokens of the current token are considered as "context features" since the tokens surrounding named entities provide useful information for recognition. Takuechi and Collier showed the positive effect of using preceding and following tokens in addition to the current one on the recognition performance of biomedical NER systems [204]. Following subsections explain the basic features used in depth.

5.5.1 Orthographic Features

Orthographic features provide information on the structure of words such as capitalization, combination of numbers and letters in a token, use of mix cases, presence of punctuation marks, existence of foreign characters like Greek letters etc. The investigation of the training data revealed that some of the entities of interests contain special word formation patterns, such as use of more than two punctuation marks, or existence of combination of alphabets and numbers etc. that may give a clue about their identification of entities. In the experiments, orthographic feature is represented using a binary vector. Table 5.3 shows the extracted orthographic with an example for each.

Table 5.3: Orthographic features with examples

Feature	Example	Feature	Example
UpperCase	IL-2	UpperOther	2-M
InitCap	D3	LowerUpper	25-Dihydroxyvitamin
TwoUpper	FasL	UpperDigits	AP-1
AlphaOther	AML1/ETO	LowerOther	dehydratase/dimerization
Hyphen	product-albumin	Allupper	DNA, GR, T
Upper_or_Digit	3H	Greek	NF-Kappa, beta
Digits	40	lowerDigits	gp39
AlphaDigit	IL-1beta	StartHyphen	-mediated

Each of the orthographic features in Table 5.3 corresponds to a single binary entry in the feature vector. For instance the feature vector for "*Sulfamate*" can be seen as: *1100010001000000.*

5.5.2 Morphological Features

Morphological features refer to the affixes (prefixes and suffixes) of a token. Simply *N* characters from the beginning or the end of the token can be considered as affixes. Usually these type of features are known as *N*-gram characters where the exact number of elements of the feature vector depends on the value of *N* and use of prefixes or suffixes in isolation versus in combination. For example, assuming *N* equals to 3 and taking the prefixes and suffixes separately, the length of feature vector is 6 for each token. Morphological information has been used extensively by other researchers in different NER tasks.

5.5.3 Space Features

Space features determine whether there is a space before or after a token based on its position in a given sentence. The binary feature vector of length of 3 is used to represent these features where the first entry shows the existence of a space before the token, the second one shows the existence of a space after the token and the third one corresponds to the existence of space for both before and after the token.

ChemSpot [95], one of the state of the art chemical NER system, uses space features extensively.

5.5.4 Bag of Words Features

The bag-of-words model is a simple representation of text used in natural language processing and information retrieval (IR). In this model, text is represented as a bag of words, disregarding grammar and even word order [205]. In the NLP related tasks it is common to weigh terms by various schemes instead of showing the existence or nonexistence of a word in the given document. The most popular of those are term frequency (tf) and term frequency – inverse document frequency (tf-idf) [206]. We applied both representation schemes in this thesis. tf (*t*,*D*) simply shows the number of occurrences of term, *t* in the document *D*. The inverse document frequency *idf (t, Corpus*) is a measure that shows how much information the word provides. In other words it determines whether the word is common or rare across all given documents. It is the logarithmically scaled ratio of the total number of documents in the corpus to the total number of documents which contain the term. *idf* is often given as:

$$idf\ (t,\ Corpus) = log \frac{M}{|\{D \in Corpus : t \in D\}| + 1} \quad (5.1)$$

where, *M* shows the total number of documents in the corpus, $M = |Corpus|$ and $|\{D \in Corpus : t \in D\}|$: the number of documents where the term t appears. It is common to sum up the denominator by 1 to avoid division by zero in case the term does not exist in the documents. Hence *tf-idf* can be calculated according to:

$$tfidf\ (t, D, corpus) = tf\ (t, D) * idf\ (t, Corpus) \quad (5.2)$$

A high *tf–idf* weight can be obtained by a high term frequency in the given document and a low document frequency of the term in the corpus. Hence, the lower weights tend to filter out common terms.

5.5.5 Word Shape

Word shape features are another kind of word representation schemes based on the appearance of a token as well as the order and number of its constituent characters. Three types of word shape features are created for each token in our experiments. The first one is a simple word shape feature, in which each character from special categories such as digits, upper case letters, lower case letters, punctuation marks, and Greek characters is replaced by their representative characters. For example if we consider D, U, L, P, and G as representative letters for the mentioned character groups respectively and O for other characters that do not belong to mentioned ones, then word shape of word "β-13-Galactosidase" is "GPDDULLLLLLLLLLL" . The second type of word shape features is the squeezed word shape feature which can be seen as the summarized version of the simple word shape feature. In this case, instead of repeating representative letters, they are followed by their number of occurrences (for the cases where the number of occurrences is greater than 1). The squeezed word shape of above example is given as "GPD2UL12". The last form of word shape feature is the digital sign of each token, which shows the number of constituent characters in a predefined order. For instance, if the order representative characters is in the form of D, U, L, P, G, and O, then the digital sign of the given example is given as "2112110".

5.5.6 Output of OSCAR classifier

Using class predictions of other well performing NER systems as a feature is a common approach in the design of NER systems. OSCAR [94] is a state-of-the-art chemical NER system. Its output is used as features in our task. Our experiments show that using this feature alone outperforms some other classifiers using other individual features.

5.5.7 Domain Specific Features

These features are binary type features, which illustrate the existence or nonexistence of tokens in predefined lists [199], [201] of common chemical or drug names, chemical elements, abbreviation of chemical elements, and amino acids. Moreover the existence of most well-known chemical affixes (prefixes and suffixes) [200] for each token are considered.

5.5.8 Lexical Features

Grammatical roles of tokens in the data are considered as lexical features. These include part-of-speech (POS) tags, phrase position and base noun phrase tags. Description of these features is given blow:

- **POS Tag**: The effect of POS features in the recognition of named entities in the biological domain has been tested previously by various researchers and different views on its impact have been reported [206]. Because of the similarities between biological and chemical domains, it has been used as a feature in our experiments. Since none of the released data sets in the ChemDNER corpus include POS tags, the Genia tagger [207] tool, which is trained on both the newswire and biomedical domains, is used for adding POS tags to the tokens.
- **Phrase Tag**: Phrase boundaries are expected to coincide with the boundary of multi-token names. All data sets in the corpus were tagged for all phrase tags including Noun, Verb, Adverb, SBAR, Prepositional, Adjective phrases using the IOB2 representation. Genia tagger was used for extracting this feature.
- **Base Form**: This feature shows the base form of the word.

5.5.9 Word Clustering Feature

Different researchers have shown that utilizing unlabeled data can improve the quality of NER systems [208], [209], [210]. Hence, training and development data without their labels are mixed together to get large amount of unlabeled data. Since Brown clustering was successfully applied in NER in [211], it is also applied as a cluster creation approach for words used to create unlabeled data set for our experiments. Brown's algorithm is a hierarchical clustering algorithm, which clusters the words that have a higher mutual information of bigrams [118]. The output of the algorithm is a dendrogram. A path from the root of the dendrogram represents a word and can be encoded with a bit sequence. The prefixes of length 50 of such encodings is chosen as the word clustering feature, which produced 10368 clusters.

5.5.10 Feature Sets Used

The aforementioned features are used in isolation as well as in various combinations. Only the combinations of features that were proven to be useful in previous NER studies have been used in our experiments. All 19 feature sets used during the experiments are illustrated in Table 5.4.

5.6 Classifier Training

All baseline classifiers using different feature sets listed in Table 5.4 are trained using SimpleTagger interface of Mallet [128] with default parameters, where the number of iterations was set to 500 and Gaussian variance was 10. Mallet toolkit makes use of CRFs as its classification algorithm (more detail about CRFs is provided in Appendix B). The classification performance of the baseline classifiers trained using the train data in its original form (without undersampling) on development and test data are shown in Table 5.5.

Table 5.4: Feature sets used in experiments

	Domain Context	Cluster	Context	Morphological	Ortho	OSCAR	Lexical	Space	tf	tfidf	Word Shape
F1	√	√	√	√	√	√	√	√	√	√	√
F2	√		√	√	√		√	√			√
F3	√		√	√	√	√	√	√			√
F4	√										
F5		√									
F6				√							
F7				√	√						
F8				√			√				
F9				√				√			
F10				√			√	√			
F11				√			√	√			√
F12				√							√
F13					√						
F14						√					
F15							√				
F16								√			
F17								√	√		
F18								√		√	
F19											√

The best performing classifier on development and test data is classifier $E_{1,}$ which uses the combination of all features (F_1). Moreover, it is clear that almost all classifiers suffer from low recall value in comparison to corresponding precision values as a consequence of highly imbalanced data. Since the sampling ratio for which the classifier will be maximized, (R_{best}), for training data is not known beforehand, the sampling process is done for a range of input sampling ratios (R_s). Furthermore, since the best classification performance for classifiers trained using different feature sets is likely to occur at different R_{best} values, BUS is applied on training data with R_s in the range [2:50] incremented by one to find the best value for sampling ratio experimentally. The number of all sentences in the given training data is 30418, of which 3651 sentences do not include any chemical NEs.

Table 5.5: Performance of baseline classifiers with different feature sets on development and test data

	Development			Test		
	Recall	**Precision**	**F Score**	**Recall**	**Precision**	**F-score**
E_1	77.54	85.19	81.19	77.55	85.25	81.21
E_2	74.16	82.97	78.32	73.21	84.35	78.39
E_3	77.29	84.57	80.77	76.45	84.96	80.48
E_4	51.39	75.5	61.15	51.28	77.54	61.73
E_5	54.36	75.26	63.13	54.22	77.22	63.71
E_6	67.35	76.87	71.8	67.5	79.38	72.96
E_7	69.11	72.52	70.77	64.68	78.31	70.85
E_8	65.71	77.53	71.13	65.48	79.71	71.9
E_9	68.4	79.86	73.69	67.92	81.97	74.29
E_{10}	72.31	75.67	73.95	68.01	81.63	74.2
E_{11}	72.47	80.83	76.42	71.55	82.69	76.72
E_{12}	67.99	77.28	72.34	65.1	78.3	71.09
E_{13}	52.43	75.32	61.82	52.13	77.36	62.29
E_{14}	65.45	80.66	72.26	63.77	78.17	70.24
E_{15}	49.74	72.72	59.07	49.18	74.52	59.25
E_{16}	56.66	79.57	66.19	56.34	81.34	66.57
E_{17}	51.3	79.02	62.21	50.33	80.53	61.95
E_{18}	50.19	76.19	60.52	30.48	79.96	44.14
E_{19}	56.83	73.12	63.96	55.75	76.08	64.35

Table 5.6 shows the percentage as well as the number of sentences whose sampling ratios can be categorized into different R_s ranges in the training data. Moreover, the average sampling ratio for each range is also depicted.

Table 5.6: Distribution of training data

Range	**Percentage of existed sentences**	**Exact number of sentences**	**Average N/P**
[2,10]	20.03%	5364	5.04
(10,20]	16.82%	4504	10.43
(20,30]	21.12%	5655	20.31
(30,40]	19.59%	5246	30.12
(40,50]	22.29%	5969	40.02
(50,∞)	0.1%	29	50.04
[2,∞)	100%	26767	20.93

As illustrated in Table 5.6 the majority of sentences in the training data have imbalance ratios between 40 and 50. Therefore, 50 is chosen as the upper bound during sampling experiments. The 49 different training data sets using sampling ratios between 2 and 50 are created for every classifier, which uses a different feature set. This results in an initial ensemble of 950 classifiers (19×49=931 classifiers trained with undersampled data)+19 classifiers trained using original data. The performance of all 950 classifiers on both development and test data are given in Appendix C. The results are discussed in detail in Chapter 6.

The 950 classifiers trained are used by the ensemble learning module described in the next section to select the most appropriate subset of classifiers and combine them in order to benefit from MSC. The diversity between classifiers is due to the variation in the undersampling ratios used on training data as well as the differences between features used during classifier generation process.

5.7 Classifier Ensemble Scheme

This module is responsible of selecting the best performing subset of classifiers among all classifiers in the initial pool with the aim of increasing classification performance. Metaheuristic algorithms provide a reasonable equilibrium between search complexity and solution quality which enables the application of these algorithms to a vast number of problems. Static classifier selection is one of those problems that can be solved by this type of algorithms. In this study, PSO is used as the underlying classifier selection technique. The constriction factor method as a variant of basic PSO is employed in the presented model. Hereafter in this dissertation the word PSO, refers to the constriction factor method version. Unlike other metaheuristic algorithms, PSO has a flexible and appropriate mechanism to

enhance and adapt the local and global exploration abilities [212]. In addition it has been shown that PSO outperforms the other metaheuristic techniques [213]. Furthermore, its characteristics such as simplicity in concept, implementation easiness, and computational efficiency make this method suitable for our experiments. The proposed system makes use of PSO to choose a set of classifiers from the, pool which when combined, makes more reliable predictions compared to initial classifiers. Therefore, in the presented ensemble approach, only a set of classifiers is selected such that their overall cumulative prediction abilities for all classes show the better results in comparison to the combination of others. The Naive Bayesian combination approach is used to merge the outputs of the selected classifiers. Details of the proposed scheme are presented in the following sections.

5.7.1 Implementation of the Ensemble Scheme using PSO

In the PSO algorithm, each possible solution of the problem is represented as a particle, which can be assumed like the chromosome in genetic algorithms. Since PSO is a population based algorithm, it needs initialization. Hence, the population of solutions at the beginning can be generated randomly. Population evolves by considering the positions and speeds of particles. At the end of the evolution phase, the most appropriate solution according to the objective function is selected as the best solution for the problem. The original PSO algorithm was presented to solve real valued problems, where each entry of the solution vector could take any real number based on the constraints of the problem at hand. For the proposed system we need binary vectors, where each entry corresponds to an individual classifier in the pool. To create binary solution vectors by the evolution process of PSO, the binary version of PSO, known as BPSO [212], is employed in the proposed architecture. The only difference between PSO and BPSO is in the ways that the entries of the particles are

updated. In PSO, after computing the speed of each particle, the current value of particle plus its speed is taken as the new particle's value. In contrast, in BPSO, the new binary value of positions is calculated according to equation 5.3:

$$X_{id} = f(x) = \begin{cases} 1, & rand() < S(V_{id}) \\ 0, & Otherwise \end{cases} \quad (5.3)$$

where S (.) is a sigmoid function, S (x) = $\frac{1}{1+e^{-x}}$ and V_{id} , corresponds to the speed of d^{th} entry of i^{th} particle. The value if V_{id} in BPSO algorithm is also limited in [V_{min}, V_{max}]. In the cases that computed V_{id} exceeds the V_{max} or it becomes less than V_{min}, it is replaces with V_{max} and V_{min} respectively.

Each solution vector represents a different ensemble member. If the value of an entry is 1, this means that the corresponding classifier is allowed to contribute to the ensemble, otherwise, the classifier is not allowed to participate in the joint decision for the final prediction. The length of the particles or solution vectors is determined by the number of candidate classifiers in the classifiers repository. In our experiments in the worst case the length of the solution vectors is 950. At the initialization phase, the entries of each particle are randomly set to either 1 or 0. The fitness of each particle is defined as the F-score achieved by the combination of classifiers whose corresponding entries are 1. In order to label a given test sample, the class receiving the maximum combined score is selected as the collaborative decision. When Naïve Bayesian combination approach is applied, the combined score of a particular class y_i for each given sample is defined as in equation 5.4:

$$F(y_i) = \frac{N_i}{n} \left\{ \prod_{j=1}^{L} \frac{CM_{i,s_j}^{j} + \frac{1}{N}}{N_i + 1} \right\}^{B} \quad (5.4)$$

where N_i is the number of samples from class y_i in the development data, N is the total number of classes in the corpus. CM_{i,s_j} is the number of samples whose true

class labels was y_i and is assigned by classifier *j* to class *S*. *L* denotes the number of classifiers participating in the ensemble. *B* is the Titterington coefficient which is discussed in the Section 3.4.3.

The classifier selection based on BPSO, depicted in Figure 5.5, begins with a randomly initialized population of particles and evolves by means of updating the particles' speed and subsequently changing their positions.

The velocity and positions of particles would be updated according to the equations 5.5 and 5.3 respectively.

$$v_{id} = \chi\ (v_{id} + c_1 r_1 (pBest_{id} - x_{id}) + c_2 r_2 (gBest_{id} - x_{id})) \quad (5.5)$$

where c_1 and c_2 are acceleration coefficients. Let $\varphi = c_1 + c_2$ then χ would be defined as the constriction factor according to equation 5.6.

$$\chi = \frac{2}{|\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}|} \quad , \quad \text{for } \varphi > 4 \quad (5.6)$$

```
For each candidate_ensemble
   Initialize candidate ensemble
END

Do
   For each candidate_ensemble
      Calculate fitness value   /* Performance of combination result in terms of F score */
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   End

   Choose the candidate_ensemble with the best fitness value of all the candidate_ensemble as the gBest
   For each candidate_ensemble
      Calculate candidate_ensemble velocity according Equation (5.5)
      Update candidate_ensemble position according Equation (5.3)
   End
While maximum iterations or minimum error criteria is not attained
```

Figure 5.5: BPSO Algorithm

The population size, and constriction factor have important impact on the quality of the solution found in the CFM version of PSO. In some studies the appropriate ranges of values for the mentioned parameters are suggested for different kinds of problems [213], [214], [215]. However, the V_{min} and V_{max} are considered to be -6 and 6 according similar to the work has been done in [218]. Table 5.7 shows the suggested ranges of candidate values per each parameter.

Table 5.7: Parameter ranges

Parameter	Range	Slope
φ	(4.00,5]	0. 1
Iterations	[100,300]	50
N (Pop size)	[1,3] × Particle size	1

In the ensemble training phase, the different parameters for BPSO are selected based on the given values in Table 5.6 using validation data. Moreover, Titterington parameter for Bayesian combiner is chosen. Naïve Bayesian combination method needs to have a confusion matrix to combine independent classifiers. 25% of development data is used for creating the confusion matrices using trained classifiers with different feature sets. Remaining development data is used for tuning the needed parameters of presented ensemble scheme. The selected best performing subset of classifiers after tuning the model's parameter is applied on test data using same confusion matrices created on small portion of development data. Three series of experiments are conducted by varying N, φ, and iterations with the given slope for each in turn, and keeping the other two parameters unchanged respectively. Acceleration coefficients which adjust the relative velocity toward the local and global best particle (C_1 and C_2 respectively), are both considered equal ($C_1 = C_2 =$

φ/2). For each parameter tuning process, the algorithm is run for up to 300 iterations and the value of parameters which corresponds to the best classification performance are selected.

The value of φ is selected as 4.2 when N=100, and number of iterations is set to 300. It is clear from equation 5.6 that increasing the value of φ will cause a decrease in the value of constriction factor χ. A larger χ means that the search distance of every step for each particle becomes larger. This in turn allows avoiding local exploitations and facilitates global explorations. On the contrary, in this case the algorithm cannot achieve the refinement of the optimal solution. A smaller χ will decrease the search distance and will direct more attention on local exploitations and the algorithm can hardly cover the search space. The optimal value for χ is related to the type of function that needs to be optimized. For unimodal functions, large values of χ may refine the solution whereas for multimodal functions small values of χ are usually generate better results.

Since the size of particles in the experiments is very large, the range [100, 200] together with slope value 50 is taken into account instead of proposed values in Table 5.6 as the population size. The selected value for population size is 200. A large population size may causes an increase in the reliability of algorithm but it is nevertheless required for more solution evaluations and increase the computing effort for convergence. For all sets of experiments in this dissertation, the Titterington factor, β, for Naïve Bayesian combiner is assumed as 1. Titterington in his work suggested that β can be taken as 0.5, 0.8, or 1. Hence, to find the optimum value, after choosing the aforementioned parameters above, the experiments are repeated

twice for $B = 0.5$, and $B = 0.8$. Experiments have shown that using $B = 1$ generates slightly better results.

The number of all candidate classifiers in the initial pool is very large. Moreover, it has been shown during experiments that some classifiers perform poorly. It is expected that including such classifiers in the ensemble may have a negative effect on the ensemble's overall performance. Hence members from the pool are divided into five subsets in ascending order of performances. All experiments have been repeated for all new subsets. The results on the development data show that using only the classifiers which achieved an F-score greater than 70% among all 950 classifiers generate better results in comparison to the other subsets as well as using the full ensemble of 950 classifiers. As a result, the number of classifiers included in the initial pool is limited to 515. In the next Chapter the results of using this new subset of 515 classifiers are presented and discussed.

Chapter 6

RESULTS AND DISCUSSION

6.1 Introduction

In this chapter the performance of each individual module in the proposed architecture is discussed in detail. The efficiency of the tokenizer is presented and compared to some commonly used tokenizers in the next section. The characteristics of each individual classifier and effects of different undersampling approaches follow next. The effect of using multiple classifier approaches is investigated in the final part of the chapter.

6.2 Effect of Tokenization Method

In this section we compare the results of the proposed tokenizer, ChemTok, to two other well-known tokenizers in this domain: ChemSpot [94] tokenizer, and the tokenizer module of tmChem [25], the best ranked ChemNER system in BioCreative IV shared task. The result of using only the white space tokenizer as a basic approach for tokenization is also presented as a baseline. Table 6.1 shows the number and average length of tokens produced by each of the mentioned tokenizers used for the corpus described in Chapter 5. Additionally the number of incorrectly segmented entity names is given in the third column for each data set.

It can be seen in Table 6.1 that the white space tokenizer tokenizes text into fewer number of longer tokens but produces very large number of incorrectly segmented entities compared to the rest. On the other hand it can be observed that even though

ChemTok produces slightly longer tokens compared to ChemSpot and tmChem, the number of incorrectly segmented entities is minimized, showing that the boundaries of NEs are correctly identified by the proposed tokenizer.

Table 6.1: Comparison of number of tokens (NT), average token length (ATL), and number of incorrectly segmented entities (NISE) for various tokenizers

Data Set		**ChemSpot**			**tmChem**			**White Space Tokenizer**			**ChemTok**		
		NT	**ATL**	**NISE**	**NT**	**ATL**	**NISE**	**NT**	**ATL**	**NISE**	**NT**	**ATL**	**NISE**
ChemDNER	**Train**	907405	4.62	40	965056	4.35	11	718244	5.84	9189	899343	4.66	6
	Development	901610	4.64	36	958475	4.36	11	714287	5.85	9174	893180	4.68	3
	Test	779700	4.63	8	828001	4.36	3	513630	5.85	7804	772847	4.67	3

The concept we refer to as the incorrectly segmented entities is important since the NER classifiers will not be able to identify NEs correctly if the entity mentions are not segmented at the right boundaries. In addition to incorrect segmentation problems associated with the white space tokenization, several other factors lead to incorrect segmentation even when other types of tokenizers are used. For instance, often an entity name appears in its plural form in text, such as '***salicylates***' or '***clonidines***' where the actual entity mention is annotated as '***salicylate***' or '***clonidine***'. Such plural forms are usually incorrectly segmented by many tokenizers. Since this issue is taken into account in the proposed method, ChemTok does not suffer from problem related with plural forms. Finally, sometimes NEs are joined to other parts of text due to mistakes during various stages of text preprocessing as in the example of '***CONCLUSIONGlucose***' where the annotators mark '***Glucose***' as the NE but a tokenizer which uses a rule to split NEs at the point where there is a case change will incorrectly segment the NE to '***lucose***'. The second and third types

of incorrect segmentation are very difficult to detect. In order to further investigate the impact of the mentioned tokenization methods on NER performance, classification experiments are performed using data segmented by each of the four tokenizers. As classification algorithm, CRFs classifier is employed to conduct the experiments. All systems are trained using common features in [28].Table 6.2 shows the overall performances achieved when the classifiers are trained on BioCreative train data and tested on development and test data sets.

Table 6.2: NER performance of classifiers using ChemDNER corpus

	Development			**Test**		
Tokenizer	**Recall**	**Precision**	**F-Score**	**Recall**	**Precision**	**F-Score**
ChemSpot	77.31	84.43	78.46	74.62	83.68	78.89
tmChem	71.57	81.36	76.15	71.47	82.29	76.50
White Space	66.98	86.21	75.39	68.25	84.33	75.44
ChemTok	**76.76**	**87.49**	**81.77**	**76**	**88.78**	**81.89**

It can be seen from Tables 6.2 that the overall NER performance of the classifiers which use white space tokenization is very inferior compared to that of all other classifiers mainly due to the large number of incorrectly segmented entities. On the other hand, the performance of the classifiers utilizing ChemTok is higher than the others. The higher improvement over the other tokenizers when BioCreative data set is used for testing can be attributed to the fact that ChemTok uses rules extracted from BioCreative training data set. However, the good performance of ChemSpot has been shown to generalize to other data sets in the experiments presented in [28]. Since ChemTok has resulted in best NER performance, it has been used through all classification tasks in this study.

6.3 Effect of Undersampling

A total of 19 feature sets were utilized in the experiments presented in this chapter. These feature sets contain features used commonly in NER tasks as well as features specific to the chemical domain. All base features extracted from the data are given in section 5.5 and all 19 feature sets created by combination of different base features are presented in section 5.5.10. In this section the effect of the proposed balanced undersampling approach along with two other popular undersampling methods namely random undersampling and stop word filtering on classification performance of classifiers trained using these feature sets is discussed. The performance of all classifiers using undersampled training data with different sampling ratios for both BUS and RUS methods is given in Appendix C. Results are based on the generalization on both development and test data. Due to the random selection of negative samples by the RUS method, all sampling experiments with this approach, are repeated 5 times and the averaged results are presented.

Figures 6.1 - 6.3 show the effect of different undersampling approaches on each individual evaluation measure; recall, precision, F-score; for each classifier applying on test data. Results obtained for BUS and RUS algorithms are given for the sampling ratio for which the maximum F-score is achieved.

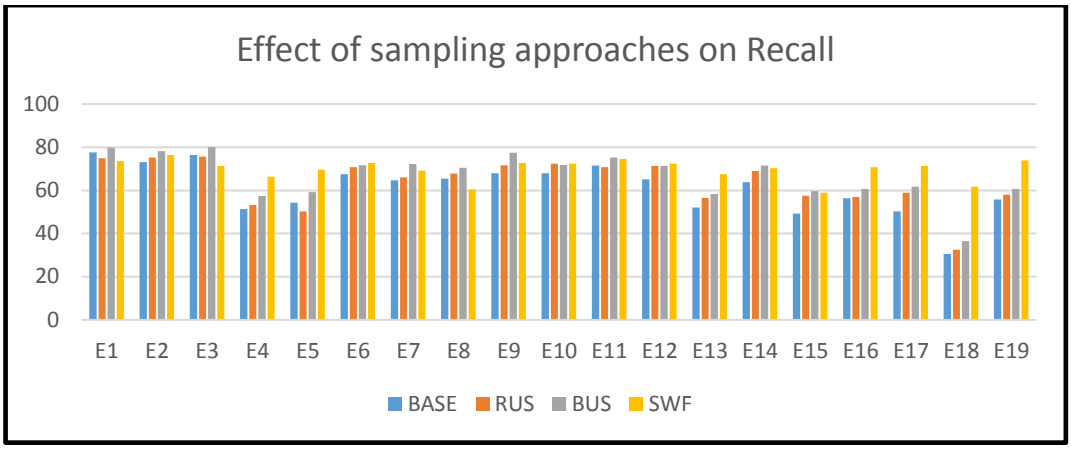

Figure 6.1: Effect of undersampling on Recall

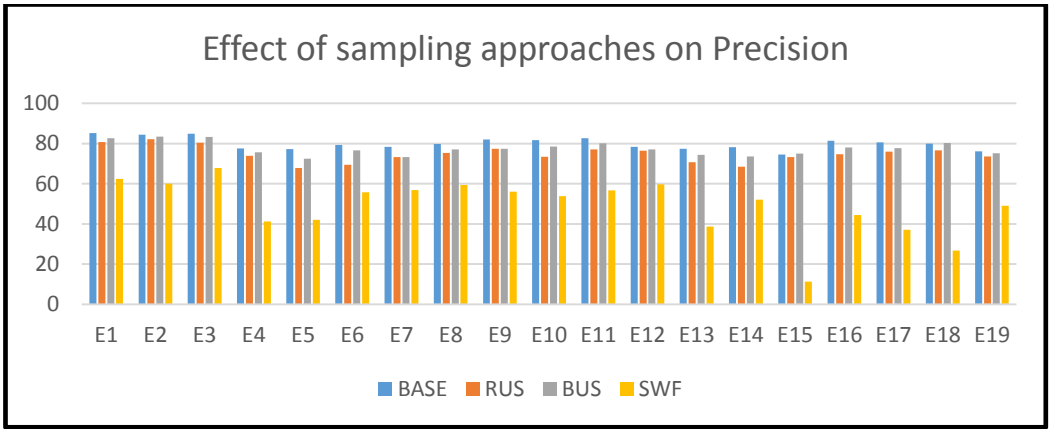

Figure 6.2: Effect of undersampling on Precision

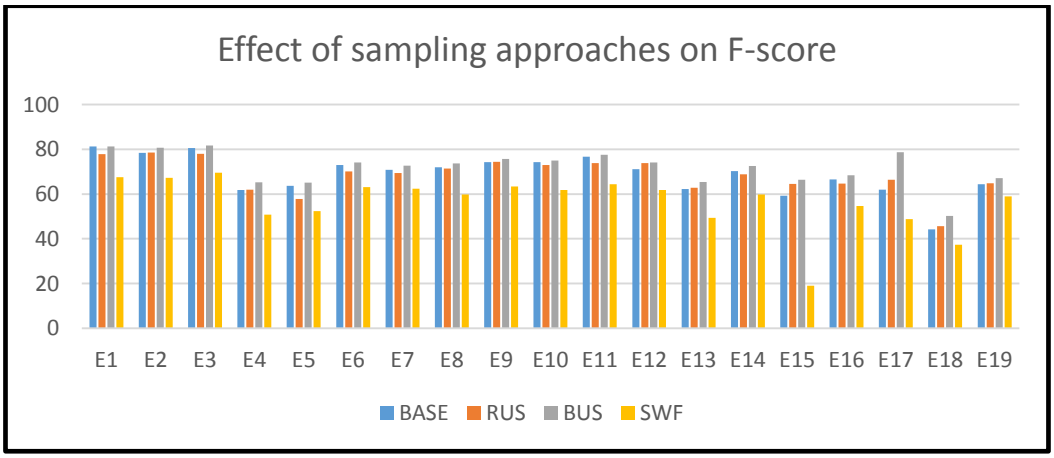

Figure 6.3: Effect of undersampling on F-score

As it can be seen from Figure 6.1, applying RUS improves the recall value of all baseline classifiers except for E_1, E_3, E_5, and $E_{11.}$ However, applying BUS improves the recalls value of all classifiers while applying SWF increases the recall for some classifiers. In general the amount of improvement in recall is more when BUS is applied compared to RUS.

Considering Figure 6.2, it can be seen that the precision of almost all classifiers is decreased by all undersampling approaches, only E_{15} and E_{18} have shown to slightly improve when BUS is applied. However, the amount of degradation in precision values for all classifiers when BUS is applied is less than the two other methods.

Table 6.3 shows the detailed performance of baseline classifiers using different feature sets which are trained using original training data to the performance of classifiers trained using different undersampling strategies. The results presented in Table 6.3 for the BUS and RUS approaches shows the classification performance at best sampling ratios (R_{best}) determined as explained in section 5.6 experimentally. For example considering classier E_2, Figure 6.4 shows the baseline performance and the sampling ratio, R_{best}, for which the maximum classification performance is achieved in terms of F-score (R_{best}=36,F-score=80.39)

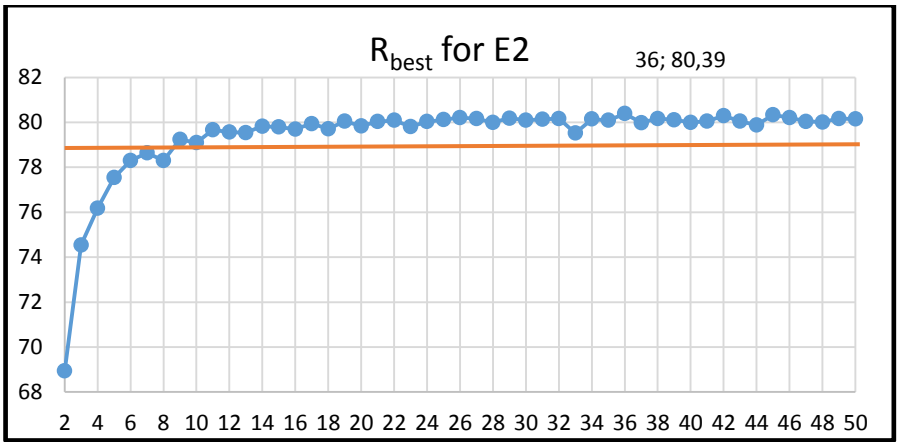

Figure 6.4: R_{best} selection for classifier E_2

In Figure 6.4, the performance of classifier trained with original training data without undersampling is 78.32%, however the maximum improvement using undersampled data is obtained in the sampling ratio 36, which the obtained performance is 80.39.

Table 6.3: Classification performance using different undersampling approaches

Classifiers		Development				Test		
		Recall	Precision	F score	R_{best}	Recall	Precision	F score
E_1	**Baseline**	77.54	85.19	81.19	-	77.55	85.25	81.21
	BUS	80.99	82.41	81.69	41	79.8	82.73	81.24
	RUS	77.74	81.35	79.5	47	74.99	80.78	77.78
	Stop Word Filtering	72.95	63.56	67.94	-	73.6	62.35	67.51
E_2	**Baseline**	74.16	82.97	78.32	-	73.21	84.35	78.39
	BUS	79.08	81.75	80.39	36	78.23	83.42	80.74
	RUS	76.2	80.12	78.11	45	75.24	82.25	78.59
	Stop Word Filtering	73.42	60.82	66.53	-	76.34	59.98	67.18
E_3	**Baseline**	77.29	84.57	80.77	-	76.45	84.96	80.48
	BUS	81.02	82.48	81.74	41	80.11	83.26	81.65
	RUS	78.2	80.52	79.34	42	75.69	80.37	77.96
	Stop Word Filtering	73.31	65.98	69.45	-	71.38	67.85	69.57
E_4	**Baseline**	51.39	75.5	61.15	-	51.28	77.54	61.73
	BUS	57.61	73.42	64.56	32	57.44	75.62	65.29
	RUS	53.8	71.4	61.36	38	53.27	73.88	61.9
	Stop Word Filtering	70.63	41.66	52.41	-	66.29	41.23	50.84
E_5	**Baseline**	54.36	75.26	63.13	-	54.22	77.22	63.71
	BUS	59.62	70.84	64.75	16	59.17	72.48	65.15
	RUS	55.2	70.43	61.89	32	50.23	67.84	57.72
	Stop Word Filtering	67.33	48.54	56.41	-	69.6	42.03	52.41
E_6	**Baseline**	67.35	76.87	71.8	-	67.5	79.38	72.96
	BUS	71.46	74.05	72.73	49	71.61	76.63	74.04
	RUS	70	72.25	71.11	11	70.75	69.47	70.09
	Stop Word Filtering	71.49	56.28	62.98	-	72.73	55.73	63.1
E_7	**Baseline**	69.11	72.52	70.77	-	64.68	78.31	70.85
	BUS	72.56	72.28	72.46	50	72.18	73.25	72.71
	RUS	67.79	72.33	69.99	50	65.96	73.21	69.4
	Stop Word Filtering	61.86	62.24	62.05	-	69.08	56.91	62.41
E_8	**Baseline**	65.71	77.53	71.13		65.48	79.71	71.9
	BUS	70.66	74.81	72.68	50	70.45	77.12	73.63
	RUS	67.94	72.64	70.21	41	67.81	75.38	71.39
	Stop Word Filtering	71.34	56.84	63.27	-	60.35	59.32	59.83
E_9	**Baseline**	68.4	79.86	73.69	-	67.92	81.97	74.29
	BUS	77.11	75.27	76.18	7	77.48	77.36	75.68
	RUS	72.11	74.85	73.45	30	71.66	77.35	74.4
	Stop Word Filtering	68.73	60.87	64.56	-	72.64	56.1	63.31
E_{10}	**Baseline**	72.31	75.67	73.95	-	68.01	81.63	74.2
	BUS	71.27	80.78	75.73	50	71.75	78.44	74.95
	RUS	74.25	72.34	73.28	9	72.47	73.34	72.9
	Stop Word Filtering	71.69	57.59	63.87	-	72.44	53.78	61.73
E_{11}	**Baseline**	72.47	80.83	76.42	-	71.55	82.69	76.72
	BUS	76.28	78.1	77.18	46	75.22	80.09	77.58
	RUS	73.35	76.34	74.82	24	70.77	77.11	73.8
	Stop Word Filtering	74.69	61.26	67.31	-	74.57	56.61	64.36
E_{12}	**Baseline**	67.99	77.28	72.34	-	65.1	78.3	71.09
	BUS	71.68	74.56	73.09	37	71.43	77.09	74.15
	RUS	69.06	72.32	70.65	27	71.33	76.46	73.81
	Stop Word Filtering	71.87	52.85	60.91	-	72.44	59.79	61.73

(E_i corresponds to classifier trained and tested using feature set F_i in Table 5.4)

Table 6.4: (Continued.)

Classifiers		Development				Test		
		Recall	Precision	F score	R_{best}	Recall	Precision	F score
E_{13}	**Baseline**	52.43	75.32	61.82	-	52.13	77.36	62.29
	BUS	58.86	72.49	64.97	28	58.33	74.42	65.4
	RUS	56.94	68.65	62.25	16	56.59	70.69	62.86
	Stop Word Filtering	66.38	39.4	49.45	-	67.49	38.75	49.33
E_{14}	**Baseline**	65.45	80.66	72.26	-	63.77	78.17	70.24
	BUS	73.97	75.79	74.87	26	71.54	73.6	72.56
	RUS	72.94	71.87	72.4	39	68.94	68.54	68.74
	Stop Word Filtering	72.35	55.92	63.08	-	70.36	52.04	59.83
E_{15}	**Baseline**	49.74	72.72	59.07	-	49.18	74.52	59.25
	BUS	60.05	72.91	65.86	37	59.64	74.97	66.43
	RUS	58.09	71.32	64.03	50	57.56	73.23	64.46
	Stop Word Filtering	58.01	11.56	19.28	-	58.87	11.3	18.96
E_{16}	**Baseline**	56.66	79.57	66.19	-	56.34	81.34	66.57
	BUS	61.31	76.55	68.09	16	60.74	78.08	68.33
	RUS	58.48	73.62	65.18	13	57.03	74.61	64.65
	Stop Word Filtering	70.68	49.55	58.25	-	70.79	44.45	54.61
E_{17}	**Baseline**	51.3	79.02	62.21	-	50.33	80.53	61.95
	BUS	62.13	75.78	68.28	16	61.64	77.65	68.72
	RUS	59.02	73.26	65.37	16	58.91	76.01	66.38
	Stop Word Filtering	71.56	40.51	51.73	-	71.36	37.11	48.83
E_{18}	**Baseline**	50.19	76.19	60.52	-	30.48	79.96	44.14
	BUS	58.83	73.65	64.16	25	36.44	80.3	50.13
	RUS	55.21	70.15	61.79	13	32.4	76.64	45.55
	Stop Word Filtering	69.84	40.61	51.36	-	61.73	26.76	37.34
E_{19}	**Baseline**	56.83	73.12	63.96	-	55.75	76.08	64.35
	BUS	60.72	72.79	66.21	28	60.73	75.11	67.16
	RUS	58.37	71.37	64.22	49	57.94	73.51	64.8
	Stop Word Filtering	72.06	49.3	58.55	-	73.94	49.05	58.97

(E_i corresponds to classifier trained and tested using feature set F_i in Table 5.4)

The results presented in Table 6.3 can be summarized as follows:

- The performance of classifiers using different feature sets varies.
- It can be seen that using stop word filtering as the undersampling method degrades the classification performance. This is mainly due to the fact that each sentence contains many stop words and removing stop words which are not part of entities (negative samples) and keeping the ones which are within entities from the training data causes the classifier to learn and predict every such token in test data as positive thus making a false positive prediction decreasing precision.

- Employing RUS improves the performance of classifiers (E_4, E_{13}, E_{14}, E_{15}, E_{17}, E_{18}, and E_{19}). However the amount of improvements for classifiers using feature sets E_4, E_{13}, E_{14}, and E_{19} is less than others in terms of F-score. Nevertheless it can be observed that the recall performance of all classifiers except E_1 improves and the precision scores degrade, resulting in classifiers which are more balanced in terms of precision-recall scores.
- On the other hand, using BUS, improves the performances of all classifiers. In this case the recall values are further improved where the loss in precision values are less compared to results obtained using RUS. As a result, classifiers, which are more balanced in terms of precision-recall values, are obtained achieving higher F-scores. The main reason for this improvement is due to the fact that when BUS is used the negative samples neighboring the positive samples are mainly preserved as explained in Chapter 5.
- The maximum improvement applying BUS is achieved by classifier E_{15} which performs as the worst baseline classifier. For this case improvement is 6.98% in terms of F-score using test data. On the other hand, the minimum improvement is achieved by classifier E_1, which is the baseline classifier with the highest F-score. It is important to note that for the case of E_{15} the gap between precision and recall using original data (not undersampled) is the largest whereas this gap is minimal for the case of E_1. Therefore, it can be deduced that BUS contributes to the generation of more balanced classifiers when the original precision-recall gap is bigger.
- By comparing results on development and test data, it can be seen that using BUS method, all classifiers which achieve an improvement in F-

score on development data using R_{best} value as the undersampling ratio also achieve an improvement on test data for the same undersampling ratio, R_{best} . The best performing classifier on test data is E_3 using R_{best}=41. The amount of improvement is 1.17% in F-score compared to the classifier which is trained using the original data.

Further analysis of Table 6.3 reveals that the R_{best} value at which the classifiers perform best on development data are different for classifiers using different feature sets. Table 6.4 shows the R_{best} value for each classifier using BUS and the amount of classification gain in terms of F-score on test data compared to the respective baseline classifiers.

Table 6.5: R_{best} values for different classifiers using BUS

Classifier	R_{best}	Amount of Improvement (%)
E_1	41	0.5
E_2	36	2.35
E_3	41	0.97
E_4	32	3.56
E_5	16	1.62
E_6	49	1.08
E_7	50	1.62
E_8	50	1.55
E_9	7	1.39
E_{10}	50	0.75
E_{11}	46	0.86
E_{12}	37	0.75
E_{13}	28	3.15
E_{14}	26	2.32
E_{15}	37	6.79
E_{16}	16	1.76
E_{17}	16	6.07
E_{18}	25	5.99
E_{19}	28	2.25

In the next section we discuss the classification results achieved when a pool of classifiers obtained using different R_{best} values are selected and combined using different methods.

6.4 Effect of Classifier Combination

In this section the performance of the proposed scheme for classifier ensemble is compared to the performance of 11 other MCS approaches implemented as part of this thesis. In addition, the proposed method is compared to the single best (SB) classifier which is E_3 using R_{best}=41. The classifiers that make up the pool are created by applying the BUS method with different sampling ratios (R_s) in the range [2:50] using different 19 feature sets. Since the classifiers in the pool are desired to have a relatively good performance in addition to being diverse from each other, only the classifiers whose performances in terms of F-score was greater than 70% are included in the initial pool. This resulted in 515 classifiers in the initial pool. 6 different ensembles are formed using: i) classifiers selected using PSO, ii) classifier selected using FS, iii) classifier selected using BE, iv) the set of all classifiers, v) the set of all baseline classifiers trained using full data and vi) the set of all classifiers using undersampled data with BUS method. All ensembles are combined using 2 methods; Naïve Bayesian approach and majority voting. The single best classifier in the pool is assumed as the baseline classifier. All MSCs implemented are given in Table 6.5 below.

BPSO is the proposed ensemble scheme which uses particle swarm optimization as the population based classifier selection algorithm along with Naïve Bayesian method as the combination approach. To investigate the effect of combination approach on the ensemble, we repeated the experiments using majority voting

approach which is the most commonly used classifiers combination method used in MCSs (MVPSO). Forward Selection and Backward Elimination as static classifier selection methods are also implemented using both majority voting and Naïve Bayesian combination methods (BFS, MVFS, MVBE, and BBE).

Table 6.6: MCS methods investigated

No	Selection Method	Combination Method	Abbreviation
1	PSO	Naïve Bayesian Approach	BPSO
2	PSO	Majority Voting	MVPSO
3	Forward Selection	Naïve Bayesian Approach	BFS
4	Forward Selection	Majority Voting	MVFS
5	Backward Elimination	Majority Voting	MVBE
6	Backward Elimination	Naïve Bayesian Approach	BBE
7	All members of Pool	Majority Voting	MVFULL
8	All members of Pool	Naïve Bayesian Approach	BFULL
9	-	-	SB
10	All baselines W/O Sampling	Majority Voting	MVAWOS
11	All baselines W/O Sampling	Naïve Bayesian Approach	BAWOS
12	All baselines With Sampling	Majority Voting	MVAWS
13	All baselines With Sampling	Naïve Bayesian Approach	BAWS

The pool of all 515 classifiers is also considered as the Full set. The set of classifiers which are trained using original training data and 19 different feature sets are also considered (MVAWOS, BAWOS). Furthermore, the best performing 19 classifiers for each feature set are combined using the Bayesian and majority voting approaches (BAWS, MVAWS). It is important to note that SB, MVFULL, MVAWS, and MVAWOS do not use development data at any stage of the ensembling process whereas all other ensembles use the development data set at some stage of ensemble constitution.

The classification results achieved by each ensemble described above are presented in Table 6.6.

Table 6.7: Performance of different MSCs on test data

MCS	Recall	Precision	F-score	No. Selected Classifiers
BPSO	84.97	86.61	85.78	49
MVPSO	83.58	86.68	85.1	51
BFS	83.79	84.31	84.04	58
MVFS	84.33	83.05	83.69	61
MVBE	78.28	82.43	80.30	176
BBE	79.41	81.60	80.49	188
MVFULL	77.25	75.92	76.58	515
BFULL	76.45	77.46	76.95	515
SB	80.11	83.26	81.65	1
MVAWOS	79.02	81.99	80.48	19
BAWOS	78.93	81.93	80.40	19
MVAWS	80.96	83.66	82.29	19
BAWS	80.57	83.69	82.10	19

As it can be seen from the above table, some of the ensemble approaches (BPSO, MVPSO, BFS, MVFS, MVAWS, and BAWS) outperform the single best classifier while the performance of other classifiers (MVBE, BBE, MVFULL, BFULL, MVAWOS, BAWOS) rank below the single best. However, regardless of the combination method used, all static classifier selection approaches except forward selection method perform worse than the single best classifier. Moreover combination of different fixed number of classifiers using MVFULL, BFULL, MVAWOS, BAWOS methods perform worse than the single best classifier in terms of F-score, clearly showing the significance of selecting an optimal subset of classifiers from the repository of initially created classifier (MVAWS and BAWS outperform the single best classifier). We first compare the performance of combination of classifiers without using any selection process to the single best classifier. The single best classifier achieves an F-score of 81.65% whereas the

performance of the combination of all 515 classifiers in the pool using majority voting and Naïve Bayesian combination approaches are 76.58% and 76.95% respectively, ranking at the bottom of list. In fact the performance degrades by 5.07% in the case of MVFULL and 4.7% for BFULL. This is mainly because the full set of classifiers include many low recall high precision classifiers due to classifiers generated using different under sampling ratios, which do not perform well. Thus, the combination of such classifiers is not expected to improve the performance. Similarly MVAWOS and BAWOS do not outperform the SB classifier, but the results achieved are slightly better. This is again due to the fact that classifiers which use full data, have low recall-high precision characteristics. When compared to the performance of the SB classifier as the baseline, MVAWOS and BAWOS systems have shown a degradation in performance by 1.17% and 1.25% respectively. Considering the MVAWS and BAWS systems when 19 classifiers whose performances have been increased using BUS are combined, it is seen that their performances are 82.29% and 82.1% for MVAWS and BAWS respectively. The improvement in classification achieved is mostly because of the nature of classifiers used for combination, where the recall values of such classifiers are improved after the undersampling process and the classifiers become more balanced in terms of recall-precision values compared to those which use original imbalanced data. It can also be seen that the combination method (majority voting vs Naïve Bayesian do not significantly affect the performance).

We next compare the effect of using classifier selection algorithms prior to combining the members of the selected ensemble. Employing the backward elimination approach for static selection of optimal subset of classifiers among the pool of 515 classifiers with different combination methods improves the performance

of the ensemble by 3.72% and 3.54% using MVBE and BBE methods respectively compared to MVFULL and BFULL methods. However, neither of the ensembles formed using BE outperform the performance of the SB classifier. Considering the number of selected classifiers by each method (176 classifiers using MVBE and 188 by means of BBE) it can be deduced that given the possibility to choose the optimal subset of classifiers, the combination may generate improved results regardless of the method used for combination.

It can be seen that, using forward selection as the selection strategy to find a well performing subset of classifiers from the pool can outperform the single best classifier by 2.04 % and 2.39 % using majority voting with 61 selected classifiers and Naïve Bayesian combination with 58 selected classifiers respectively. It can be concluded that the Naïve Bayesian approach acts as a better combination approach than the majority voting approach.

Overall, the proposed ensemble method using PSO as the selection approach and Naïve Bayesian method for combination (BPSO) achieved the best result among all other 11 ensemble approaches. The performance of the MCS produced by proposed BPSO scheme is 85.78% in F-score, which outperforms the single best classifier by 4.13 %. Comparing the performance of BPSO to BFS and BBE approaches, we can deduce that using a heuristic method to search a large solution space created by large number of classifiers (in our case the size of search space was (515 ^ 2) -1), is the best choice among other basic search approaches. Additionally, in order to test the effect of the Naïve Bayesian combination method on the selection process employed, we implemented the MVPSO system using majority voting. MVPSO has also outperformed the single best classifier's performance considerably. The classification

performance is 85.1% in terms of F-score, where the amount of improvement over SB classifier's performance is 3.45%. The parameters adjusted for the BPSO scheme are employed during the implementation of MVPSO as well.

It should be noted that, although the BPSO approach performs better than the MVPSO scheme by 0.68%, it cannot be deduced that using the Naïve Bayesian method will always reveal better results than using majority voting approach. This is due to the fact that in these types of ensemble schemes, the performance of the ensemble depends on the classifier selection criteria. If for example some parameters of the selection method such as the number of iterations or population size etc. change, it is possible to generate slightly different results.

6.4.1 Discussion on Classifiers Selected using Different Ensemble Schemes

The last column of Table 6.6 depicts the number of selected classifiers using each MCS approach. As mentioned in the previous section only the classifiers, whose performance was greater than 70% in terms of F-score, were included in the initial pool. By applying this constraint, the number of members in the repository decreased to the 515 from the possible 950. Moreover, this constraint lead to elimination of some classifiers which were trained using some of the feature sets described in Chapter 5. Table 6.7 illustrates the feature sets used in the classifiers selected by different ensemble schemes discussed.

Table 6.8: Feature sets used by different ensembles

	BPSO	MVPSO	BFS	MVFS	MVBE	BBE	MVFULL	BFULL	SB	MVAWOS	BAWOS	MVAWS	BAWS
F1	√	√	√	√			√	√		√	√	√	√
F2	√	√	√	√			√	√		√	√	√	√
F3	√	√	√	√	√	√	√	√	√	√	√	√	√
F4										√	√	√	√
F5										√	√	√	√
F6	√	√			√	√	√	√		√	√	√	√
F7	√	√			√	√	√	√		√	√	√	√
F8	√	√		√	√	√	√	√		√	√	√	√
F9	√	√		√	√	√	√	√		√	√	√	√
F10	√	√	√	√	√	√	√	√		√	√	√	√
F11	√	√	√	√	√	√	√	√		√	√	√	√
F12	√	√	√	√		√	√	√		√	√	√	√
F13										√	√	√	√
F14	√	√	√	√	√	√	√	√		√	√	√	√
F15										√	√	√	√
F16										√	√	√	√
F17										√	√	√	√
F18										√	√	√	√
F19										√	√	√	√

As it can be seen from the table, except for the last four ensembles (MVAWOS, BAWOS, MVAWS, and BAWS), all other ensemble approaches contain classifiers, which use only 11 out of the original 19 feature sets, (feature sets 1, 2, 3, 6-12, and 14). The last four ensemble schemes contain classifiers which uses all 19 feature sets, since they combine all individual classifiers with or without undersampled data.

We next compare the individual classifiers shared by different ensembles. Table 6.8 shows the percentage of shared classifiers between different MCS approaches. Combination of all classifiers (MVFULL, and BFULL) and the combinations of 19 baseline classifiers (MVAWOS, MVAWS, BAWOS, BAWS) are excluded from the table because for the full combination cases it is clear that 100% of other classifiers are shared since the combination of all classifiers is the superset.

Table 6.9: Percentages of classifiers shared between pairs of MCSs

	BPSO	MVPSO	BFS	MVFS	BBE	MVBE
BPSO		83.67% (41)	30.61% (15)	28.57% (14)	12.24% (6)	10.20% (5)
MVPSO			29.41% (15)	25.49% (13)	9.8% (5)	7.84% (4)
BFS				87.93% (51)	1.72% (1)	3.44% (2)
MVFS					3.27% (2)	3.27% (2)
BBE						57.38% (101)

The first observation that can be made from the table is the fact that the classifier selection algorithm is the primary deciding factor on the classifiers to be included in the final ensemble rather than the combination method. For example, comparing the BPSO and MVPSO ensembles we can see that 83.67% of the classifiers in the BPSO ensemble are the same as the classifiers in the MVPSO ensemble. However, the percentages of shared classifiers between BPSO and MVPSO and any other ensemble is not more than 31%. Similar arguments can be made for other ensembles, which use the same selection algorithm but different combination methods.

Secondly, the fact that the percentage of classifier shared between an ensemble, which uses the PSO algorithm for selection versus other selection schemes (FS and BE), is less than 30 % indicates that the PSO scheme is successful in selecting a diverse set of well performing classifiers in a large space.

Table 6.10 shows the shared selected classifiers using BPSO and MVPSO. The first column shows the feature sets used during classifier creation and the second column

shows the sampling ratios which is applied to training data using the feature sets mentioned.

Table 6.10: Shared classifiers between BPSO and MVPSO and the respective sampling ratios used

	Sampling Ratio
$\mathbf{F_1}$	5, 7, 14, 19, 21, 41, 0
$\mathbf{F_2}$	8, 11, 12, 21, 26, 27, 36, 0
$\mathbf{F_3}$	5, 7, 10, 16, 17, 27, 41, 0
$\mathbf{F_9}$	51
$\mathbf{F_{10}}$	14, 17, 19, 23, 28, 31, 0
$\mathbf{F_{11}}$	13, 20, 39, 28
$\mathbf{F_{12}}$	21, 37, 50
$\mathbf{F_{14}}$	15, 17, 30

It can be seen that, firstly only a few number of feature sets among all 19 are very effective in improving the classification performance (i.e. 8 out of 19 feature sets are used in the final ensemble). In addition, considering the distribution of sampling ratios for classifiers trained with undersampled data shows that the majority of the classifiers are sampled with sampling ratios in the range [10, 20]. It is important to note that the average imbalance ratio for the train data was found to be 24.42 as given in Table 5.1. This result may hint that only classifiers trained with undersampled data at sampling ratios below the average are well performing classifiers when grouped together in an ensemble. However, this points requires further investigation. In the second column, 0 represents training data without any undersampling. This fact validates our point for including the 19 classifiers trained with original data together with the 931 classifiers which were trained with undersampled data in the original ensemble of 950 classifiers.

6.5 Error Analysis

We have performed error analysis on the results obtained in order to gain an insight about the type of misclassifications made and the possible reasons that may account for the errors.

Table 6.11 shows the number of False Positives (FPs) and False Negatives (FNs) for each of the systems discussed in the previous section using the ChemDNER test data. It can be observed that for all methods the majority of false predictions are FNs resulting in lower recall. However, when the proposed system is compared to the SB system, it can be seen that there is a significant decrease in the number of FNs with a slight increase in FPs. 44% of all false predictions are classified as FPs and 56% as FNs for the case of the SB system as opposed to 47% FPs and 53% FNs for the BPSO method resulting in a more balanced classifier ensemble.

Further analysis of the train and test data provides some insight for reasons accounting for the false predictions. As mentioned in Chapter 5 we use the IOB2 tag scheme for representing each token as part of the entity ("B-EntityClass" if it is the first token of the entity and "I-EntityClass" if it is not the first token) and "O" to mark a token as a non-entity token.

The evaluation script used to calculate the performance of the classifiers however uses strict evaluation which requires that all parts of the entity are correctly recognised in order to classify it as a True Positive. However, analysis of the ChemDNER test data shows that around 5% of tokens are annotated by the annotators as both "B-EntityClass" and "I-EntityClass" depending on the position of the token. Typical examples are entity mentions which begin with a lower case letter.

Table 6.11: Number of FPs and FNs on ChemDNER Test Data

	TP	FP	FN
BPSO	20129	3112	3561
MVPSO	20561	3160	4039
BFS	20432	3802	3953
MVFS	19538	3988	3631
MVBE	19418	4139	5388
BBE	19647	4430	5094
MVFULL	17123	5431	5043
BFULL	17445	5076	5374
SB	19187	3858	4764
MVAWOS	18329	4026	4866
MVAWS	19981	3903	4699
BAWOS	18396	4057	4911
BAWS	18192	3545	4387

Such cases account for most of the FP mistakes since the classifier ensemle is sometimes not capable of distinguishing between the two cases as the entity mentions are the same except from their positions within the entity. We name such errors as Type I Errors. The first two rows of Table 6.10 show example sentences for this first case. Similarly around 4% of tokens are marked as both “I-EntityClass” and “O” class and 4.5% are marked as both “B-EntityClass” and “O” class by the annotators. The classifier is clearly unable to correctly predict such entity mentions which account for majority of the FN predictions. We name such errors as Type II Errors. Examples of Type II Errors are given in the last 4 rows of Table 6.12.

Table 6.12: Example Sentences for Type I and Type II Errors

Example Sentence	PubMed ID / Reference	Annotation	Error Type
"Alternatively, when TMS **diazomethane** is used as the dipole, the resulting 2-pyrazoline obtained after desilylation may be reduced with NaCNBH3 to provide the trans azakainate analog exclusively."	[23481645]/[217]	I-CHEMICAL	Type I
"The 1,3-dipolar cycloaddition of **diazomethane** with trans-dibenzyl glutaconate yields a 1-pyrazoline, which may be reduced directly to the pyrazolidine."	[23481645]/[217]	B-CHEMICAL	
"Similar concentrations of non-decolorized (unpurified, high **anthraquinone**) Aloe vera extracts tested in other studies have resulted in an increased incidence and severity of diarrhea and colon adenomas and carcinomas."	[23500775]/[218]	B-CHEMICAL	Type II
"Safety of purified decolorized (low **anthraquinone**) whole leaf Aloe vera (L) Burm."	[23500775]/[218]	O	
"Three PEMs are considered: Nafion, sulfonated **polystyrene** (sPS) that forms the hydrophilic subphase of segregated sPS-polyolefin block copolymers, and random sPS-polyethylene copolymer."	[23205740]/[219]	I-CHEMICAL	Type II
"The adsorption of human serum fibrinogen on **polystyrene** latex particles was studied using the microelectrophoretic and concentration depletion methods."	[23421850]/[220]	O	

Chapter 7

CONCLUSION AND FUTURE WORK

In this thesis, a novel framework for the chemical named entity recognition task using a machine learning approach is proposed. The proposed architecture consists of five different modules; data preprocessing, undersampling, feature extraction, classifier training, and classifier ensemble. Furthermore, a new approach is proposed for the first, second and the fifth module. For the data preprocessing module, a new rule based tokenizer, ChemTok [28], which is designed especially to be used in the chemical or biological domain, is proposed. The main idea behind the proposed tokenizer is to create longer discriminative tokens as much as possible while preventing incorrectly segmentation of text. ChemTok uses the rules extracted from the training data set in ChemDNER corpus.

Due to the imbalanced nature of the data used in the chemical NER problem, usually the classifiers trained are biased towards the majority classes. Since the data used in NER are sentences, making use of commonly used undersampling methods such as random undersampling in order to decrease class skewness is not very effective. Therefore, during this study a new undersampling strategy to be used in particular with NER is proposed. Since it is known that machine learning algorithms make use of information provided by surrounding samples around sample to be predicted, our proposed undersampling method mainly focuses on selecting negative samples from each sentence in such a way as to keep the positions of negative and positive samples

in the sentence, thus preserving the sentence structure as much as possible. Since it works in a systematic way to choose negative samples around each positive sample, achieving close to equal number of negatives on both left and right hand sides of each positive sample, we name it balanced undersampling. Experiments have shown that the performance of the classifiers trained using undersampled training data by means of BUS perform better for all kind of feature sets extracted.

As a third contribution of the thesis, a new ensemble scheme is proposed which uses particle swarm optimization algorithm as a heuristic population based approach to select a best performing subset of classifiers from a large pool of classifiers and combine the selected classifiers with Naïve Bayesian fusion approach. Classifiers in the pool are trained using the proposed undersampling method on training data with different feature sets. For each feature set, the training data is undersampled at different sampling ratios such that 515 classifiers are created. Investigation of the results achieved by each ensemble scheme shows that the proposed approach can outperform 11 other MCSs in terms of entity recognition performance.

Finally, using the proposed system a classification performance which ranks it within the six best systems that competed in BioCreative IV ChemDNER is achieved.

Future work includes making use of effective post processing methods to further improve classification performance. In addition, combination of the proposed architecture with other solutions provided for NER such as dictionary methods or rule-based strategies, may also be addressed as the next step. Moreover, using Bagging and Boosting [180] approaches with classifiers, which are trained using undersampled data, can be considered.

REFERENCES

[1] Cowie, J, & Lehnert, W. (1996). Information Extraction. *Communications of the ACM*, *39*(1), 80-91.

[2] Mucke, H. (2009). Data Mining in Drug Development and Translational Medicine. Insight Pharma Reports, Cambridge Healthtech Institute.

[3] Banville, D. L. (2006). Mining chemical structural information from the drug literature. *Drug Discovery Today*, *11*(1), 35-42.

[4] Cohen, K. B; & Hunter, L. (2008). Getting started in text mining. *PLoS Comput Biol*, *4*(1), e20.

[5] Yang, Y., Akers, L., Klose, T., & Yang, C. B. (2008). Text mining and visualization tools–impressions of emerging capabilities. *World Patent Information*, *30*(4), 280-293.

[6] Marsh, E., & Perzanowski, D. (1998, April). MUC-7 evaluation of IE technology: Overview of results. In *Proceedings of the seventh message understanding conference (MUC-7)* (Vol. 20).

[7] Grishman, R., & Sundheim, B. (1996, August). Message Understanding Conference-6: A Brief History. In *COLING* (Vol. 96, pp. 466-471).

[8] Nguyen, N., & Guo, Y. (2007, June). Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning* (pp. 681-688). ACM.

[9] Ananiadou, S., Friedman, C., & Tsujii, J. I. (2004). Introduction: named entity recognition in biomedicine. *Journal of Biomedical Informatics*, *37*(6), 393-395.

[10] Kim, J. D., Ohta, T., Tsuruoka, Y., Tateisi, Y., & Collier, N. (2004, August). Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications* (pp. 70-75). Association for Computational Linguistics.

[11] Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2015). CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminform*, *7* (Suppl 1), S1.

[12] Gurulingappa, H., Mudi, A., Toldo, L., Hofmann-Apitius, M., & Bhate, J. (2013). Challenges in mining the literature for chemical information. *RSC Advances*,*3*(37), 16194-16211.

[13] Warr, W. A. (2011). Representation of chemical structures. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *1*(4), 557-579.

[14] Eltyeb, S., & Salim, N. (2014). Chemical named entities recognition: a review on approaches and applications. *Journal of Cheminformatics*, *6*(1), 17.

[15] Klinger, R., Kolářik, C., Fluck, J., Hofmann-Apitius, M., & Friedrich, C. M. (2008). Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, *24*(13), 268-276.

[16] Townsend, J., Copestake, A., Murray-Rust, P., Teufel, S., & Waudby, C. (2005). Language technology for processing chemistry publications. In *Proceedings of the fourth UK e-Science All Hands Meeting* (Vol. 2005).

[17] Grego, T., Pesquita, C., Bastos, H. P., & Couto, F. M. (2012). Chemical entity recognition and resolution to ChEBI. *ISRN Bioinformatics*, *2012*.

[18] Hettne, K. M., Stierum, R. H., Schuemie, M. J., Hendriksen, P. J., Schijvenaars, B. J., Van Mulligen, E. M., & Kors, J. A. (2009). A dictionary to identify small molecules and drugs in free text. *Bioinformatics*, 25(22), 2983-2991.

[19] Lana-Serrano, S., Sanchez-Cisneros, D., Campillos, L., & Segura-Bedmar, I. (2013, October). Recognizing chemical compounds and drugs: a rule-based approach using semantic information. In *BioCreative Challenge Evaluation Workshop vol* (Vol. 2, p. 121).

[20] Vazquez, M., Krallinger, M., Leitner, F., & Valencia, A. (2011). Text mining for drugs and chemical compounds: methods, tools and applications. *Molecular Informatics*, *30*(6-7), 506-519.

[21] Akhondi, S. A., Hettne, K. M., van der Horst, E., van Mulligen, E. M., & Kors, J. A. (2014). Recognition of chemical entities: combining dictionary-based and grammar-based approaches. *Journal of Cheminform*, *7*(Suppl 1), S10.

[22] Dieb, M. (2013, October). Ensemble approach to extract chemical named entity by using results of multiple cner systems with different characteristic. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 162).

[23] Ravikumar, K., Li, D., Jonnalagadda, S., Wagholikar, K. B., Xia, N., & Liu, H. (2013, October). An ensemble approach for chemical entity mention detection and indexing. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 140).

[24] Tjong Kim Sang, E. F., & De Meulder, F. (2003, May). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 142-147).

[25] Leaman, R., Wei, C. H., & Lu, Z. (2015). tmChem: a high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics*, *7*(supplement 1).

[26] Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429-449.

[27] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01*

Proceedings of the Eighteenth International Conference on Machine Learning, pp.282-289, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[28] Akkasi, A., Varoğlu, E., & Dimililer, N. (2016). ChemTok: A New Rule Based Tokenizer for Chemical Named Entity Recognition. *Biomedical Research International*, vol. 2016, http://dx.doi.org/10.1155/2016/4248026.

[29] Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2015). CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminform*, *7*(Suppl 1), S1.

[30] Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2013, October). Overview of the chemical compound and drug name recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 2).

[31] Jiang, J., & Zhai, C. (2007). An empirical study of tokenization strategies for biomedical information retrieval. *Information Retrieval*, *10*(4-5), 341-363.

[32] Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 3). IEEE.

[33] Simpson, E., Roberts, S., Psorakis, I., & Smith, A. (2013). Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection* (pp. 1-35). Springer Berlin Heidelberg.

[34] Khabsa, M., & Giles, C. L. (2015). Chemical entity extraction using CRF and an ensemble of extractors. *Journal of Cheminform, 7*(Suppl 1), S12.

[35] Usié, A., Cruz, J., Comas, J., Solson, F., & Alves, R. (2015). CheNER: a tool for the identification of chemical entities and their classes in biomedical literature. *Journal of Cheminform, 7*(Suppl 1), S15.

[36] Lowe, D. M., & Sayle, R. (2014). LeadMine: A grammar and dictionary driven approach to entity recognition. *Journal of Cheminform, 7*(Suppl 1), S5.

[37] Karanikas, H., Tjortjis, C., & Theodoulidis, B. (2000, September). An approach to text mining using information extraction. In *Proc. Knowledge Management Theory Applications Workshop,(KMTA 2000).*

[38] Jurafsky, D., & Martin, J. H. [2009] Speech and Language Processing: An Introduction to Natural Language Processing, *Computational Linguistics, and Speech Recognition*.

[39] Malik, R. (2006). *CONAN: Text Mining in the Biomedical Domain* (Vol. 2006, No. 15). Utrecht University.

[40] Hersh, W. (2005). Evaluation of biomedical text-mining systems: lessons learned from information retrieval. *Briefings in Bioinformatics*, *6*(4), 344-356.

[41] Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2013, October). Overview of the chemical compound and drug name

recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop*(Vol. 2, p. 2).

[42] Simpson, M. S., Voorhees, E., & Hersh, W. (2014). Overview of the TREC 2014 Clinical Decision Support Track. In *Proc. 23rd Text Retrieval Conference (TREC 2014). National Institute of Standards and Technology (NIST).*

[43] Nédellec, C., Bossy, R., Kim, J. D., Kim, J. J., Ohta, T., Pyysalo, S., & Zweigenbaum, P. (2013, August). Overview of BioNLP shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop* (pp. 1-7).

[44] Hirschman, L., Yeh, A., Blaschke, C., & Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, *6*(Suppl 1), S1.

[45] Krallinger, M., Rabal, O., Lourenço, A., Perez, M. P., Rodriguez, G. P., Vazquez, M., & Valencia, A. (2015). Overview of the CHEMDNER patents task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop.*

[46] *Verspoor, K., Shatkay, H., Hirschman, L., Blaschke, C., & Valencia, A. (2014).* Summary of the BioLINK SIG 2013 meeting *at ISMB/ECCB 2013.Bioinformatics, btu412.*

[47] Krallinger, M., Erhardt, R. A. A., & Valencia, A. (2005). Text-mining approaches in molecular biology and biomedicine. *Drug Discovery Today*, *10*(6), 439-445.

[48] Erhardt, R. A., Schneider, R., & Blaschke, C. (2006). Status of text-mining techniques applied to biomedical text. *Drug Discovery Today*, *11*(7), 315-325.

[49] Chemical IUPAC naming guideline, available at: http://www.iupac.org/home/publications/e-resources/inchi.html

[50] Toropov, A. A., Toropova, A. P., Mukhamedzhanova, D. V., & Gutman, I. (2005). Simplified molecular input line entry system (SMILES) as an alternative for constructing quantitative structure-property relationships (QSPR). *Indian Journal of Chemistry Section A*, *44*(8), 1545.

[51] Author, A. (1964). International Union of Pure and Applied Chemistry. *Proceedings of the Society for Analytical Chemistry*, *1*(6), 73-74.

[52] Gibb, B. C. (2013). Bouquets, whiffs and pongs. *Nature Chemistry*, *5*(10), 805-806.

[53] Golden, J. M. (2008). Construing Patent Claims According to Their'Interpretive Community': A Call for an Attorney-Plus-Artisan Perspective. *Harvard Journal of Law and Technology*, *21*, 321.

[54] Chemistry. (n.d.). Abbreviations.com. Retrieved November 17, 2015, fromhttp://www.abbreviations.com/acronyms/CHEMISTRY/99999.

[55] Finkel, J. R., & Manning, C. D. (2009, August). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in*

Natural Language Processing: Volume 1-Volume 1 (pp. 141-150). Association for Computational Linguistics.

[56] Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security*, *8* (2), 339-344.

[57] Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, *32*(suppl 1), D267-D270.

[58] Li, Q., Cheng, T., Wang, Y., & Bryant, S. H. (2010). PubChem as a public resource for drug discovery. *Drug Discovery Today*, *15*(23), 1052-1057.

[59] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., & Wishart, D. S. (2014). DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, *42*(D1), D1091-D1097.

[60] Huang, M., Névéol, A., & Lu, Z. (2011). Recommending MeSH terms for annotating biomedical articles. *Journal of the American Medical Informatics Association*, *18*(5), 660-667.

[61] Pence, H. E., & Williams, A. (2010). ChemSpider: an online chemical information resource. *Journal of Chemical Education*, *87*(11), 1123-1124.

[62] Garten, Y., & Altman, R. B. (2009). Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinformatics, 10*(Suppl 2), S6.

[63] Alfonseca, E., & Manandhar, S. (2002, January). An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India* (p. 34).

[64] Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997, March). Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing* (pp. 194-201).

[65] McCallum, A., Freitag, D., & Pereira, F. C. (2000, June). Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML* (Vol. 17, pp. 591-598).

[66] Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004, July). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning* (p. 104). ACM.

[67] Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., & Wilks, Y. (1998, April). University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7).*

[68] Budi, I., & Bressan, S. (2003). Association rules mining for name entity recognition. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE 2003.* , pp. 325-328.

[69] Narayanaswamy, M., Ravikumar, K. E., & Vijay-Shanker, K. (2003, January). A biological named entity recognizer. *In Pacific Symposium on Biocomputing* (Vol. 8, pp. 427-438).

[70] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3-26.

[71] Neves, M., & Leser, U. (2014). A survey on annotation tools for the biomedical literature. *Briefings in Bioinformatics*, *15*(2), 327-340.

[72] Kim, J. D., Ohta, T., Tateisi, Y., & Tsujii, J. I. (2003). GENIA corpus—a semantically annotated corpus for bio text mining. *Bioinformatics*, *19*(suppl 1), i180-i182.

[73] Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., & Hunter, L. E. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, *13*(1), 161.

[74] The linguistic Data Consortium, LDC, available at: https://catalog.ldc.upenn.edu/LDC2008T20

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

[76] Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, 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.

[77] Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., & Woolsey, J. (2006). DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, *34*(suppl 1), D668-D672.

[78] Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (2000, January). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pac Symp Biocomput* (Vol. 5, pp. 514-25).

[79] Nobata, C., Dobson, P. D., Iqbal, S. A., Mendes, P., Tsujii, J. I., Kell, D. B., & Ananiadou, S. (2011). Mining metabolites: extracting the yeast metabolome from the literature. *Metabolomics*, *7*(1), 94-101.

[80] Kolárik, C., Klinger, R., Friedrich, C. M., Hofmann-Apitius, M., & Fluck, J. (2008). Chemical names: terminological resources and corpora annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference).*

[81] Corbett, P., & Copestake, A. (2008). Cascaded classifiers for confidence-based chemical named entity recognition. *BMC Bioinformatics*, *9*(Suppl 11), S4.

[82] Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., ... & Hirschman, L. (2008). Overview of BioCreative II gene normalization. *Genome Biology*, *9*(Suppl 2), S3.

[83] Leitner, F., Mardis, S., Krallinger, M., Cesareni, G., Hirschman, L., & Valencia, A. (2010). An overview of BioCreative II. 5. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(3), 385-399.

[84] Leaman, R., & Gonzalez, G. (2008, January). BANNER: an executable survey of advances in biomedical named entity recognition. *In Pacific Symposium on Biocomputing* (Vol. 13, pp. 652-663).

[85] Hanisch, D., Fundel, K., Mevissen, H. T., Zimmer, R., & Fluck, J. (2005). ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*, *6*(Suppl 1), S14.

[86] Wei, C. H., Harris, B. R., Kao, H. Y., & Lu, Z. (2013). tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, btt156.

[87] Hakenberg, J., Gerner, M., Haeussler, M., Solt, I., Plake, C., Schroeder, M., & Bergman, C. M. (2011). The GNAT library for local and remote gene mention normalization. *Bioinformatics*, *27*(19), 2769-2771.

[88] Heym, D. R., Siegel, H., Steensland, M. C., & Vo, H. V. (1976). Computer Recognition and Segmentation of Chemically Significant Words for KWIC Indexing. *Journal of Chemical Information and Computer Sciences,* 16(3), 171-176.

[89] Kemp, N., & Lynch, M. (1998). Extraction of information from the text of chemical patents. 1. identification of specific chemical names. *Journal of Chemical Information and Computer Sciences*, *38*(4), 544-551.

[90] Wilbur, W. J., Hazard Jr, G. F., Divita, G., Mork, J. G., Aronson, A. R., & Browne, A. C. (1999). Analysis of biomedical text for chemical names: a comparison of three methods. In *Proceedings of the AMIA Symposium* (p. 176). American Medical Informatics Association.

[91] The CAS registry file of substances, available at: http://www.chemie.fu-berlin.de/chemistry/chemdb/stn/registry. txt

[92] Corbett, P., Batchelor, C., & Teufel, S. (2007, June). Annotation of chemical named entities. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing* (pp. 57-64). Association for Computational Linguistics.

[93] Jessop, D. M., Adams, S. E., Willighagen, E. L., Hawizy, L., & Murray-Rust, P. (2011). OSCAR4: a flexible architecture for chemical text-mining. *J. Cheminformatics*, *3*(1), 41.

[94] Rocktäschel, T., Weidlich, M., & Leser, U. (2012). ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, *28*(12), 1633-1640.

[95] Segura-Bedmar, I., Martínez, P., & Segura-Bedmar, M. (2008). Drug name recognition and classification in biomedical texts: a case study outlining approaches underpinning automated systems. *Drug Discovery Today*, *13*(17), 816-823.

[96] Divita, G., Tse, T., & Roth, L. (2004). Failure analysis of MetaMap transfer (MMTx). *Medinfo*, *11*(Pt 2), 763-7.

[97] World Health Organization. (1988). Ethical criteria for medicinal drug promotion.

[98] Lu, Y., Ji, D., Yao, X., Wei, X., & Liang, X. (2015). CHEMDNER system with mixed conditional random fields and multi-scale word clustering. *Journal of Cheminformatics*, *7*(Suppl 1), S40.

[99] Batista-Navarro, R., Rak, R., & Ananiadou, S. (2015). Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. *Journal of Cheminform*, *7*(Suppl 1), S6.

[100] Huber, T., Rocktäschel, T., Weidlich, M., Thomas, P., & Leser, U. (2013, October). Extended feature set for chemical named entity recognition and indexing. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 88).

[101] Campos, D., Matos, S., & Oliveira, J. L. (2014). A document processing pipeline for annotating chemical entities in scientific documents. *Journal of Cheminform*, *7*(Suppl 1), S7.

[102] Tang, B., Feng, Y., Wang, X., Wu, Y., Zhang, Y., Jiang, M., & Xu, H. (2015). A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature. *Journal of Cheminformatics*, *7*(supplement 1).

[103] Munkhdalai, T., Li, M., Batsuren, K., Park, H. A., Choi, N. H., & Ryu, K. H. (2015). Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *Journal of Cheminformatics*, *7*(Suppl 1), S9.

[104] Ramanan, S., & Nathan, P. S. (2013, October). Adapting cocoa, a multi-class entity detector, for the ChemDNER task of BioCreative IV. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 60).

[105] Zitnik, S., & Bajec, M. (2013, October). Token-and constituent-based linear-chain crf with svm for named entity recognition. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 144).

[106] Irmer, M., Bobach, C., Böhme, T., Laube, U., Püschel, A., & Weber, L. (2013, October). Chemical named entity recognition with ocminer. In*BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 92).

[107] Xu, S., An, X., Zhu, L., Zhang, Y., & Zhang, H. (2014). A CRF-based system for recognizing chemical entity mentions (CEMs) in biomedical literature. *Journal of Cheminform*, *7*(Suppl 1), S11.

[108] Chemaxson provides cheminformatics software platforms, available at : www.chemaxon.com

[109] Sikdar, U. K., Ekbal, A., & Saha, S. (2013, October). Domain-independent model for chemical compound and drug name recognition. In *Proceedings of the Fourth BioCreative Challenge Evaluation Workshop* (Vol. 2, pp. 158-161).

[110] Choi, M., Yepes, A. J., Zobel, J., & Verspoor, K. (2013, October). Neroc: Named entity recognizer of chemicals. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 97).

[111] Lamurias, A., Ferreira, J. D., & Couto, F. M. (2014). Improving chemical entity recognition through h-index based semantic similarity. *Journal of Cheminform.*

[112] Li, L., Guo, R., Liu, S., Zhang, P., Zheng, T., Huang, D., & Zhou, H. (2013, October). Combining machine learning with dictionary lookup for chemical compound and drug name recognition task. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 171).

[113] Shu, C. Y., Lai, P. T., Wu, C. Y., Dai, H. J., & Tsai, R. T. H. (2013, October). A chemical compound and drug named recognizer for BioCreative IV. In *BioCreative Challenge Evaluation Workshop* (Vol. 2, p. 168).

[114] Ata, C., & Can, T. (2013, October). Dbchem: A database query based solution for the chemical compound and drug name recognition task. In *BioCreative Challenge Evaluation Workshop vol* (Vol. 2, p. 42).

[115] Kudo, T. (2005). CRF++: Yet another CRF toolkit. *Software available at http://crfpp. sourceforge. Net.*

[116] Campos, D., Matos, S., & Oliveira, J. L. (2013). Gimli: open source and high-performance biomedical name recognition. *BMC bioinformatics*, *14*(1), 54.

[117] Campos, D., Matos, S., & Oliveira, J. L. (2013). A modular framework for biomedical concept recognition. *BMC bioinformatics*, *14*(1), 281.

[118] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467-479.

[119] Kanerva, P., Kristofersson, J., & Holst, A. (2000, August). Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd annual conference of the cognitive science society* (Vol. 1036). Mahwah, NJ: Erlbaum.

[120] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[121] Coca is a dense annotator for biological text, available at : http://npjoint.com

[122] Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., & Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(D1), D1100-D1107.

[123] Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., & Tang, A. (2014). DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, *42*(D1), D1091-D1097.

[124] Wishart, D. S., Knox, C., Guo, A. C., Eisner, R., Young, N., Gautam, B., & Mandal, R. (2009). HMDB: a knowledgebase for the human metabolome. *Nucleic acids research*, *37*(suppl 1), D603-D610.

[125] Huang, R., Southall, N., Wang, Y., Yasgar, A., Shinn, P., Jadhav, A., & Austin, C. P. (2011). The NCGC pharmaceutical collection: a comprehensive resource of clinically approved drugs enabling repurposing and chemical genomics. *Science translational medicine*, *3*(80), 80ps16-80ps16.

[126] Zhu, F., Han, B., Kumar, P., Liu, X., Ma, X., Wei, X., & Chen, Y. (2010). Update of TTD: therapeutic target database. *Nucleic Acids Research*, *38*(suppl 1), D787-D791.

[127] Bolton, E. E., Wang, Y., Thiessen, P. A., & Bryant, S. H. (2008). PubChem: integrated platform of small molecules and biological activities. *Annual reports in computational chemistry*, *4*, 217-241.

[128] McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.

[129] Bolelli, L., Lu, X., Liu, Y., Jaiswal, A., Bai, K., Councill, I., & Garrison, B. (2007). ChemXSeer: a chemistry web portal for scientific literature and datasets. In *Open Repositories Conference, San Antonio, Texas*.

[130] Torii, M., Hu, Z., Wu, C. H., & Liu, H. (2009). BioTagger-GM: a gene/protein name recognition system. *Journal of the American Medical Informatics Association*, *16*(2), 247-255.

[131] Torii, M., Wagholikar, K., & Liu, H. (2011). Using machine learning for concept extraction on clinical documents from multiple data sources. *Journal of the American Medical Informatics Association*, *18*(5), 580-587.

[132] Shen, H., & Sarkar, A. (2005). Voting between multiple data representations for text chunking (pp. 389-400). *Springer Berlin Heidelberg*.

[133] Text Mining solutions, TEMIS, available at: http://www.temis.com/?id=103&selt=16.

[134] Rebholz-Schuhmann, D., Arregui, M., Gaudan, S., Kirsch, H., & Jimeno, A. (2008). Text processing through Web services: calling Whatizit. *Bioinformatics, 24*(2), 296-298.

[135] Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, *36*(2), 223-254.

[136] Yan, S., Spangler, W. S., & Chen, Y. (2012, August). Learning to extract chemical names based on random text generation and incomplete dictionary. *In Proceedings of the 11th International Workshop on Data Mining in Bioinformatics* (pp. 21-25). ACM.

[137] Mack, R., Mukherjea, S., Soffer, A., Uramoto, N., Brown, E., Coden, A., ... & Matsuzawa, H. (2004). Text analytics for life science using the unstructured information management architecture. *IBM Systems Journal*,*43*(3), 490-515.

[138] Wu, X., Zhang, L., Chen, Y., Rhodes, J., Griffin, T. D., Boyer, S. K., & Cai, K. (2010). ChemBrowser: a flexible framework for mining chemical documents. In *Advances in Computational Biology* (pp. 57-64). Springer New York.

[139] Hawizy, L., Jessop, D. M., Adams, N., & Murray-Rust, P. (2011). ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, *3*(1), 17.

[140] Tharatipyakul, A., Numnark, S., Wichadakul, D., & Ingsriswang, S. (2012). ChemEx: information extraction system for chemical data curation. *BMC Bioinformatics, 13*(Suppl 17), S9.

[141] Feng, C., Yamashita, F., & Hashida, M. (2007). Automated extraction of information from the literature on chemical-CYP3A4 interactions. *Journal of Chemical Information and Modeling*, *47*(6), 2449-2455.

[142] Dietterich, T. G. (1997). Machine-learning research. *AI magazine*, *18*(4), 97.

[143] Gams, M., Bohanec, M., & Cestnik, B. (1994, July). A schema for using multiple knowledge. In *Proceedings of the workshop on Computational learning theory and natural learning systems (vol. 2*), (pp. 157-170). MIT Press.

[144] Kuncheva, L. I. (2004). Combining pattern classifiers: methods and algorithms. John Wiley & Sons.

[145] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

[146] Opitz, D. W., & Shavlik, J. W. (1996). Generating accurate and diverse members of a neural-network ensemble. *Advances in Neural Information Processing Systems*, 535-541.

[147] Zenobi, G., & Cunningham, P. (2001). Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. In *Machine Learning: ECML 2001* (pp. 576-587). Springer Berlin Heidelberg.

[148] Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, *51*(2), 181-207.

[149] Ruta, D., & Gabrys, B. (2001). Analysis of the correlation between majority voting error and the diversity measures in multiple classifier systems. *In: Soft Computing and Intelligent Systems for Industry: Proceedings and Scientific Program : Fourth International ICSC Symposium 2001, Paisley, Scotland*, p. 50

[150] Shipp, C. A., & Kuncheva, L. I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, *3*(2), 135-148.

[151] Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140.

[152] Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification* (pp. 149-171). Springer New York.

[153] Mao, K. Z. (2004). Orthogonal forward selection and backward elimination algorithms for feature subset selection. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *34*(1), 629-634.

[154] Ruta, D., & Gabrys, B. (2001). Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *Multiple Classifier Systems* (pp. 399-408). Springer Berlin Heidelberg.

[155] Ruta, D., & Gabrys, B. (2005). Classifier selection for majority voting. *Information Fusion*, *6*(1), 63-81.

[156] Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33-57.

[157] Glover, F., Laguna, M., & Marti, R. (2007). Principles of Tabu search. *Approximation Algorithms and Metaheuristics*, *23*, 1-12.

[158] Palanisamy, S., & Kanmani, S. (2012). Artificial bee colony approach for optimizing feature selection. *International Journal of Computer Science*, vol. 9 (1), pp.432-438.

[159] Sahu, A., Panigrahi, S. K., & Pattnaik, S. (2012). Fast convergence particle swarm optimization for functions optimization. *Procedia Technology*, 4, 319-324.

[160] Lim, S. Y., Montakhab, M., & Nouri, H. (2002). A constriction factor based particle swarm optimization for economic dispatch. In *The 2009 European Simulation and Modelling Conference (ESM'2009)*.

[161] Biswas, S., Mandal, K. K., & Chakraborty, N. (2013). Constriction factor based particle swarm optimization for analyzing tuned reactive power dispatch. *Frontiers in Energy*, *7*(2), 174-181.

[162] Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (Vol. 1, pp. 84-88). IEEE.

[163] Simpson, E., Roberts, S. J., Smith, A., & Lintott, C. (2011). Bayesian combination of multiple, imperfect classifiers.

[164] Titterington, D. M., Murray, G. D., Murray, L. S., Spiegelhalter, D. J., Skene, A. M., Habbema, J. D. F., & Gelpke, G. J. (1981). Comparison of discrimination techniques applied to a complex data set of head injured patients. *Journal of the Royal Statistical Society. Series A (General)*, 145-175.

[165] Gu, Q., Cai, Z., Zhu, L., & Huang, B. (2008, December). Data mining on imbalanced data sets. In *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on* (pp. 1020-1024). IEEE.

[166] He, H., & Garcia, E. (2009). Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, *21*(9), 1263-1284.

[167] Menzies, T., DiStefano, J., Orrego, A., & Chapman, R. (2004). Assessing predictors of software defects. In *Proc. Workshop Predictive Software Models.*

[168] He, H., & Ma, Y. (Eds.). (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

[169] Wang, S. (2011). *Ensemble diversity for class imbalance learning* (Doctoral dissertation, University of Birmingham).

[170] Japkowicz, N. (2001). Concept-learning in the presence of between-class and within-class imbalances. In *Advances in Artificial Intelligence* (pp. 67-77). Springer Berlin Heidelberg.

[171] Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, *6*(1), 40-49.

[172] Chawla, N. V. (2003, August). C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML* (Vol. 3).

[173] Prati, R. C., Batista, G. E., & Monard, M. C. (2004). Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence* (pp. 312-321). Springer Berlin Heidelberg.

[174] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 321-357.

[175] Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing* (pp. 878-887). Springer Berlin Heidelberg.

[176] He, H., Bai, Y., Garcia, E., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on* (pp. 1322-1328). IEEE.

[177] Tomek, I. (1976). Two modifications of CNN. *IEEE Trans. Syst. Man Cybern. 6*, 769-772.

[178] Gowda, K. C., & Krishna, G. (1979). The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Transactions on Information Theory*, *25*(4), 488-490.

[179] Kubat, M., & Matwin, S. (1997, July). Addressing the curse of imbalanced training sets: one-sided selection. In *ICML* ,Vol. 97, pp. 179-186.

[180] Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, (3), 408-421.

[181] Laurikkala, J. (2001). *Improving identification of difficult small classes by balancing class distribution* (pp. 63-66). Springer Berlin Heidelberg.

[182] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, *6*(1), 20-29.

[183] Gu, Q., Cai, Z., Zhu, L., & Huang, B. (2008, December). Data mining on imbalanced data sets. In *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on* (pp. 1020-1024). IEEE.

[184] Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 1651-1686.

[185] Zhu, X. (2007, October). Lazy bagging for classifying imbalanced data. In*Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on* (pp. 763-768). IEEE.

[186] Tao, D., Tang, X., Li, X., & Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(7), 1088-1099.

[187] Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *39*(2), 539-550.

[188] Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In *Knowledge Discovery in Databases: PKDD 2003* (pp. 107-119). Springer Berlin Heidelberg.

[189] Mease, D., Wyner, A. J., & Buja, A. (2007). Boosted classification trees and class probability/quantile estimation. *The Journal of Machine Learning Research*, *8*, 409-439.

[190] Fan, W., Stolfo, S. J., Zhang, J., & Chan, P. K. (1999, June). AdaCost: misclassification cost-sensitive boosting. In *ICML* (pp. 97-105).

[191] Joshi, M. V., Kumar, V., & Agarwal, R. C. (2001). Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (pp. 257-264). IEEE.

[192] Gliozzo, A. M., Giuliano, C., & Rinaldi, R. (2005). Instance filtering for entity recognition. *ACM SIGKDD Explorations Newsletter*, *7*(1), 11-18.

[193] Maragoudakis, M., Kermanidis, K., Garbis, A., & Fakotakis, N. Dealing with Imbalanced Data using Bayesian Techniques. In *proceeding of fifth*

international conference on Language Resources and Evaluation, LREC 2006, pp. 1045-1050.

[194] Tomanek, K., & Hahn, U. (2009, September). Reducing class imbalance during active learning for named entity annotation. In *Proceedings of the fifth international conference on Knowledge capture* (pp. 105-112). ACM.

[195] Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (1999). Automatic extraction of rules for sentence boundary disambiguation. In *Proceedings of the Workshop on Machine Learning in Human Language Technology* (pp. 88-92).

[196] OpenNLP, A. Welcome to Apache OpenNLP. *2012. http://opennlp. apache. Org.*

[197] Webster, J. J., & Kit, C. (1992, August). Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on Computational linguistics-Volume 4* (pp. 1106-1110). Association for Computational Linguistics.

[198] Barrett, N., & Weber-Jahnke, J. (2011). Building a biomedical tokenizer using the token lattice design pattern and the adapted Viterbi algorithm. *BMC Bioinformatics*, *12*(Suppl 3), S1.

[199] Amino Acids. (n.d.). In Twenty Amino Acids. Retrieved from http://www.cryst.bbk.ac.uk/education/AminoAcid/the_twenty.html

[200] Chemical Affixes. In Affixes: The building block of English. Retrieved from http://www.affixes.org/themes/index.html

[201] Periodic table of elements. In Periodic table of elements: LANL. Retrieved from http://periodic.lanl.gov/downloads.shtml

[202] Akkasi, A., Varoğlu, E., Enhancement of Automatic Detection of Chemical Named Entities in Text using Random Undersampling. (2016), In *Proceeding 2nd International Conference on Information Technology Communications and Telecommunications, irICT2016,* Tehran, Iran. (pp. 1-9).

[203] Jain, A. K., & Chandrasekaran, B. (1982). Dimensionality and sample size consideration in pattern recognition practice in *Handbook of Statistics*.

[204] Takeuchi, K., & Collier, N. (2005). Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine*, *33*(2), 125-137.

[205] Ko, Y. (2012, August). A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval* (pp. 1029-1030). ACM.

[206] 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*(6), 423-435.

[207] Tsuruoka, Y. (2006). GENIA tagger: Part-of-speech tagging, shallow parsing, and named entity recognition for biomedical text. *Available at: www-tsujii. is. su-tokyo. ac. jp/GENIA/tagger*.

[208] Ando, R. K., & Zhang, T. (2005, June). A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 1-9).

[209] Suzuki, J., & Isozaki, H. (2008, June). Semi-Supervised Sequential Labeling and Segmentation Using Giga-Word Scale Unlabeled Data. In *ACL* (pp. 665-673).

[210] Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384-394).

[211] Miller, S., Guinness, J., & Zamanian, A. (2004, May). Name Tagging with Word Clusters and Discriminative Training. In *HLT-NAACL* (Vol. 4, pp. 337-342).

[212] Kennedy, J., & Eberhart, R. C. (1997, October). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997 IEEE International Conference on* (Vol. 5, pp. 4104-4108). IEEE.

[213] Abido, M. A. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, *24*(7), 563-571.

[214] Li-Ping, Z., Huan-Jun, Y., & Shang-Xu, H. (2005). Optimal choice of parameters for particle swarm optimization. *Journal of Zhejiang University Science A*, *6*(6), 528-534.

[215] Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, *6*(1), 58-73.

[216] Chuang, L. Y., Chang, H. W., Tu, C. J., & Yang, C. H. (2008). Improved binary PSO for feature selection using gene expression data. *Computational Biology and Chemistry*, *32*(1), 29-38.

[217] Wang, W., Simovic, D. D., Di, M., Fieber, L., & Rein, K. S. (2013). Synthesis, receptor binding and activity of iso and azakainoids. *Bioorganic & Medicinal Chemistry letters*, 23(7), 1949-1952.

[218] Shao, A., Broadmeadow, A., Goddard, G., Bejar, E., & Frankos, V. (2013). Safety of purified decolorized (low anthraquinone) whole leaf Aloe vera (L) Burm. f. juice in a 3-month drinking water toxicity study in F344 rats. *Food and Chemical Toxicology*, vol.57, 21-31.

[219] Lee, M. T., Vishnyakov, A., Gor, G. Y., & Neimark, A. V. (2012). Interactions of sarin with polyelectrolyte membranes: A molecular dynamics simulation study. *The Journal of Physical Chemistry B*, 117(1), 365-372.

[220] Lim, S. Y., Montakhab, M., & Nouri, H. (2002). A constriction factor based particle swarm optimization for economic dispatch. In *The 2009 European Simulation and Modelling Conference (ESM'2009)*.

[221] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, pp.37-63.

[222] Conditional Random fields, retrieved from https://en.wikipedia.org/wiki/Conditional_random_field

[223] He, X., Zemel, R. S., & Carreira-Perpiñán, M. Á. (2004, June). Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on* (Vol. 2, pp. II-695). IEEE.

[224] Sha, F., & Pereira, F. (2003, May). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 134-141). Association for Computational Linguistics.

[225] Chang, K. Y., Lin, T. P., Shih, L. Y., & Wang, C. K. (2015). Analysis and prediction of the critical regions of antimicrobial peptides based on conditional random fields. *PloS one, 10*(3), e0119490.

[226] Gupta, R. (2006). Conditional random fields. *Unpublished report, IIT Bombay.*

[227] Forney Jr, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE,61*(3), 268-278.

APPENDICES

Appendix A: Performance Evaluation for NER Systems

To evaluate the performance of NER systems, the three well known confusion matrix based measures are used: i) Precision, represents the ability of a system to detect only relevant entities, ii) Recall, shows the ability of a system to recognize all the relevant entities, iii) F-score, is a harmonic mean of precision and recall [221].

Computing the measurements involves following counts based on the given confusion matrix: 1) True Positive (TP), the number of positive samples correctly recognized. 2) False Negative (FN), the number of positive samples incorrectly recognized as negative. 3) True Negative (TN): the number of negative samples correctly recognized and 4) False Positive (FP), the number of negative samples incorrectly recognized. Then Precision and Recall can be calculated according to the Formulas A.1, and A.2 respectively:

$$Precision = p = \frac{TP}{TP+FP} \ , \qquad \text{(A.1)}$$

$$Recall = r = \frac{TP}{TP+FN} \qquad \text{(A.2)}$$

The general formula to compute F-Score for positive real β is as following:

$$F_\beta = (1+\beta^2).\frac{p.r}{(\beta^2.p)+r} \qquad \text{(A.3)}$$

where β shows the impact of precision over recall. Traditional F-Score or balanced F-Score is calculated for $\beta = 1$ and known as F_1.

Appendix B: Conditional Random Fields (CRFs)

Conditional random fields (CRFs) are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction. Whereas an ordinary classifier predicts a label for a single sample without regard to "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF popular in named entity recognition predicts sequences of labels for sequences of input samples [222].

CRFs are a type of discriminative undirected probabilistic graphical model. It is used to encode known relationships between observations and construct consistent interpretations. It is often used for labeling or parsing of sequential data, such as natural language text or biological sequences [222] and in computer vision [27]. Specifically, CRFs find applications in shallow parsing [223], named entity recognition [71], gene finding and peptide critical functional region finding [224], among other tasks, being an alternative to the related hidden Markov models (HMMs).

B.1 The CRF Model

Let $x_{1:\,N}$ be he observations (e.g. words in a sentence), and $y_{1:\,N}$ the hidden labels (e.g. tags or class labels). A linear chain CRF defines a conditional probability according to equation B.1:

$$p(y_{1:N}|x_{1:N}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{N}\sum_{j}^{F}\lambda_j f_j(y_{i-1}, y_i, x_{1:N}, i)\right) \quad \text{(B.1)}$$

The Scalar Z is the normalization factor to make it a valid probability. It defined as the sum of exponential number of sequences (B.2):

$$Z = \sum_{s_{1:N}} exp\left(\sum_{i=1}^{N} \sum_{j}^{F} \lambda_j f_j(y_{i-1}, y_i, x_{1:N}, i)\right) \quad \text{(B.2)},$$

where f () as feature function and λ as parameter of CRF, can take arbitrary real values, and the whole *exp* function will be non-negative. λ_i can be assume as weight of f_i ().

B.2 Feature Functions

The feature functions are the key elements of CRF. They take a sentence, the position of ith word in the sentence, the label of current word, and label of the previous word as input, $f\ (y_{i-1}, y_i, x_{1:N}, i)$. These are arbitrary functions that produce a real value; usually they are adjusted to produce binary values. Consider the following examples in the context of POS tagging problem:

A. $f_1(y_{i-1}, y_i, x_{1:N}, i)$ = 1 if y_i=ADVERB and the ith word ends in"-ly"; 0 otherwise. If the weight λ_1 associated with this feature is large and positive, then this feature is essentially telling that we prefer labeling where words ending in "-ly" get labeled as ADVERB.

B. $f_2(y_{i-1}, y_i, x_{1:N}, i)$=1 if i=1, y_i=VERB, and the sentence ends in a question mark; 0 otherwise. Again, if the weight λ_2 associated with this feature is large and positive, then labeling that assign VERB to the first word in a question are preferred. (E.g. "Is this yours?)

To build a conditional random fields, just feature functions and corresponding weights should be defined. The mentioned weights of feature functions will be determined during CRF training process.

B.3 CRF Training

Associated weight to feature functions can be found using different techniques such as: Gradient ascent, penalized log-likelihood criteria, pseudo log-likelihood, voted

perceptron, etc. [226]. They are different mainly in the objective function they try to optimize. Considering gradient ascent, assume there is a bunch of annotated sentences. Randomly initialize the weights to shift these randomly initialized weights to the correct ones for each training sentence:

- Go through each feature function f_i , and compute the gradient of the log probability of the training example with respect to λ_i :

  $\frac{\partial}{\partial \lambda_i} \log p(y \ \ |x \ \ ) = \sum_{i=1}^{m} f_j(y_{i-1}, y_i, x \ \ , i) - \sum_{y'} p(y' \ \ |x \ \ ) \sum_{i=1}^{m} f_j(y'_{i-1}, y'_i, x \ \ , i)$

  The first term in the gradient is the contribution of feature f_j under the true label, and the second term is the expected contribution of f_j under the current model.

- Move λ_j in the direction of the gradient:

  $\lambda_j = \lambda_j + \alpha[\sum_{i=1}^{m} f_j(y_{i-1}, y_i, x \ \ , i) - \sum_{y'} p(y' \ \ |x \ \ ) \sum_{i=1}^{m} f_j(y'_{i-1}, y'_i, x \ \ , i)]$, where α is some learning rate.

- Previous steps will be repeated until some stopping conditions is reached.

B.4 CRF Inference

After learning the weights of feature functions, learnt model should be applied on unlabeled samples to find optimum set of labels. The naïve way is to compute ***P(Y|X)*** for every possible label sequences, and then choose the label that maximizes this probability. This way is not rational, since there are T^m possible labels for a tag set of size T and sentence of length m. A better way is to use a dynamic programming algorithm to find the optimal labels, such as Viterbi algorithm [227] in HMM.

Appendix C: Details of Individual Classifiers

C.1 Results of applying BUS and RUS methods using different feature sets (F_{1-19}) on Development and Test data

R_s in the given tables in section D.1 shows the desired sampling ratios. The last one in each table (e.g. R_s = 51) shows the baseline classifier's performance without any undersampling.

Table C1.1: Effect of BUS and RUS on Development and Test data using Feature F1

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	86.52	60.04	70.89	79.55	60.16	68.51	85.95	60.37	70.92	78.43	61.06	68.66
3	85.02	68.15	75.66	81.28	65.68	72.65	84.13	68.52	75.53	80.11	66.00	72.37
4	84.75	71.60	77.62	79.8	69.24	74.15	83.88	72.04	77.51	79.03	70.37	74.45
5	83.86	74.33	78.81	80.47	72.64	76.35	83.07	74.84	78.74	79.17	73.30	76.12
6	83.84	75.59	79.5	81.15	72.25	76.44	83.01	75.81	79.25	80.13	72.94	76.37
7	83.3	76.88	79.96	79.54	74.74	77.07	82.52	77.09	79.71	78.76	75.61	77.15
8	82.59	77.79	80.12	79.65	75.52	77.53	81.65	78.02	79.79	78.71	75.96	77.31
9	82.06	78.50	80.24	79.88	75.45	77.60	81.03	78.64	79.82	79.04	76.45	77.72
10	82.32	79.26	80.76	79.58	76.31	77.91	81.10	79.21	80.14	78.42	77.41	77.91
11	82.33	77.53	79.86	78.10	75.62	76.84	80.89	77.28	79.04	76.85	76.49	76.67
12	82.10	80.12	81.1	79.60	77.32	78.44	81.05	80.29	80.67	78.54	78.14	78.34
13	80.59	80.33	80.46	78.40	77.59	77.99	79.16	80.13	79.64	75.97	77.12	76.54
14	82.30	80.63	81.46	78.83	78.41	78.62	81.09	80.81	80.95	77.41	79.27	78.33
15	81.39	80.51	80.95	79.00	78.60	78.80	80.42	80.71	80.56	76.62	78.02	77.31
16	80.72	81.14	80.93	77.89	77.30	77.59	79.76	81.46	80.60	76.96	78.26	77.60
17	80.86	81.11	80.98	78.19	77.58	77.88	79.53	81.32	80.42	77.06	78.23	77.64
18	81.11	81.03	81.07	78.29	79.04	78.66	79.83	81.26	80.54	77.01	79.43	78.20
19	81.3	81.43	81.36	78.09	79.19	78.64	80.15	81.75	80.94	76.97	80.05	78.48
20	80.99	81.31	81.15	78.01	79.18	78.59	79.68	81.53	80.59	77.11	79.77	78.42
21	81.66	81.61	81.63	78.86	79.23	79.04	80.45	81.97	81.20	77.58	80.07	78.81
22	80.83	81.34	81.08	78.05	79.22	78.63	79.69	81.77	80.72	76.70	79.86	78.25
23	81.35	81.81	81.58	78.27	79.47	78.87	80.14	82.02	81.07	77.03	80.16	78.56
24	80.88	81.47	81.17	78.29	79.97	79.12	79.66	81.72	80.68	75.91	79.25	77.54
25	81.22	81.44	81.33	78.38	79.83	79.10	79.95	81.63	80.78	77.18	80.36	78.74
26	80.92	82.00	81.46	78.25	79.80	79.02	79.56	81.86	80.69	76.87	80.45	78.62
27	78.60	79.95	79.27	77.83	74.62	76.19	77.47	80.60	79.00	77.14	74.65	75.87
28	81.01	81.89	81.45	77.92	80.02	78.96	79.76	82.27	81.00	76.99	81.03	78.96
29	80.72	82.03	81.37	78.17	79.75	78.95	79.42	82.35	80.86	77.09	80.38	78.70
30	79.96	81.66	80.80	76.59	78.76	77.66	78.80	81.69	80.22	75.53	79.74	77.58

Table C1.1 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
31	81.03	81.74	81.38	76.91	79.09	77.98	80.01	82.14	81.06	75.84	79.68	77.71
32	80.94	82.26	81.59	77.72	80.19	78.94	79.78	82.65	81.19	76.70	80.68	78.64
33	81.18	81.76	81.47	78.08	79.60	78.83	80.09	82.09	81.08	76.74	80.10	78.38
34	79.74	81.87	80.79	76.74	78.46	77.59	78.25	82.42	80.28	75.86	79.05	77.42
35	80.92	82.12	81.52	78.31	80.05	79.17	79.72	82.52	81.10	75.85	79.61	77.68
36	80.98	81.39	81.18	77.53	80.36	78.92	79.74	81.62	80.67	74.95	79.79	77.29
37	80.87	82.48	81.67	77.02	80.49	78.72	79.68	82.94	81.28	75.94	81.43	78.59
38	80.47	82.17	81.31	77.9	80.58	79.22	79.46	82.53	80.97	76.69	81.34	78.95
39	80.62	82.19	81.40	77.79	80.19	78.97	79.36	82.58	80.94	76.49	80.71	78.54
40	81.04	82.08	81.56	78.02	80.36	79.17	79.97	82.58	81.25	76.86	81.03	78.89
41	80.99	82.41	81.69	77.87	80.18	79.01	79.80	82.73	81.24	76.86	80.81	78.79
42	80.94	82.29	81.61	78.7	80.2	79.44	79.64	82.64	81.11	76.31	79.50	77.87
43	80.99	82.08	81.53	77.95	79.68	78.81	79.83	82.39	81.09	76.87	80.58	78.68
44	80.84	82.17	81.50	77.40	80.15	78.75	79.75	82.59	81.15	76.48	80.79	78.58
45	80.64	82.44	81.53	78.1	80.53	79.3	79.71	83.10	81.37	76.98	81.35	79.10
46	81.02	82.25	81.63	77.93	80.55	79.22	79.79	82.48	81.11	76.81	81.40	79.04
47	80.63	82.56	81.58	77.74	81.35	79.5	79.42	83.02	81.18	74.99	80.78	77.78
48	80.73	82.56	81.63	77.89	80.35	79.1	79.46	82.92	81.15	77.03	81.30	79.11
49	81.09	82.03	81.56	77.14	80.32	78.7	79.89	82.49	81.17	75.93	80.79	78.28
50	80.67	81.79	81.23	77.92	80.49	79.18	79.68	82.34	80.99	75.41	79.90	77.59
51	77.54	85.19	81.19	77.54	85.19	81.19	76.57	85.79	80.92	76.57	85.79	80.92

Table C1.2: Effect of BUS and RUS on Development and Test data using Feature F2

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	84.91	58.03	68.94	78.59	56.83	65.96	84.82	58.64	69.34	78.87	58.35	67.08
3	84.22	66.85	74.54	80.24	65.27	71.98	83.67	67.75	74.87	79.45	66.29	72.28
4	83.32	70.17	76.18	79.74	68.39	73.63	82.68	71.35	76.6	79.37	70.04	74.41
5	82.64	73.04	77.54	80.10	70.06	74.74	82.01	74.78	78.23	79.56	71.77	75.46
6	82.10	74.83	78.30	78.88	71.69	75.11	81.32	76.47	78.82	78.55	73.65	76.02
7	82.08	75.48	78.64	79.05	73.38	76.11	81.41	77.34	79.32	78.27	75.33	76.77
8	80.01	76.66	78.30	78.13	74.33	76.18	78.95	78.17	78.56	76.04	74.94	75.49
9	81.13	77.44	79.24	78.22	75.49	76.83	80.44	79.14	79.78	77.48	77.82	77.65
10	80.31	77.90	79.09	77.81	75.76	76.77	79.48	79.55	79.51	75.69	76.66	76.17

Table C1.2 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	Recall	Precision	F score	Recall	Precision	F score
11	80.9	78.48	79.67	77.91	76.33	77.11	80.06	80.05	80.05	76.99	78.79	77.88
12	80.37	78.76	79.56	77.44	76.12	76.77	79.73	80.79	80.26	76.77	78.34	77.55
13	79.82	79.26	79.54	77.06	76.67	76.86	79.12	81.17	80.13	76.27	78.86	77.54
14	79.84	79.82	79.83	77.2	77.1	77.15	78.94	81.6	80.25	76.48	79.59	78.00
15	79.85	79.74	79.79	76.92	77.61	77.26	79.09	81.6	80.33	76.18	79.79	77.94
16	79.82	79.59	79.70	76.80	77.05	76.92	79.12	81.50	80.29	75.84	79.14	77.45
17	79.83	80.05	79.94	76.49	78.77	77.61	78.93	82.02	80.45	75.6	80.87	78.15
18	80.34	79.09	79.71	77.09	77.97	77.53	79.15	80.65	79.89	74.87	78.82	76.79
19	79.52	80.58	80.05	76.62	78.47	77.53	78.69	82.44	80.52	75.81	80.88	78.26
20	79.69	80.00	79.84	77.04	78.30	77.66	79.13	81.97	80.52	76.26	80.79	78.46
21	79.12	80.96	80.03	76.92	78.29	77.6	78.34	82.72	80.47	74.95	79.29	77.06
22	79.37	80.83	80.09	76.50	78.85	77.66	78.38	82.47	80.37	74.38	79.57	76.89
23	78.95	80.68	79.81	76.96	78.63	77.79	78.29	82.64	80.41	74.58	79.44	76.93
24	79.45	80.61	80.03	76.37	78.81	77.57	78.66	82.38	80.48	75.70	81.16	78.33
25	79.67	80.58	80.12	76.46	79.46	77.93	78.97	82.62	80.75	75.61	81.52	78.45
26	79.46	80.97	80.21	76.89	78.95	77.91	78.70	82.77	80.68	76.01	81.17	78.51
27	79.62	80.71	80.16	75.72	78.90	77.28	78.69	82.52	80.56	74.92	80.82	77.76
28	79.99	80.02	80.00	76.34	78.60	77.45	79.12	81.62	80.35	75.69	81.08	78.29
29	79.06	81.33	80.18	76.00	78.74	77.35	78.11	83.03	80.49	75.40	81.07	78.13
30	79.32	80.90	80.1	76.73	79.48	78.08	78.41	82.68	80.49	74.36	80.05	77.10
31	79.49	80.78	80.13	76.48	79.23	77.83	78.73	82.72	80.68	75.62	81.43	78.42
32	79.43	80.92	80.17	76.45	79.05	77.73	78.71	82.82	80.71	75.94	81.47	78.61
33	79.75	79.29	79.52	74.91	77.66	76.26	78.55	80.79	79.65	74.27	79.93	77.00
34	79.25	81.07	80.15	76.39	79.13	77.74	78.60	83.07	80.77	75.95	81.66	78.70
35	79.18	81.03	80.09	76.44	78.93	77.67	78.63	83.19	80.85	75.73	81.48	78.50
36	79.08	81.75	80.39	75.60	78.59	77.07	78.23	83.42	80.74	75.11	80.85	77.87
37	79.15	80.83	79.98	75.97	79.64	77.76	78.42	82.79	80.55	75.3	81.91	78.47
38	79.36	81.00	80.17	76.06	79.21	77.60	78.75	83.06	80.85	75.47	81.83	78.52
39	78.90	81.36	80.11	76.26	79.27	77.74	78.23	83.28	80.68	75.73	81.70	78.6
40	79.06	80.96	80.00	76.49	79.54	77.99	78.18	82.76	80.40	74.42	80.48	77.33
41	78.77	81.38	80.05	59.83	69.85	64.45	78.00	83.18	80.51	59.75	72.27	65.42
42	79.30	81.33	80.30	76.36	79.18	77.74	78.53	83.18	80.79	75.70	81.66	78.57
43	79.07	81.05	80.05	76.67	79.42	78.02	78.29	82.79	80.48	74.49	79.99	77.14
44	79.02	80.76	79.88	76.12	79.45	77.75	78.29	82.30	80.24	74.03	80.40	77.08

Table C1.2 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	Recall	Precision	F score	Recall	Precision	F score
45	79.25	81.44	80.33	76.20	80.12	78.11	78.39	83.35	80.79	75.24	82.25	78.59
46	79.38	81.06	80.21	76.65	79.33	77.97	78.58	82.89	80.68	75.9	81.65	78.67
47	78.67	81.45	80.04	76.59	78.91	77.73	77.9	83.39	80.55	75.9	81.07	78.4
48	78.81	81.25	80.01	74.79	78.37	76.54	77.90	83.01	80.37	73.99	80.49	77.10
49	79.09	81.27	80.17	76.24	79.72	77.94	78.38	83.32	80.77	75.44	81.95	78.56
50	78.75	81.61	80.15	76.02	79.09	77.52	77.81	83.31	80.47	75.46	81.43	78.33
51	74.16	82.97	78.32	74.16	82.97	78.32	73.21	84.35	78.39	73.21	84.35	78.39

Table C1.3: Effect of BUS and RUS on Development and Test data using Feature F3

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**
2	86.67	59.79	70.76	80.47	59.08	68.14	86.42	60.08	70.88	80.18	60.37	68.88
3	85.29	67.21	75.18	82.46	65.68	73.12	84.98	67.38	75.16	81.36	66.14	72.96
4	85.09	71.42	77.66	81.65	69.76	75.24	84.33	72.09	77.73	80.83	70.76	75.46
5	84.41	74.07	78.90	81.20	72.23	76.45	83.51	74.76	78.89	80.16	73.32	76.59
6	83.56	76.11	79.66	80.08	73.56	76.68	82.76	76.77	79.65	79.24	75.01	77.07
7	83.15	76.12	79.48	80.53	74.08	77.17	82.54	76.87	79.60	79.73	75.22	77.41
8	83.08	77.38	80.13	79.9	74.93	77.34	82.55	78.44	80.44	79.21	76.22	77.69
9	82.99	78.03	80.43	79.87	76.01	77.89	82.19	78.48	80.29	78.89	77.43	78.15
10	81.92	78.98	80.42	79.70	76.50	78.07	81.31	79.54	80.42	77.65	76.33	76.98
11	82.07	79.31	80.67	79.59	77.06	78.30	81.26	79.68	80.46	78.47	78.14	78.30
12	81.61	79.49	80.54	78.52	76.72	77.61	80.77	80.14	80.45	77.62	77.97	77.79
13	81.79	79.98	80.87	79.04	77.37	78.20	80.75	80.45	80.60	78.10	78.33	78.21
14	81.94	80.27	81.10	78.83	77.85	78.34	81.10	80.97	81.03	78.17	79.06	78.61
15	81.03	79.48	80.25	77.41	76.53	76.97	80.14	80.01	80.07	76.47	77.83	77.14
16	81.45	81.72	81.58	78.64	78.93	78.78	80.49	82.29	81.38	77.78	80.00	78.87
17	79.39	81.11	80.24	77.24	77.14	77.19	78.47	82.07	80.23	76.13	78.04	77.07
18	80.97	80.81	80.89	78.68	78.71	78.69	80.11	81.49	80.79	77.54	79.82	78.66
19	81.40	80.90	81.15	77.30	78.17	77.73	80.14	81.61	80.87	76.34	79.17	77.73
20	81.92	81.03	81.47	78.51	79.29	78.9	80.93	81.64	81.28	77.60	80.43	78.99
21	81.02	81.63	81.32	78.45	78.92	78.68	79.95	82.26	81.09	77.62	80.10	78.84
22	81.28	80.98	81.13	78.38	79.60	78.99	80.57	81.88	81.22	77.41	80.77	79.05

Table C1.3 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**
23	81.00	81.61	81.3	78.73	79.63	79.18	80.06	82.30	81.16	77.73	80.58	79.13
24	81.42	81.61	81.51	78.34	79.69	79.01	80.43	82.20	81.31	77.37	80.94	79.11
25	81.19	81.51	81.35	78.18	79.95	79.06	80.12	82.12	81.11	75.83	79.76	77.75
26	78.05	80.06	79.04	80.59	72.66	76.42	77.02	81.00	78.96	80.11	72.70	76.23
27	81.48	81.71	81.59	78.20	79.71	78.95	80.39	82.33	81.35	77.19	80.80	78.95
28	80.53	81.47	81.00	77.07	78.56	77.81	79.80	82.29	81.03	76.08	79.61	77.80
29	80.96	82.01	81.48	77.43	77.64	77.53	80.09	82.61	81.33	76.86	78.66	77.75
30	81.13	81.95	81.54	78.48	71.23	74.68	80.14	82.69	81.40	77.40	71.37	74.26
31	81.09	82.08	81.58	78.23	79.77	78.99	80.25	82.85	81.53	77.35	80.96	79.11
32	80.91	81.75	81.33	76.92	78.78	77.84	79.73	82.41	81.05	76.22	80.15	78.14
33	80.40	80.06	80.23	76.47	78.89	77.66	79.42	81.03	80.22	75.46	80.26	77.79
34	81.61	81.17	81.39	77.99	80.10	79.03	80.66	82.00	81.32	76.94	81.48	79.14
35	81.20	81.71	81.45	78.03	79.55	78.78	80.37	82.40	81.37	77.08	80.6	78.80
36	80.58	81.66	81.12	76.71	78.94	77.81	79.43	82.19	80.79	75.92	80.13	77.97
37	81.98	79.43	80.68	76.37	78.48	77.41	81.13	80.08	80.60	75.69	79.88	77.73
38	80.78	81.83	81.30	78.2	79.86	79.02	79.79	82.50	81.12	75.78	79.62	77.65
39	80.83	81.97	81.40	78.29	79.97	79.12	79.86	82.67	81.24	77.31	81.16	79.19
40	80.99	82.06	81.52	78.29	79.74	79.01	79.79	82.69	81.21	76.09	79.56	77.79
41	81.02	82.48	81.74	78.17	80.21	79.18	80.11	83.26	81.65	77.25	81.41	79.28
42	80.79	82.25	81.51	77.86	80.61	79.21	79.86	83.07	81.43	75.69	80.37	77.96
43	81.12	82.26	81.69	75.04	78.37	76.67	80.26	83.13	81.67	73.98	79.21	76.51
44	81.02	82.15	81.58	78.20	80.52	79.34	80.15	83.06	81.58	77.28	81.80	79.48
45	80.63	82.43	81.52	77.00	79.08	78.03	79.51	82.91	81.17	76.09	80.29	78.13
46	78.13	80.24	79.17	69.00	77.24	72.89	77.29	81.60	79.39	68.82	75.99	72.23
47	80.49	81.53	81.01	77.61	80.07	78.82	79.40	82.08	80.72	75.36	79.70	77.47
48	80.74	82.74	81.73	77.75	80.59	79.14	79.7	83.35	81.48	76.7	81.53	79.04
49	81.41	81.81	81.61	78.14	80.42	79.26	80.43	82.68	81.54	77.35	81.81	79.52
50	80.97	82.27	81.61	77.97	80.27	79.10	79.96	83.02	81.46	77.22	81.61	79.35
51	77.29	84.57	80.77	77.29	84.57	80.77	76.45	84.96	80.48	76.45	84.96	80.48

Table C1.4: Effect of BUS and RUS on Development and Test data using Feature F4

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	64.57	47.11	54.48	51.56	42.27	46.46	64.77	48.10	55.20	52.38	43.69	47.64
3	55.90	57.47	56.67	57.46	50.02	53.48	56.41	59.54	57.93	57.32	50.44	53.66
4	62.1	62.05	62.07	55.88	61.78	58.68	62.18	63.42	62.79	55.83	63.81	59.55
5	60.45	66.79	63.46	57.65	65.39	61.28	60.55	68.87	64.44	56.89	66.72	61.41
6	59.87	68.12	63.73	56.29	66.41	60.93	59.48	70.08	64.35	55.27	67.76	60.88
7	54.40	67.54	60.26	55.65	60.83	58.12	53.97	70.03	60.96	55.15	62.30	58.51
8	53.70	67.44	59.79	47.55	60.66	53.31	53.40	69.40	60.36	47.72	63.00	54.31
9	57.51	70.56	63.37	54.63	67.49	60.38	57.04	72.75	63.94	54.51	70.30	61.41
10	49.51	65.93	56.55	45.37	59.32	51.42	49.64	68.17	57.45	45.15	60.34	51.65
11	58.58	71.49	64.39	54.58	69.75	61.24	58.32	74.00	65.23	54.36	72.46	62.12
12	57.60	71.94	63.98	48.48	65.6	55.76	57.46	74.55	64.90	48.32	67.84	56.44
13	56.33	70.66	62.69	54.04	69.08	60.64	55.63	73.55	63.35	52.85	71.35	60.72
14	55.61	70.49	62.17	53.56	64.99	58.72	55.52	73.12	63.12	53.22	67.01	59.32
15	58.28	72.24	64.51	54.08	70.50	61.21	58.15	74.57	65.34	54.11	73.18	62.22
16	39.70	64.02	49.01	36.30	64.61	46.48	40.28	65.79	49.97	36.35	66.30	46.96
17	49.22	67.79	57.03	47.65	64.10	54.66	49.40	68.55	57.42	45.63	67.69	54.51
18	55.98	72.12	63.03	46.46	62.45	53.28	55.78	74.53	63.81	46.48	64.86	54.15
19	57.68	72.38	64.20	51.82	66.81	58.37	57.32	74.65	64.85	51.64	69.19	59.14
20	48.66	67.08	56.40	9.27	43.82	15.30	48.27	69.08	56.83	9.35	46.11	15.55
21	55.06	70.91	61.99	51.62	70.06	59.44	55.13	73.76	63.10	51.31	72.35	60.04
22	46.92	68.25	55.61	43.39	65.50	52.20	46.76	70.48	56.22	44.63	68.38	54.01
23	58.12	72.29	64.44	49.6	69.08	57.74	57.94	74.69	65.26	49.86	71.89	58.88
24	54.05	70.85	61.32	50.63	69.95	58.74	53.63	73.34	61.96	50.20	72.04	59.17
25	57.06	71.89	63.62	53.47	69.84	60.57	56.61	74.03	64.16	53.31	72.57	61.47
26	57.39	73.36	64.40	51.79	68.54	59.00	57.32	75.67	65.23	51.66	71.46	59.97
27	55.27	69.79	61.69	51.03	69.62	58.89	54.26	74.32	62.73	52.20	70.75	60.08
28	56.90	72.37	63.71	53.86	71.15	61.31	56.44	74.63	64.27	52.27	72.55	60.76
29	54.62	70.88	61.70	52.53	68.16	59.33	54.37	73.31	62.44	52.55	70.64	60.27
30	56.61	73.36	63.91	49.25	63.93	55.64	56.31	75.73	64.59	49.56	66.61	56.83
31	54.85	72.58	62.48	52.04	70.34	59.82	55.02	74.68	63.36	51.61	73.2	60.54
32	57.61	73.42	64.56	24.69	60.69	35.10	57.44	75.62	65.29	24.52	61.76	35.10
33	56.33	73.08	63.62	50.23	69.64	58.36	55.96	75.39	64.24	49.91	71.95	58.94
34	56.05	73.39	63.56	53.13	70.77	60.69	55.69	75.28	64.02	53.39	73.79	61.95
35	56.40	73.44	63.81	50.49	69.45	58.47	55.76	75.44	64.12	50.48	72.29	59.45

Table C1.4 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
36	56.60	72.50	63.57	51.64	69.57	59.28	56.19	74.93	64.22	51.21	71.83	59.79
37	52.40	72.58	60.86	49.97	63.23	55.82	52.04	74.59	61.31	49.77	65.91	56.71
38	56.79	73.22	63.97	53.80	71.40	61.36	56.34	75.62	64.57	53.27	73.88	61.90
39	55.81	73.44	63.42	52.57	70.22	60.13	55.48	75.50	63.96	52.49	73.06	61.09
40	54.86	71.69	62.16	43.81	67.59	53.16	54.81	74.49	63.15	43.61	69.24	53.51
41	54.17	71.57	61.67	50.33	69.52	58.39	53.83	73.79	62.25	49.97	71.95	58.98
42	52.19	69.24	59.52	48.68	69.58	57.28	52.31	72.13	60.64	48.62	71.50	57.88
43	51.03	68.24	58.39	31.90	63.27	42.41	51.01	70.85	59.31	31.56	64.38	42.36
44	53.12	69.99	60.40	15.70	27.50	19.99	53.44	73.15	61.76	15.20	27.47	19.57
45	55.38	72.31	62.72	52.89	70.39	60.40	55.73	74.89	63.90	52.85	72.71	61.21
46	54.61	71.46	61.91	52.08	70.46	59.89	54.45	73.81	62.67	51.79	73.05	60.61
47	52.12	69.81	59.68	48.15	69.39	56.850	51.99	71.98	60.37	47.81	71.96	57.45
48	53.92	69.92	60.89	48.98	69.41	57.43	53.84	72.43	61.77	50.29	73.54	59.73
49	55.54	73.16	63.14	49.91	67.86	57.52	55.04	75.28	63.59	49.96	70.77	58.57
50	54.43	69.97	61.23	50.95	70.10	59.01	53.88	71.81	61.57	49.30	72.98	58.85
51	51.39	75.50	61.15	51.39	75.50	61.15	51.28	77.54	61.73	51.28	77.54	61.73

Table C1.5: Effect of BUS and RUS on Development and Test data using Feature F5

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	69.51	41.73	52.15	57.22	43.00	49.10	69.46	42.17	52.48	57.00	43.61	49.41
3	65.54	52.53	58.32	57.91	53.82	55.79	65.17	53.39	58.69	57.82	55.51	56.64
4	63.23	57.47	60.21	57.8	48.47	52.73	62.25	58.17	60.14	56.93	49.27	52.82
5	62.40	61.69	62.04	55.06	55.62	55.34	62.27	63.23	62.75	54.78	56.72	55.73
6	58.67	61.01	59.82	58.85	51.58	54.98	58.13	62.16	60.08	57.92	51.99	54.80
7	61.14	67.21	64.03	57.72	63.73	60.58	60.76	68.98	64.61	57.91	66.03	61.70
8	59.48	66.46	62.78	56.43	62.62	59.36	58.81	67.73	62.96	56.17	64.34	59.98
9	59.63	66.19	62.74	56.02	65.12	60.23	59.12	67.55	63.05	55.93	66.97	60.95
10	56.68	68.25	61.93	55.11	64.81	59.57	56.25	70.09	62.41	54.25	65.79	59.47
11	58.69	69.72	63.73	55.75	63.69	59.46	58.36	71.42	64.23	55.92	65.54	60.35
12	58.36	68.23	62.91	54.66	65.83	59.73	57.50	69.40	62.89	54.73	68.02	60.66
13	59.19	70.20	64.23	55.94	67.66	61.24	58.80	71.81	64.66	56.02	69.98	62.23

Table C1.5 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
14	58.57	70.34	63.92	55.61	65.24	60.04	58.27	72.30	64.53	55.11	66.95	60.46
15	57.61	68.79	62.71	53.83	66.66	59.56	57.38	70.61	63.31	53.60	68.67	60.21
16	59.62	70.84	64.75	49.80	64.56	56.23	59.17	72.48	65.15	49.60	66.36	56.77
17	56.65	69.06	62.24	51.46	66.92	58.18	56.08	70.54	62.48	51.45	69.30	59.06
18	56.52	68.45	61.92	53.90	65.26	59.04	55.65	69.68	61.88	53.67	67.47	59.78
19	58.06	71.15	63.94	54.08	68.94	60.61	57.80	73.04	64.53	53.97	71.28	61.43
20	58.44	71.35	64.25	55.00	69.51	61.41	57.93	73.07	64.63	54.81	71.78	62.16
21	56.69	69.89	62.6	53.07	66.66	59.09	55.74	71.38	62.6	52.91	68.75	59.80
22	57.76	69.73	63.18	53.88	68.22	60.21	56.98	70.80	63.14	53.75	70.50	61.00
23	58.14	71.29	64.05	47.97	63.77	54.75	57.83	73.29	64.65	48.15	65.56	55.52
24	58.10	68.99	63.08	54.21	68.49	60.52	58.13	71.11	63.97	53.95	70.45	61.11
25	56.44	70.56	62.72	54.18	68.56	60.53	56.33	72.54	63.42	54.10	70.78	61.33
26	55.16	70.05	61.72	28.54	37.81	32.53	54.93	72.40	62.47	27.39	37.25	31.57
27	56.50	69.12	62.18	53.35	65.43	58.78	56.40	71.08	62.89	52.96	67.25	59.26
28	56.96	69.03	62.42	53.41	69.26	60.31	56.57	70.33	62.70	52.74	71.06	60.54
29	55.45	68.27	61.20	45.38	62.6	52.62	55.24	70.13	61.80	44.82	63.92	52.69
30	55.28	69.50	61.58	50.78	68.31	58.25	55.07	71.44	62.20	51.78	71.64	60.11
31	54.70	70.02	61.42	41.94	58.27	48.77	54.46	72.22	62.10	41.44	59.38	48.81
32	58.62	71.91	64.59	55.20	70.43	61.89	58.48	73.85	65.27	55.53	72.94	63.06
33	54.15	69.67	60.94	53.82	59.41	56.48	53.93	72.05	61.69	53.09	60.40	56.51
34	58.24	71.91	64.36	45.38	60.97	52.03	57.74	73.59	64.71	45.45	62.89	52.77
35	54.49	69.51	61.09	24.43	55.29	33.89	54.69	71.97	62.15	23.58	56.59	33.29
36	52.60	67.55	59.14	51.5	61.95	56.24	52.59	69.92	60.03	51.47	63.45	56.84
37	57.10	69.97	62.88	53.55	68.02	59.92	57.12	71.85	63.64	53.21	70.24	60.55
38	54.76	69.67	61.32	50.43	66.17	57.24	54.48	71.89	61.99	50.23	67.84	57.72
39	56.13	68.19	61.58	43.74	64.10	52.00	55.66	69.68	61.89	43.32	65.50	52.15
40	55.12	70.64	61.92	49.36	66.52	56.67	54.93	73.06	62.71	49.38	68.20	57.28
41	55.59	70.16	62.03	52.48	68.34	59.37	55.60	72.28	62.85	52.46	70.75	60.25
42	53.12	67.18	59.33	50.95	63.97	56.72	52.85	69.23	59.94	50.60	65.37	57.04
43	56.15	68.28	61.62	45.60	59.13	51.49	55.86	69.97	62.12	45.21	60.42	51.72
44	55.86	67.30	61.05	51.93	65.48	57.92	55.65	69.24	61.71	52.2	67.78	58.98
45	57.18	72.06	63.76	39.18	66.70	49.36	57.03	74.05	64.44	38.48	68.13	49.18
46	51.21	65.40	57.44	48.35	62.78	54.63	51.04	67.34	58.07	48.13	64.06	54.96
47	57.24	70.17	63.05	53.40	66.35	59.17	57.19	72.38	63.89	53.51	68.71	60.16

Table C1.5 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
48	50.11	65.32	56.71	48.79	61.05	54.24	49.74	66.5	56.91	47.52	63.18	54.24
49	56.37	69.85	62.39	51.45	63.83	56.98	56.42	71.81	63.19	51.13	65.61	57.47
50	53.91	70.22	60.99	50.28	63.17	55.99	53.84	72.75	61.88	50.01	64.58	56.37
51	54.36	75.26	63.13	54.36	75.26	63.13	54.22	77.22	63.71	54.22	77.22	63.71

Table C1.6: Effect of BUS and RUS on Development and Test data using Feature F6

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	79.99	38.57	52.04	73.16	49.26	58.87	80.24	38.87	52.37	72.73	49.93	59.21
3	79.03	50.67	61.75	73.08	55.48	63.07	78.47	51.11	61.90	72.59	56.33	63.43
4	77.78	57.30	65.99	73.39	54.83	62.58	77.30	58.00	66.27	72.83	55.59	62.85
5	76.30	62.52	68.73	72.95	62.29	67.19	75.88	63.83	69.34	72.57	63.80	67.90
6	75.75	64.64	69.76	72.53	64.31	68.18	75.37	66.29	70.54	72.52	66.31	69.28
7	74.85	67.00	70.71	72.33	65.37	68.67	74.64	68.96	71.69	72.24	67.10	69.58
8	74.51	68.01	71.11	71.72	66.81	69.18	73.87	69.95	71.86	71.65	68.91	70.25
9	73.05	68.22	70.55	71.77	67.23	69.42	72.89	70.60	71.73	71.78	69.46	70.61
10	73.50	68.77	71.06	70.96	67.92	69.40	73.00	70.70	71.83	70.66	69.91	70.28
11	73.19	70.24	71.68	70.97	67.39	69.12	72.78	72.29	72.53	70.75	69.47	70.09
12	73.16	69.87	71.48	71.09	69.62	70.35	72.80	71.99	72.39	71.01	71.88	71.44
13	72.01	70.82	71.41	70.99	69.01	69.98	71.75	73.25	72.49	70.88	71.25	71.06
14	71.05	71.22	71.13	71.04	69.50	70.26	70.79	73.46	72.10	70.95	71.73	71.33
15	72.57	71.40	71.98	70.27	69.18	69.72	72.46	73.95	73.20	69.90	71.14	70.51
16	72.35	71.74	72.04	70.68	70.11	70.39	72.09	73.89	72.98	70.48	72.38	71.42
17	72.49	72.02	72.25	70.3	70.96	70.63	72.42	74.47	73.43	70.24	73.23	71.7
18	72.31	72.47	72.39	70.64	70.99	70.81	72.30	74.99	73.62	70.53	73.40	71.93
19	72.21	72.34	72.27	70.10	70.69	70.39	72.21	74.87	73.52	69.97	72.98	71.44
20	71.88	72.04	71.96	70.10	71.06	70.58	71.65	74.23	72.92	70.08	73.52	71.76
21	69.86	72.78	71.29	70.32	71.38	70.85	69.41	75.07	72.13	70.21	73.63	71.88
22	72.20	72.20	72.20	70.28	71.00	70.63	72.05	74.40	73.21	70.16	73.37	71.73
23	71.86	72.74	72.30	70.41	70.83	70.62	71.77	74.94	73.32	70.28	73.14	71.68
24	71.96	72.96	72.46	70.43	71.45	70.94	71.9	75.31	73.57	70.28	73.75	71.97
25	72.39	72.77	72.58	70.3	71.48	70.88	72.06	74.87	73.44	70.09	73.74	71.87
26	71.77	73.12	72.44	70.12	71.17	70.64	71.85	75.39	73.58	70.07	73.37	71.68

Table C1.6 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
27	71.10	72.88	71.98	63.54	69.66	65.98	71.15	75.13	73.09	63.45	71.91	66.93
28	71.70	73.17	72.43	70.06	70.88	70.47	71.74	75.56	73.6	69.83	72.96	71.36
29	71.70	73.46	72.57	70.17	71.79	70.97	71.59	75.71	73.59	70.08	74.22	72.09
30	70.68	71.31	70.99	70.25	71.82	71.03	70.21	72.93	71.54	70.09	74.06	72.02
31	71.69	73.50	72.58	69.77	71.75	70.75	71.8	75.96	73.82	69.64	74.01	71.76
32	71.29	73.76	72.5	70.12	71.70	70.90	71.07	76.01	73.46	70.02	73.99	71.95
33	71.57	73.21	72.38	70.11	71.72	70.90	71.18	75.15	73.11	69.87	73.81	71.79
34	71.87	73.50	72.68	69.71	71.74	70.71	71.83	75.88	73.80	69.54	74.03	71.71
35	69.14	70.94	70.03	69.5	71.88	70.67	68.8	72.60	70.65	69.4	74.16	71.69
36	71.34	73.31	72.31	69.89	71.86	70.86	71.33	75.78	73.49	69.85	74.26	71.99
37	71.85	73.41	72.62	69.82	71.96	70.87	71.88	75.91	73.84	69.67	74.26	71.89
38	71.11	72.33	71.71	70.23	71.90	71.06	70.92	74.52	72.68	70.05	74.24	72.08
39	71.95	73.42	72.68	70.22	71.79	70.99	71.71	75.73	73.67	70.09	74.07	72.03
40	71.21	73.78	72.47	69.74	71.39	70.55	71.26	76.34	73.71	69.37	73.41	71.33
41	71.80	73.59	72.68	68.98	69.35	69.12	71.75	75.95	73.79	68.81	71.38	70.02
42	71.77	73.36	72.56	70.00	72.25	71.11	71.78	75.70	73.69	69.99	74.66	72.25
43	71.36	73.87	72.59	69.82	71.49	70.64	71.49	76.42	73.87	69.59	73.62	71.55
44	70.72	72.46	71.58	70.02	72.12	71.05	70.32	74.33	72.27	69.90	74.51	72.13
45	70.75	73.87	72.28	70.14	72.00	71.06	70.72	76.14	73.33	70.04	74.36	72.13
46	71.31	73.72	72.49	68.82	70.95	69.86	71.54	76.32	73.85	68.51	72.94	70.64
47	71.22	73.90	72.54	70.00	71.72	70.85	71.39	76.53	73.87	69.9	74.08	71.93
48	71.13	73.92	72.50	69.45	72.10	70.75	71.15	76.36	73.66	69.39	74.55	71.87
49	71.46	74.05	72.73	70.07	72.07	71.05	71.61	76.63	74.04	69.89	74.41	72.07
50	68.91	70.65	69.77	69.67	72.16	70.89	68.49	72.37	70.38	69.63	74.56	72.01
51	67.35	76.87	71.8	67.35	76.87	71.8	67.50	79.38	72.96	67.50	79.38	72.96

Table C1.7: Effects BUS and RUS on Development and Test data using Feature F7

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	72.11	61.88	66.60	70.18	51.83	59.63	71.48	61.89	66.34	70.36	53.23	60.61
3	72.43	58.74	64.87	44.71	37.28	40.66	71.62	60.00	65.30	44.33	37.27	40.49
4	72.48	67.07	69.67	71.53	62.66	66.8	71.83	67.35	69.52	71.00	63.95	67.29
5	71.34	65.43	68.26	62.02	50.12	55.44	70.91	67.12	68.96	61.93	50.29	55.51

Table C1.7 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
6	70.48	67.06	68.73	68.07	64.45	66.21	69.92	69.15	69.53	69.19	64.91	66.98
7	70.47	67.42	68.91	67.47	62.11	64.68	70.37	69.9	70.13	67.22	62.27	64.65
8	72.83	69.66	71.21	69.54	68.36	68.94	72.28	70.34	71.30	67.84	69.34	68.58
9	68.58	68.60	68.59	66.23	64.71	65.46	68.08	70.77	69.4	67.24	66.57	66.9
10	73.05	67.91	70.39	67.99	67.61	67.8	69.74	72.06	70.88	69.54	67.12	68.31
11	72.73	69.12	70.88	67.78	68.00	67.89	72.1	69.73	70.9	67.18	70.40	68.75
12	72.54	69.79	71.14	67.81	68.62	68.21	72.08	70.61	71.34	66.07	71.37	68.62
13	69.2	70.71	69.95	69.6	65.59	67.54	69.18	73.62	71.33	70.38	65.99	68.11
14	69.12	71.13	70.11	63.22	56.03	59.41	69.08	73.65	71.29	63.08	56.33	59.51
15	68.79	71.23	69.99	66.74	66.50	66.62	68.8	74.18	71.39	65.95	66.92	66.43
16	72.00	66.55	69.17	66.33	67.46	66.89	71.37	67.22	69.23	64.31	69.48	66.8
17	71.25	67.95	69.56	66.23	64.86	65.54	70.66	68.75	69.69	65.04	66.22	65.62
18	72.66	68.93	70.75	67.45	70.00	68.7	69.06	74.32	71.59	70.55	68.18	69.34
19	73.62	70.75	72.16	68.48	71.55	69.98	73.11	71.65	72.37	67.36	73.20	70.16
20	68.28	71.21	69.71	68.17	66.27	67.21	68.45	74.23	71.22	69.26	66.95	68.09
21	73.69	69.84	71.71	67.11	70.07	68.56	73.12	70.74	71.91	67.17	72.89	69.91
22	72.48	68.40	70.38	65.62	70.48	67.96	66.89	74.68	70.57	68.97	67.55	68.25
23	72.85	69.59	71.18	66.21	70.62	68.34	72.53	70.35	71.42	65.74	73.25	69.29
24	73.43	68.60	70.93	66.43	70.53	68.42	67.98	74.94	71.29	70.10	67.68	68.87
25	68.36	71.99	70.13	67.12	68.44	67.77	68.29	74.92	71.45	67.84	68.95	68.39
26	68.1	71.92	69.96	48.44	43.96	46.09	67.88	74.56	71.06	48.55	44.34	46.35
27	73.04	71.19	72.10	66.83	70.54	68.63	72.44	72.03	72.23	66.76	73.15	69.81
28	67.49	70.46	68.94	59.88	48.96	53.87	66.94	72.56	69.64	59.61	49.18	53.90
29	73.46	67.73	70.48	65.38	70.67	67.92	68.52	75.14	71.68	71.34	66.78	68.98
30	73.84	69.73	71.73	66.28	71.00	68.56	73.28	70.36	71.79	64.84	73.74	69.00
31	72.04	71.40	71.72	66.74	70.23	68.44	71.46	72.05	71.75	65.24	73.38	69.07
32	73.08	71.26	72.16	66.87	71.08	68.91	72.41	72.03	72.22	66.85	73.82	70.16
33	73.79	69.65	71.66	56.35	57.32	56.83	73.34	70.63	71.96	55.97	58.95	57.42
34	70.71	70.41	70.56	66.65	70.42	68.48	67.67	74.73	71.02	67.28	69.84	68.54
35	73.52	68.50	70.92	66.96	70.71	68.78	68.52	75.03	71.63	70.13	67.90	69.00
36	68.33	72.12	70.17	71.79	64.34	67.86	67.62	74.23	70.77	71.01	64.84	67.78
37	68.46	70.37	69.40	66.90	62.44	64.59	68.01	72.69	70.27	66.3	62.84	64.52
38	67.87	73.13	70.40	72.43	64.39	68.17	67.86	75.89	71.65	71.97	64.87	68.24
39	67.97	72.37	70.10	71.68	64.36	67.82	68.10	75.54	71.63	72.99	66.55	69.62

Table C1.7 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
40	71.95	67.90	69.87	67.91	67.40	67.65	71.36	68.41	69.85	66.17	67.71	66.93
41	71.98	68.93	70.42	65.26	71.11	68.06	68.02	75.27	71.46	70.06	68.16	69.10
42	67.70	72.89	70.20	71.06	64.59	67.67	67.49	75.67	71.35	70.84	65.29	67.95
43	70.74	69.16	69.94	64.79	70.63	67.58	67.78	75.05	71.23	68.30	68.11	68.20
44	73.69	68.91	71.22	66.41	70.97	68.61	68.21	75.57	71.70	70.28	68.05	69.15
45	73.16	68.46	70.73	66.57	70.60	68.53	68.14	75.12	71.46	70.01	67.82	68.90
46	73.31	67.52	70.30	64.49	71.61	67.86	67.39	75.95	71.41	71.31	66.64	68.90
47	72.76	70.61	71.67	66.83	71.25	68.97	72.28	71.45	71.86	64.96	73.76	69.08
48	70.52	70.68	70.60	66.5	70.72	68.55	68.13	75.37	71.57	68.19	69.73	68.95
49	69.54	72.99	71.22	66.37	70.03	68.15	68.72	73.65	71.1	64.75	72.69	68.49
50	72.65	72.28	72.46	67.79	72.33	69.99	72.18	73.25	72.71	65.96	73.21	69.40
51	69.11	72.52	70.77	69.11	72.52	70.77	64.68	78.31	70.85	64.68	78.31	70.85

Table C1.8: Effect of BUS and RUS on Development and Test data using Feature F8

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	77.28	53.89	63.50	71.04	54.17	61.47	77.25	54.32	63.79	69.78	53.96	60.86
3	76.09	61.14	67.8	71.08	60.16	65.17	75.45	61.65	67.86	69.16	60.22	64.38
4	75.16	65.26	69.86	70.92	62.66	66.53	74.47	66.32	70.16	70.66	64.23	67.29
5	74.44	67.45	70.77	70.94	64.59	67.62	73.97	68.94	71.37	70.84	66.69	68.70
6	73.68	68.51	71.00	70.61	66.34	68.41	73.29	70.3	71.76	70.46	68.75	69.59
7	73.48	69.73	71.56	69.89	67.49	68.67	73.24	71.70	72.46	70.10	70.14	70.12
8	73.17	69.35	71.21	68.89	66.98	67.92	72.76	71.17	71.96	68.67	69.52	69.09
9	72.37	70.97	71.66	68.95	69.42	69.18	71.9	72.86	72.38	67.58	70.70	69.1
10	72.32	71.53	71.92	69.14	69.64	69.39	71.99	73.86	72.91	69.23	72.42	70.79
11	72.17	71.97	72.07	69.11	69.83	69.47	71.61	74.24	72.90	67.91	71.05	69.44
12	71.65	72.09	71.87	67.65	69.17	68.4	71.43	74.46	72.91	67.77	71.94	69.79
13	72.04	71.33	71.68	68.85	69.44	69.14	71.59	73.21	72.39	68.66	72.09	70.33
14	71.58	72.83	72.20	68.85	70.56	69.69	70.86	74.72	72.74	67.3	71.46	69.32
15	71.63	72.56	72.09	68.68	71.16	69.9	71.22	74.73	72.93	67.03	72.51	69.66
16	71.31	72.62	71.96	68.53	71.33	69.9	71.02	74.89	72.9	66.88	72.46	69.56
17	70.66	73.32	71.97	68.22	71.11	69.64	70.08	75.24	72.57	66.51	72.10	69.19
18	69.94	72.77	71.33	66.21	70.33	68.21	69.50	74.55	71.94	66.21	72.89	69.39

Table C1.8 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
19	71.25	73.74	72.47	68.67	71.33	69.97	70.94	75.87	73.32	68.21	73.37	70.7
20	70.99	73.49	72.22	67.21	70.34	68.74	70.64	75.56	73.02	66.97	72.96	69.84
21	70.79	73.11	71.93	66.56	70.55	68.5	70.32	74.92	72.55	66.65	73.49	69.90
22	71.29	73.77	72.51	68.06	71.96	69.96	71.20	75.91	73.48	68.16	74.68	71.27
23	71.14	73.90	72.49	67.9	72.22	69.99	70.97	75.97	73.38	67.84	74.75	71.13
24	71.05	73.38	72.2	67.93	71.77	69.80	70.75	75.45	73.02	67.92	74.37	71.00
25	70.94	74.23	72.55	67.32	72.14	69.65	70.84	76.40	73.52	66.57	73.92	70.05
26	71.45	73.54	72.48	68.08	70.30	69.17	71.24	75.62	73.36	67.54	72.22	69.80
27	70.86	74.27	72.52	67.26	71.72	69.42	70.7	76.44	73.46	67.10	73.88	70.33
28	70.85	73.84	72.31	67.17	68.55	67.85	70.79	76.03	73.32	66.70	70.44	68.52
29	70.34	74.21	72.22	67.86	72.41	70.06	69.9	76.48	73.04	66.13	73.53	69.63
30	70.87	74.38	72.58	68.08	72.29	70.12	70.69	76.48	73.47	67.93	75.05	71.31
31	71.10	73.80	72.42	68.27	71.65	69.92	70.92	76.11	73.42	67.67	73.76	70.58
32	70.68	73.76	72.19	67.88	72.28	70.01	70.52	76.12	73.21	67.80	74.86	71.16
33	70.81	74.20	72.47	67.71	72.26	69.91	70.68	76.54	73.49	67.58	74.91	71.06
34	70.61	73.69	72.12	67.47	72.22	69.76	70.36	75.99	73.07	67.13	74.48	70.61
35	70.92	73.64	72.25	66.53	71.19	68.78	70.83	76.07	73.36	66.53	73.91	70.03
36	70.42	74.73	72.51	66.28	71.36	68.73	70.07	76.99	73.37	65.85	73.39	69.42
37	70.74	74.47	72.56	58.75	68.98	63.46	70.47	76.73	73.47	58.2	71.20	64.05
38	70.53	74.82	72.61	66.93	71.77	69.27	70.22	77.04	73.47	66.54	74.09	70.11
39	70.74	74.31	72.48	67.70	72.47	70.00	70.60	76.67	73.51	67.76	75.32	71.34
40	69.95	72.85	71.37	66.04	71.05	68.45	69.12	74.35	71.64	65.80	73.45	69.41
41	70.58	74.72	72.59	67.94	72.64	70.21	70.47	77.20	73.68	67.81	75.38	71.39
42	70.38	74.37	72.32	66.57	71.35	68.88	70.27	76.88	73.43	66.42	73.92	69.97
43	70.73	73.37	72.03	66.16	71.40	68.68	70.46	75.45	72.87	66.06	73.96	69.79
44	70.43	74.52	72.42	67.43	72.04	69.66	70.4	76.88	73.5	67.02	74.64	70.63
45	70.54	74.54	72.48	67.75	72.62	70.10	70.33	76.91	73.47	67.63	75.32	71.27
46	70.54	74.57	72.50	27.57	23.58	25.42	70.65	77.11	73.74	28.88	24.86	26.72
47	70.34	74.58	72.40	67.61	72.81	70.11	70.20	77.03	73.46	67.54	75.53	71.31
48	70.22	74.39	72.24	66.93	71.79	69.27	69.95	76.56	73.11	66.60	73.94	70.08
49	70.57	74.71	72.58	67.68	72.48	70.00	70.47	76.95	73.57	67.49	75.11	71.10
50	70.66	74.81	72.68	67.24	72.63	69.83	70.45	77.12	73.63	67.01	75.37	70.94
51	65.71	77.53	71.13	65.71	77.53	71.13	65.48	79.71	71.9	65.48	79.71	71.90

Table C1.9: Effect of BUS and RUS on Development and Test data using Feature F9

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	75.23	66.83	70.78	75.66	56.67	64.80	75.80	68.65	72.05	75.76	57.99	65.69
3	78.44	72.01	75.09	75.62	62.13	68.21	78.67	74.00	76.26	75.47	63.61	69.03
4	77.79	74.21	75.96	75.49	60.37	67.09	78.12	76.12	77.11	75.58	61.82	68.01
5	75.91	71.44	73.61	74.74	67.71	71.05	76.47	73.92	75.17	74.49	69.83	72.08
6	73.57	74.96	74.26	74.65	69.17	71.81	74.12	77.31	75.68	74.68	71.74	73.18
7	77.11	75.27	76.18	74.54	70.01	72.20	77.48	77.36	77.42	74.45	72.71	73.57
8	73.97	71.92	72.93	70.11	70.58	70.34	73.84	74.50	74.17	70.4	72.75	71.56
9	73.69	72.41	73.04	68.33	73.72	70.92	73.09	74.92	73.99	68.19	75.36	71.60
10	73.49	76.22	74.83	73.42	71.95	72.68	73.98	78.55	76.20	73.4	74.46	73.93
11	73.89	72.95	73.42	70.99	71.45	71.22	74.40	75.26	74.83	70.54	74.07	72.26
12	75.16	76.89	76.02	73.22	73.40	73.31	75.41	79.12	77.22	73.05	76.12	74.55
13	72.47	76.40	74.38	71.48	71.18	71.33	72.87	79.03	75.83	72.56	74.86	73.69
14	73.01	76.86	74.89	71.68	72.43	72.05	73.22	79.28	76.13	71.67	75.00	73.30
15	73.38	75.25	74.30	71.50	72.82	72.15	74.17	77.91	75.99	70.87	75.41	73.07
16	71.80	75.81	73.75	69.80	73.19	71.45	72.48	78.31	75.28	70.94	75.74	73.26
17	73.47	75.78	74.61	71.23	73.82	72.50	74.35	78.69	76.46	70.79	76.32	73.45
18	74.12	75.93	75.01	71.08	73.25	72.15	74.64	78.54	76.54	70.56	75.83	73.10
19	71.72	76.10	73.85	70.2	73.52	71.82	72.15	78.86	75.36	69.85	75.78	72.69
20	73.27	77.35	75.25	70.65	74.02	72.30	73.45	79.58	76.39	70.27	76.54	73.27
21	73.07	75.67	74.35	71.03	73.25	72.12	73.76	78.36	75.99	70.61	75.90	73.16
22	73.10	77.31	75.15	70.85	75.58	73.14	73.44	79.60	76.40	70.26	77.90	73.88
23	71.86	74.60	73.20	67.44	75.23	71.12	71.43	77.39	74.29	66.49	77.81	71.71
24	72.72	77.29	74.94	70.47	73.92	72.15	72.92	79.53	76.08	70.06	76.86	73.30
25	72.46	75.07	73.74	66.25	72.83	69.38	71.97	77.92	74.83	66.54	75.03	70.53
26	74.77	76.10	75.43	71.44	72.35	71.89	75.58	78.75	77.13	70.74	74.33	72.49
27	72.96	76.73	74.8	70.06	74.49	72.21	73.48	79.15	76.21	69.60	76.99	73.11
28	72.89	77.07	74.92	70.70	74.14	72.38	73.41	79.63	76.39	70.41	76.78	73.46
29	71.33	76.94	74.03	68.60	72.95	70.71	71.98	79.18	75.41	69.26	76.42	72.66
30	74.80	76.57	75.67	72.11	74.85	73.45	75.51	78.98	77.21	71.66	77.35	74.40
31	72.13	75.06	73.57	67.50	74.89	71.00	71.94	77.57	74.65	67.85	77.36	72.29
32	71.19	77.15	74.05	69.57	73.47	71.47	71.69	79.46	75.38	70.53	76.15	73.23
33	72.17	76.01	74.04	68.86	75.02	71.81	71.57	78.61	74.92	67.75	77.53	72.31
34	71.94	75.58	73.72	64.27	74.59	69.05	71.42	78.15	74.63	64.21	76.83	69.96
35	71.95	75.44	73.65	37.34	46.12	41.27	71.54	78.04	74.65	37.43	46.99	41.67
36	70.74	77.19	73.82	68.52	74.29	71.29	71.10	79.62	75.12	68.16	76.96	72.29

Table C1.9 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
37	72.16	76.16	74.11	67.57	74.90	71.05	71.78	78.77	75.11	67.85	77.26	72.25
38	71.90	75.93	73.86	66.64	76.65	71.30	71.52	78.78	74.97	67.06	78.77	72.44
39	72.36	75.98	74.13	67.76	75.75	71.53	71.16	79.60	75.14	68.97	77.09	72.80
40	72.68	75.74	74.18	68.69	74.46	71.46	73.23	78.07	75.57	69.65	77.41	73.33
41	71.60	76.27	73.86	67.89	74.40	71.00	70.98	79.06	74.8	68.32	77.09	72.44
42	71.81	75.05	73.39	67.16	74.77	70.76	71.51	77.85	74.55	67.68	76.82	71.96
43	71.00	77.81	74.25	69.63	74.39	71.93	71.59	79.90	75.52	69.15	77.18	72.94
44	72.07	76.31	74.13	68.05	76.00	71.81	71.69	79.11	75.22	68.45	78.52	73.14
45	71.99	75.62	73.76	67.07	75.58	71.07	70.51	79.61	74.78	68.42	76.69	72.32
46	71.94	75.89	73.86	66.19	75.84	70.69	71.60	78.66	74.96	66.48	78.06	71.81
47	71.60	76.04	73.75	66.64	76.05	71.03	71.08	78.83	74.75	66.82	78.28	72.10
48	71.38	75.94	73.59	67.20	75.64	71.17	70.93	79.34	74.9	67.76	77.14	72.15
49	71.69	77.05	74.27	70.03	74.11	72.01	71.97	79.00	75.32	69.55	76.74	72.97
50	71.72	74.60	73.13	65.24	76.40	70.38	68.69	80.35	74.06	68.03	75.28	71.47
51	68.4	79.86	73.69	68.40	79.86	73.69	67.92	81.97	74.29	67.92	81.97	74.29

Table C1.10: Effect of BUS and RUS on Development and Test data using Feature F10

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	78.3	67.46	72.48	76.06	55.85	64.41	77.90	67.46	72.31	75.77	56.28	64.59
3	77.48	69.49	73.27	76.06	60.51	67.40	76.85	69.4	72.94	76.15	62.32	68.54
4	77.43	71.88	74.55	75.33	63.40	68.85	76.77	71.86	74.23	74.78	64.96	69.52
5	76.70	69.53	72.94	73.88	66.57	70.03	76.41	69.57	72.83	71.92	68.50	70.17
6	77.08	71.77	74.33	74.75	68.81	71.66	76.52	71.84	74.11	72.75	69.04	70.85
7	76.64	72.19	74.35	72.83	69.03	70.88	76.25	72.39	74.27	72.70	71.49	72.09
8	77.11	72.72	74.85	73.78	71.95	72.85	76.72	72.84	74.73	71.90	72.91	72.40
9	76.93	74.09	75.48	74.25	72.34	73.28	76.23	74.02	75.11	72.47	73.34	72.90
10	76.37	72.69	74.48	72.05	71.53	71.79	73.05	75.44	74.23	72.42	71.07	71.74
11	76.34	72.72	74.49	72.30	71.93	72.11	73.21	75.97	74.56	73.01	71.43	72.21
12	75.74	73.52	74.61	71.17	71.99	71.58	75.00	73.59	74.29	69.14	74.75	71.84
13	76.15	73.85	74.98	71.84	72.75	72.29	75.55	73.95	74.74	69.72	75.12	72.32
14	75.97	73.20	74.56	71.48	72.87	72.17	72.46	76.82	74.58	72.63	71.93	72.28

Table C1.10 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
15	75.54	73.68	74.6	71.59	71.60	71.59	75.01	73.92	74.46	70.88	73.80	72.31
16	76.27	73.32	74.77	70.70	73.24	71.95	75.69	73.34	74.50	68.87	75.80	72.17
17	76.09	73.54	74.79	71.60	72.11	71.85	75.45	73.64	74.53	69.50	74.50	71.91
18	72.84	74.89	73.85	71.52	71.26	71.39	72.36	77.31	74.75	72.49	71.32	71.90
19	75.03	73.85	74.44	70.66	72.58	71.61	71.86	76.28	74.00	71.21	72.21	71.71
20	75.87	73.39	74.61	70.96	73.76	72.33	71.89	77.85	74.75	72.32	72.05	72.18
21	75.51	72.82	74.14	70.78	70.91	70.84	74.86	72.94	73.89	69.97	72.64	71.28
22	75.09	73.19	74.13	70.26	72.99	71.60	74.79	73.30	74.04	68.10	74.84	71.31
23	75.90	73.77	74.82	71.25	74.02	72.61	72.23	77.98	74.99	72.38	72.42	72.40
24	75.70	73.55	74.61	70.84	74.07	72.42	71.58	77.95	74.63	72.07	72.20	72.13
25	76.01	73.94	74.96	70.19	73.60	71.85	75.6	74.07	74.83	69.58	75.89	72.60
26	74.47	73.34	73.90	68.59	74.21	71.29	71.03	78.29	74.48	72.32	71.80	72.06
27	75.93	73.56	74.73	70.55	74.24	72.35	71.44	78.10	74.62	72.40	72.29	72.34
28	75.4	73.27	74.32	69.59	73.94	71.70	72.12	78.03	74.96	73.19	71.83	72.50
29	72.87	75.19	74.01	72.39	70.47	71.42	72.08	77.71	74.79	73.25	72.00	72.62
30	71.9	75.49	73.65	69.1	68.81	68.95	71.49	78.00	74.60	68.79	68.70	68.74
31	75.72	72.90	74.28	70.29	72.49	71.37	75.39	73.18	74.27	69.49	74.18	71.76
32	75.85	72.99	74.39	71.03	73.46	72.22	71.84	77.22	74.43	72.11	71.50	71.80
33	75.90	74.05	74.96	70.92	74.05	72.45	75.19	74.22	74.70	68.55	76.44	72.28
34	72.23	76.08	74.11	71.25	71.76	71.50	71.74	78.62	75.02	72.13	71.68	71.90
35	75.58	73.48	74.52	70.25	74.38	72.26	70.96	78.11	74.36	71.98	72.15	72.06
36	76.02	73.62	74.8	71.26	74.07	72.64	72.18	78.06	75.00	72.43	72.24	72.33
37	74.96	74.34	74.65	71.33	73.96	72.62	72.12	78.03	74.96	72.72	72.78	72.75
38	75.90	73.13	74.49	69.54	74.56	71.96	71.98	78.51	75.1	73.76	71.70	72.72
39	75.86	73.59	74.71	70.68	74.40	72.49	71.5	78.42	74.8	72.18	72.18	72.18
40	72.48	75.41	73.92	72.24	70.96	71.59	71.9	78.05	74.85	73.17	71.01	72.07
41	72.73	75.19	73.94	72.14	70.96	71.55	71.97	77.39	74.58	72.99	71.09	72.03
42	75.31	72.86	74.06	68.84	74.64	71.62	71.37	78.72	74.87	73.38	71.63	72.49
43	74.93	72.44	73.66	68.99	73.04	70.96	71.14	76.68	73.81	71.39	71.13	71.26
44	72.59	76.11	74.31	72.77	70.87	71.81	71.99	78.50	75.1	73.55	70.9	72.20
45	75.69	72.84	74.24	68.79	76.06	72.24	75.3	73.16	74.21	66.35	76.23	70.95
46	71.96	76.08	73.96	71.93	71.14	71.53	71.4	78.75	74.9	72.78	71.16	71.96
47	75.65	72.96	74.28	69.09	74.62	71.75	71.58	78.88	75.05	73.62	71.60	72.60
48	75.79	72.45	74.08	69.08	74.25	71.57	71.36	78.26	74.65	73.68	70.97	72.30
49	75.84	73.30	74.55	70.70	73.69	72.16	75.20	73.49	74.34	68.26	75.69	71.78

Table C1.10 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
50	71.27	80.78	75.73	72.75	70.74	71.73	71.75	78.44	74.95	73.59	70.90	72.22
51	72.31	75.67	73.95	72.31	75.67	73.95	68.01	81.63	74.20	68.01	81.63	74.20

Table C1.11: Effect of BUS and RUS on Development and Test data using Feature F11

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	82.15	56.53	66.97	77.18	55.85	64.8	81.95	57.50	67.58	76.96	57.06	65.53
3	81.19	64.07	71.62	76.88	61.27	68.19	80.71	65.18	72.12	76.39	62.40	68.69
4	80.31	67.29	73.23	76.09	65.06	70.14	79.62	69.06	73.96	75.87	66.93	71.12
5	79.64	69.61	74.29	76.06	67.29	71.41	78.89	71.43	74.97	75.73	69.63	72.55
6	79.58	71.36	75.25	76.06	69.3	72.52	78.71	73.02	75.76	75.52	71.43	73.42
7	78.04	71.93	74.86	76.03	69.64	72.69	77.36	74.13	75.71	74.11	70.77	72.40
8	78.25	72.49	75.26	75.18	71.41	73.25	77.07	74.32	75.67	73.23	72.51	72.87
9	77.61	73.69	75.60	74.47	72.04	73.23	76.54	75.70	76.12	72.67	73.04	72.85
10	77.66	74.05	75.81	71.64	70.63	71.13	76.68	76.14	76.41	71.01	72.39	71.69
11	77.71	74.83	76.24	74.86	71.57	73.18	76.72	76.85	76.78	74.01	73.82	73.91
12	77.33	75.24	76.27	75.06	72.43	73.72	76.32	77.28	76.8	73.18	73.54	73.36
13	76.16	75.59	75.87	74.66	72.17	73.39	75.31	77.46	76.37	73.86	74.34	74.10
14	77.18	76.11	76.64	74.08	74.24	74.16	76.05	78.02	77.02	73.25	76.26	74.72
15	76.32	76.11	76.21	72.79	73.03	72.91	75.16	77.87	76.49	72.05	75.55	73.76
16	76.97	76.05	76.51	74.30	74.31	74.30	75.92	78.26	77.07	72.17	75.50	73.80
17	76.94	76.27	76.6	73.93	74.27	74.10	75.85	78.40	77.10	73.11	76.77	74.90
18	76.82	76.66	76.74	73.38	68.79	71.01	75.72	78.74	77.20	72.68	70.35	71.5
19	76.61	76.81	76.71	73.50	75.25	74.36	75.62	78.82	77.19	71.43	76.32	73.79
20	75.58	76.61	76.09	72.45	73.05	72.75	74.68	78.39	76.49	71.76	75.34	73.51
21	76.69	77.19	76.94	73.66	75.39	74.51	75.56	79.06	77.27	72.81	77.78	75.21
22	76.44	76.34	76.39	72.28	73.64	72.95	75.52	78.20	76.84	71.67	76.29	73.91
23	76.33	76.68	76.50	73.57	74.69	74.13	75.53	78.95	77.20	72.62	76.64	74.58
24	77.02	76.87	76.94	73.35	76.34	74.82	75.91	78.50	77.18	70.77	77.11	73.8
25	76.43	77.48	76.95	74.22	74.55	74.38	75.41	79.50	77.40	73.44	76.84	75.10
26	76.07	77.06	76.56	69.59	74.39	71.91	75.11	78.95	76.98	68.5	76.37	72.22
27	76.35	77.12	76.73	73.17	75.99	74.55	75.44	79.12	77.24	72.34	78.37	75.23

Table C1.11 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
28	76.72	77.28	77.00	73.3	74.59	73.94	75.60	79.04	77.28	72.35	76.52	74.38
29	76.11	77.30	76.70	72.95	75.70	74.30	75.21	79.41	77.25	72.13	78.05	74.97
30	76.54	77.08	76.81	73.14	74.96	74.04	75.15	78.40	76.74	72.30	77.24	74.69
31	76.27	77.88	77.07	73.12	75.90	74.48	75.05	79.76	77.33	71.09	76.82	73.84
32	76.32	77.62	76.96	73.41	75.76	74.57	75.29	79.60	77.39	72.63	78.23	75.33
33	76.43	77.26	76.84	72.99	75.68	74.31	75.39	79.14	77.22	72.31	78.29	75.18
34	75.57	77.98	76.76	73.3	76.01	74.63	74.46	79.63	76.96	70.63	76.62	73.50
35	76.13	77.72	76.92	72.23	74.57	73.38	75.34	79.99	77.6	71.18	76.19	73.60
36	76.08	77.73	76.9	72.56	75.16	73.84	74.99	79.57	77.21	71.86	77.53	74.59
37	76.09	77.59	76.83	72.54	76.27	74.36	75.07	79.52	77.23	71.73	78.75	75.08
38	76.2	77.31	76.75	73.38	76.02	74.68	75.24	79.45	77.29	71.05	77.05	73.93
39	74.84	77.17	75.99	72.07	74.38	73.21	73.89	78.74	76.24	71.25	76.92	73.98
40	75.77	77.90	76.82	73.35	76.09	74.69	74.83	79.98	77.32	70.89	77.11	73.87
41	75.83	77.92	76.86	73.59	75.51	74.54	74.83	79.90	77.28	72.55	77.66	75.02
42	75.49	77.20	76.34	72.66	75.30	73.96	74.62	79.14	76.81	70.80	76.58	73.58
43	75.26	78.28	76.74	72.9	75.82	74.33	74.09	80.67	77.24	72.14	78.44	75.16
44	75.96	77.78	76.86	72.84	76.43	74.59	74.97	80.09	77.45	71.96	78.90	75.27
45	76.25	77.51	76.87	73.55	75.73	74.62	75.28	79.85	77.5	71.38	76.90	74.04
46	76.28	78.10	77.18	73.36	75.92	74.62	75.22	80.09	77.58	72.34	78.29	75.2
47	75.79	77.99	76.87	71.62	73.73	72.66	74.82	80.18	77.41	70.77	75.34	72.98
48	74.7	77.99	76.31	71.55	74.55	73.02	73.37	79.7	76.4	70.81	77.06	73.8
49	75.4	78.44	76.89	73.50	74.26	73.88	74.13	80.28	77.08	72.86	76.38	74.58
50	75.63	77.43	76.52	72.89	74.92	73.89	74.65	79.49	76.99	71.94	76.93	74.35
51	72.47	80.83	76.42	72.47	80.83	76.42	71.55	82.69	76.72	71.55	82.69	76.72

Table C1.12: Effect of BUS and RUS on Development and Test data using Feature F12

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	77.47	54.46	63.96	72.19	53.99	61.78	77.55	55.16	64.47	71.95	55.11	62.41
3	76.33	62.28	68.59	71.78	59.75	65.21	76.12	63.38	69.17	71.37	61.35	65.98
4	76.17	64.56	69.89	71.52	60.90	65.78	75.63	65.78	70.36	70.94	61.96	66.15
5	74.96	67.76	71.18	71.57	63.73	67.42	74.72	69.77	72.16	71.26	65.23	68.11

Table C1.12 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
6	75.03	67.85	71.26	71.76	65.21	68.33	74.63	69.72	72.09	71.42	67.00	69.14
7	73.67	70.10	71.84	70.70	66.56	68.57	73.46	72.65	73.05	69.82	68.58	69.19
8	73.77	69.92	71.79	70.3	68.00	69.13	73.27	72.10	72.68	69.87	70.24	70.05
9	71.11	69.89	70.49	68.62	67.51	68.06	70.65	72.03	71.33	68.48	70.03	69.25
10	72.90	71.64	72.26	69.65	69.22	69.43	72.53	74.30	73.40	69.49	72.05	70.75
11	72.89	71.66	72.27	69.62	69.07	69.34	72.58	74.38	73.47	69.06	71.43	70.23
12	73.12	72.11	72.61	69.95	69.27	69.61	72.79	74.81	73.79	69.85	72.14	70.98
13	72.78	72.45	72.61	69.24	71.29	70.25	72.20	74.98	73.56	68.88	74.03	71.36
14	72.14	72.56	72.35	68.36	69.43	68.89	71.93	75.20	73.53	68.25	72.30	70.22
15	71.78	70.74	71.26	67.77	69.47	68.61	70.91	72.38	71.64	65.97	72.15	68.92
16	71.88	73.35	72.61	69.67	70.82	70.24	71.45	75.93	73.62	69.15	73.66	71.33
17	71.88	72.91	72.39	67.96	70.12	69.02	71.28	75.25	73.21	67.73	73.14	70.33
18	71.88	73.86	72.86	68.29	70.87	69.56	71.65	76.47	73.98	67.84	73.50	70.56
19	71.95	73.01	72.48	69.62	71.02	70.31	71.4	75.17	73.24	68.10	72.71	70.33
20	69.18	70.79	69.98	67.04	68.32	67.67	68.84	73.41	71.05	66.83	70.74	68.73
21	71.69	73.78	72.72	69.15	70.98	70.05	71.26	76.57	73.82	69.01	74.00	71.42
22	71.52	73.67	72.58	68.34	72.43	70.33	71.45	76.36	73.82	68.28	75.53	71.72
23	71.53	73.91	72.70	65.86	70.56	68.13	71.03	76.42	73.63	65.60	72.42	68.84
24	71.43	74.33	72.85	68.93	71.93	70.40	71.19	76.77	73.87	68.60	74.83	71.58
25	71.53	73.95	72.72	67.93	72.18	69.99	71.20	76.54	73.77	67.46	74.90	70.99
26	71.47	73.80	72.62	68.44	72.17	70.26	70.81	76.07	73.35	66.77	73.71	70.07
27	71.70	74.02	72.84	69.06	72.32	70.65	71.33	76.46	73.81	67.54	73.90	70.58
28	71.49	74.38	72.91	68.86	71.97	70.38	71.27	76.91	73.98	68.68	74.86	71.64
29	71.12	74.14	72.60	67.97	70.30	69.12	70.72	76.62	73.55	67.74	73.39	70.45
30	71.03	74.38	72.67	68.71	72.06	70.35	70.64	76.91	73.64	68.32	74.92	71.47
31	71.36	73.75	72.54	68.68	72.17	70.38	70.93	76.17	73.46	66.95	73.57	70.10
32	71.43	74.00	72.69	68.66	72.19	70.38	71.12	76.52	73.72	67.18	73.93	70.39
33	71.56	73.80	72.66	68.47	72.05	70.21	71.26	76.28	73.68	68.04	74.8	71.26
34	71.78	74.21	72.97	68.80	72.47	70.59	71.41	76.73	73.97	68.57	75.33	71.79
35	71.63	74.31	72.95	68.51	72.55	70.47	71.35	76.8	73.97	68.42	75.51	71.79
36	71.28	74.09	72.66	68.53	72.32	70.37	71.06	76.81	73.82	67.04	74.02	70.36
37	71.68	74.56	73.09	60.83	68.64	64.50	71.43	77.09	74.15	59.77	70.53	64.71
38	71.32	74.11	72.69	65.88	71.63	68.63	70.95	76.79	73.75	65.33	74.18	69.47
39	71.85	73.57	72.70	68.48	72.59	70.48	71.23	75.85	73.47	66.99	74.15	70.39
40	71.38	74.57	72.94	68.05	72.31	70.12	70.98	77.09	73.91	67.61	75.18	71.19

Table C1.12 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
42	71.22	74.14	72.65	67.02	72.25	69.54	70.80	76.61	73.59	66.78	75.19	70.74
43	68.72	72.80	70.7	66.54	70.14	68.29	68.58	75.82	72.02	67.06	71.61	69.26
44	71.09	74.72	72.86	68.56	72.36	70.41	70.86	77.39	73.98	68.18	75.16	71.50
45	71.54	74.01	72.75	67.10	71.00	68.99	70.99	76.29	73.54	66.81	73.88	70.17
46	71.53	74.65	73.06	68.64	72.52	70.53	71.48	77.32	74.29	68.49	75.48	71.82
47	70.83	74.41	72.58	68.66	72.39	70.48	70.66	77.21	73.79	67.09	74.01	70.38
48	71.15	74.03	72.56	68.12	72.88	70.42	71.11	76.88	73.88	67.88	75.94	71.68
49	71.30	74.47	72.85	67.76	71.85	69.75	71.18	77.22	74.08	67.38	74.62	70.82
50	70.47	75.27	72.79	68.92	72.30	70.57	69.58	77.20	73.19	67.22	73.76	70.34
51	67.99	77.28	72.34	67.99	77.28	72.34	65.10	78.30	71.09	65.10	78.30	71.09

Table C1.13: Effect of BUS and RUS on Development and Test data using Feature set F13

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	76.2	25.22	37.90	62.03	35.19	44.45	76.14	25.24	37.91	61.86	35.58	44.71
3	72.64	38.04	49.93	60.56	47.47	52.98	72.59	38.55	50.36	60.05	48.03	53.11
4	69.72	47.84	56.74	55.86	45.25	49.8	69.03	48.61	57.05	55.06	45.75	49.78
5	66.11	57.18	61.32	59.10	58.39	58.72	65.79	58.71	62.05	58.95	59.92	59.4
6	63.08	55.50	59.05	58.4	58.17	57.91	62.3	56.48	59.25	57.93	59.43	58.27
7	61.57	64.11	62.81	53.44	59.70	56.37	61.16	66.13	63.55	53.12	61.03	56.78
8	61.38	64.02	62.67	57.07	64.17	60.41	60.55	65.11	62.75	56.62	65.71	60.83
9	56.64	61.72	59.07	57.12	66.55	61.47	56.3	63.71	59.78	56.78	68.33	62.02
10	60.25	68.54	64.13	38.88	52.14	44.24	59.76	70.39	64.64	38.62	53.50	44.57
11	57.01	68.2	62.10	45.73	57.75	50.98	56.54	70.03	62.57	45.32	58.86	51.15
12	60.09	69.68	64.53	50.58	63.41	56.2	59.65	71.54	65.06	50.06	64.78	56.41
13	55.03	68.52	61.04	46.26	59.6	51.92	54.43	70.11	61.28	45.91	61.07	52.25
14	32.39	56.32	41.13	55.33	66.17	60.25	32.34	58.55	41.67	54.82	67.89	60.64
15	58.97	71.08	64.46	56.48	67.64	61.55	58.41	72.71	64.78	56.02	69.33	61.96
16	59.30	70.19	64.29	56.94	68.65	62.25	58.84	71.93	64.73	56.59	70.69	62.86
17	59.86	70.58	64.78	56.34	68.90	61.98	59.38	72.47	65.28	55.91	70.73	62.45
18	34.93	56.70	43.23	54.89	67.97	60.72	34.73	58.18	43.50	54.24	69.50	60.92
19	57.01	71.41	63.40	49.82	64.35	56.02	56.45	73.13	63.72	49.42	65.77	56.31
20	58.36	72.02	64.47	48.86	65.66	55.77	57.73	73.91	64.83	48.35	67.18	56.00

Table C1.13 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
21	57.85	70.34	63.49	54.37	68.62	60.65	56.94	71.58	63.43	53.72	70.22	60.85
22	57.37	72.92	64.22	52.15	67.96	58.98	57.00	75.16	64.83	51.56	69.58	59.20
23	54.75	65.98	59.84	56.27	62.47	58.01	53.89	67.42	59.90	55.90	64.10	58.44
24	58.33	71.13	64.10	50.88	65.69	57.33	57.80	73.33	64.65	50.14	66.90	57.31
25	51.32	70.89	59.54	42.78	63.55	49.86	50.25	72.16	59.24	42.40	65.21	50.14
26	56.93	72.51	63.78	51.83	68.40	58.86	56.15	74.30	63.96	51.35	70.27	59.23
27	44.25	68.29	53.70	41.37	58.50	48.39	44.24	69.95	54.20	41.09	59.92	48.68
28	58.86	72.49	64.97	55.33	62.02	57.33	58.33	74.42	65.40	54.8	63.48	57.60
29	49.29	63.59	55.53	48.56	67.32	56.16	48.74	65.01	55.71	47.97	68.77	56.27
30	46.32	67.09	54.80	44.13	60.99	50.93	45.72	68.51	54.84	43.54	62.25	50.98
31	50.01	67.14	57.32	52.54	69.15	59.68	49.73	69.07	57.83	52.04	70.94	60.00
32	48.31	65.73	55.69	49.43	67.05	56.41	47.95	67.03	55.91	48.91	68.73	56.65
33	57.74	72.22	64.17	52.01	67.85	58.84	56.94	73.89	64.32	51.52	69.56	59.15
34	54.72	68.06	60.67	43.43	60.05	49.45	53.87	69.69	60.77	42.88	61.25	49.53
35	50.68	64.81	56.88	48.34	65.72	55.42	50.14	66.72	57.25	47.82	67.36	55.65
36	48.72	70.55	57.64	54.08	68.11	60.24	48.07	72.15	57.7	53.47	69.70	60.47
37	57.47	72.36	64.06	53.52	69.09	60.29	56.88	74.08	64.35	52.86	70.65	60.44
38	54.23	68.75	60.63	51.46	67.45	58.37	53.49	70.28	60.75	50.74	68.86	58.42
39	56.84	72.86	63.86	55.44	70.46	62.05	56.15	74.57	64.06	54.84	72.21	62.34
40	56.27	72.24	63.26	50.02	67.97	57.51	55.45	73.66	63.27	49.5	69.55	57.73
41	57.85	72.59	64.39	51.79	68.86	59.06	57.32	74.51	64.79	51.22	70.62	59.33
42	56.32	73.21	63.66	45.5	62.40	52.24	55.56	74.82	63.77	45.07	63.97	52.52
43	43.42	67.28	52.78	45.49	65.50	53.10	43.05	68.63	52.91	45.14	67.22	53.44
44	58.97	70.53	64.23	51.53	67.88	58.56	58.35	72.42	64.63	50.81	69.37	58.63
45	57.88	70.34	63.50	52.8	68.48	59.56	57.23	71.97	63.76	52.15	70.04	59.72
46	56.71	72.74	63.73	51.86	69.77	59.45	56.15	74.61	64.08	51.26	71.40	59.64
47	56.22	72.51	63.33	53.97	67.81	60.11	55.65	74.45	63.69	53.29	69.54	60.34
48	52.35	69.45	59.70	53.16	68.68	59.89	51.38	70.89	59.58	52.53	70.37	60.11
49	58.29	72.64	64.68	54.51	69.03	60.92	57.65	74.63	65.05	53.90	70.86	61.23
50	52.70	68.98	59.75	54.95	70.66	61.81	51.74	70.22	59.58	53.98	72.04	61.71
51	52.43	75.32	61.82	52.43	75.32	61.82	52.13	77.36	62.29	52.13	77.36	62.29

Table C1.14: Effect of BUS and RUS on Development and Test data using Feature F14

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	82.25	51.78	63.55	79.11	46.94	58.92	80.50	50.25	61.88	77.16	45.96	57.61
3	81.87	57.86	67.80	77.05	54.69	63.97	80.41	56.32	66.24	75.33	53.68	62.69
4	80.44	60.85	69.29	75.42	56.41	64.54	78.45	59.08	67.40	73.27	54.53	62.53
5	77.18	62.17	68.87	77.45	58.42	66.60	75.23	60.60	67.13	74.80	56.25	64.21
6	79.12	64.44	71.03	75.46	61.54	67.79	76.42	62.21	68.59	72.55	59.15	65.17
7	80.22	65.56	72.15	74.69	39.08	51.31	77.92	63.74	70.12	71.24	35.75	47.61
8	79.19	67.41	72.83	74.04	54.39	62.71	76.89	65.61	70.8	71.66	52.67	60.71
9	77.30	66.97	71.77	73.48	64.94	68.95	74.48	64.86	69.34	71.22	63.45	67.11
10	78.61	68.78	73.37	74.33	67.79	70.91	76.23	66.95	71.29	72.02	66.55	69.18
11	78.51	69.22	73.57	73.87	63.72	68.42	75.82	67.12	71.21	71.32	62.12	66.40
12	78.52	68.62	73.24	74.16	66.39	70.06	76.08	66.60	71.03	72.15	65.28	68.54
13	77.93	69.69	73.58	72.93	64.27	68.33	75.31	67.64	71.27	70.76	63.14	66.73
14	77.47	70.92	74.05	73.87	68.81	71.25	74.98	69.08	71.91	71.47	67.19	69.26
15	76.61	71.43	73.93	72.13	68.76	70.40	73.95	69.25	71.52	69.88	67.31	68.57
16	76.69	69.69	73.02	72.75	59.13	65.24	73.87	67.70	70.65	70.46	57.26	63.18
17	76.57	72.22	74.33	73.52	64.09	68.48	74.01	70.17	72.04	71.23	62.29	66.46
18	76.2	71.62	73.84	71.82	69.33	70.55	73.85	69.70	71.72	69.19	67.71	68.44
19	71.20	67.97	69.55	74.89	60.90	67.17	68.59	66.07	67.31	72.54	58.99	65.07
20	77.5	69.43	73.24	71.64	68.06	69.8	74.61	67.15	70.68	69.43	66.77	68.07
21	75.7	72.6	74.12	72.92	70.93	71.91	73.33	70.71	72.00	69.26	68.05	68.65
22	75.29	67.04	70.93	67.82	67.08	67.45	72.92	64.9	68.68	65.93	65.28	65.60
23	75.63	72.82	74.2	72.85	70.74	71.78	73.38	70.87	72.1	70.48	69.21	69.84
24	75.24	74.13	74.68	70.45	72.64	71.53	72.94	72.13	72.53	68.01	70.95	69.45
25	76.02	71.52	73.70	70.37	71.09	70.73	73.40	69.42	71.35	68.21	69.38	68.79
26	73.97	75.79	74.87	71.87	62.86	67.06	71.54	73.6	72.56	70.14	62.03	65.84
27	75.54	69.81	72.56	70.47	69.77	70.12	72.27	67.72	69.92	66.62	68.39	67.49
28	76.16	72.86	74.47	69.83	65.92	67.82	73.86	70.98	72.39	68.38	64.74	66.51
29	75.80	72.95	74.35	72.77	69.42	71.06	73.42	70.91	72.14	69.82	67.42	68.60
30	75.37	73.95	74.65	72.40	70.93	71.66	73.00	71.88	72.44	70.06	69.49	69.77
31	74.93	73.37	74.14	70.65	70.68	70.66	72.32	71.13	71.72	68.38	68.94	68.66
32	73.18	72.44	72.81	70.55	70.36	70.45	70.66	70.39	70.52	67.81	68.57	68.19
33	74.15	74.76	74.45	67.97	72.66	70.24	71.67	72.59	72.13	66.33	70.86	68.52
34	75.39	71.98	73.65	69.34	68.54	68.94	72.96	69.75	71.32	67.27	66.79	67.03
35	72.17	72.11	72.14	65.47	69.56	67.45	69.74	70.34	70.04	63.79	67.05	65.38

Table C1.14 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
36	74.74	73.27	74.00	72.15	66.54	69.23	72.36	71.49	71.92	69.86	65.37	67.54
37	73.31	74.02	73.66	71.17	69.96	70.56	70.86	72.06	71.45	68.64	68.24	68.44
38	73.69	74.93	74.30	71.85	71.81	71.83	71.04	72.55	71.79	69.53	69.98	69.75
39	75.06	74.06	74.56	72.94	71.87	72.40	72.60	71.78	72.19	68.94	68.54	68.74
40	75.84	72.95	74.37	68.71	70.97	69.82	73.59	71.03	72.29	67.14	69.88	68.48
41	73.43	74.50	73.96	70.87	71.14	71.00	71.14	72.35	71.74	68.46	69.64	69.04
42	74.26	75.27	74.76	67.44	72.49	69.87	71.94	73.09	72.51	65.67	70.78	68.13
43	72.30	73.11	72.7	69.29	71.20	70.23	70.08	70.95	70.51	66.76	69.05	67.89
44	73.37	71.57	72.46	69.97	66.69	68.29	70.87	69.92	70.39	68.19	63.65	65.84
45	70.92	74.43	72.63	69.66	71.41	70.52	68.69	71.92	70.27	66.01	68.07	67.02
46	73.31	75.16	74.22	69.8	72.06	70.91	70.81	72.92	71.85	67.89	70.54	69.19
47	76.47	72.14	74.24	70.96	73.21	72.07	74.08	70.23	72.10	68.56	71.46	69.98
48	74.15	74.81	74.48	70.55	72.98	71.74	71.89	72.89	72.39	68.14	71.29	69.68
49	74.79	74.73	74.76	71.30	72.87	72.08	72.41	72.52	72.46	69.10	71.15	70.11
50	75.03	70.27	72.57	71.17	69.51	70.33	70.63	69.83	70.23	69.3	66.75	68.00
51	65.45	80.66	72.26	65.45	80.66	72.26	63.77	78.17	70.24	63.77	78.17	70.24

Table C1.15: Effect of BUS and RUS on Development and Test data using Feature F15

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	76.71	22.15	34.37	63.54	41.18	49.93	76.50	22.13	34.33	63.68	41.99	50.56
3	74.42	35.17	47.77	60.19	48.12	53.45	74.24	35.31	47.86	60.10	49.02	53.96
4	70.52	36.84	48.40	60.26	53.62	56.70	70.04	36.85	48.29	60.16	54.67	57.22
5	67.81	55.99	61.34	60.98	60.03	60.50	67.28	56.98	61.70	60.73	61.51	61.12
6	66.60	60.41	63.35	58.27	61.03	59.60	66.26	61.88	64.00	58.00	62.74	60.26
7	66.09	62.27	64.12	57.06	58.64	57.71	65.72	64.02	64.86	56.99	60.19	58.41
8	62.69	65.00	63.82	57.82	63.24	60.36	62.28	66.82	64.47	57.61	65.07	61.07

Table C1.15 (Continued)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
9	62.61	68.29	65.33	58.78	65.81	62.09	62.09	70.43	66.00	58.67	67.95	62.97
10	62.72	68.35	65.41	58.66	66.79	62.46	62.31	70.45	66.13	58.60	68.85	63.31
11	53.24	59.87	56.36	57.66	66.04	61.55	53.07	60.89	56.71	57.30	67.64	62.03
12	40.80	41.46	41.13	54.50	64.99	59.20	40.79	42.00	41.39	54.43	66.87	59.95
13	52.71	68.60	59.61	57.55	67.55	62.14	52.20	70.23	59.89	57.32	69.61	62.87
14	47.33	61.65	53.55	56.53	67.64	61.56	46.73	62.68	53.54	56.27	69.56	62.18
15	60.48	70.72	65.20	57.69	68.75	62.72	60.24	72.79	65.92	57.28	70.67	63.27
16	61.42	70.91	65.82	57.21	68.86	62.49	61.03	72.92	66.45	56.87	70.70	63.03
17	61.03	71.00	65.64	51.64	61.95	56.32	60.59	73.08	66.25	51.24	63.45	56.68
18	58.11	70.73	63.80	55.68	67.97	61.19	57.80	72.68	64.39	55.22	69.62	61.56
19	46.77	68.60	55.62	55.91	68.87	61.67	46.14	69.66	55.51	55.54	70.64	62.13
20	33.09	56.30	41.68	51.71	64.88	57.53	32.92	56.93	41.72	51.40	66.52	57.97
21	57.74	71.42	63.86	56.32	68.99	61.99	57.31	73.17	64.28	55.98	70.93	62.55
22	60.07	72.08	65.53	55.35	68.34	61.15	59.70	74.19	66.16	54.92	70.15	61.60
23	51.77	69.68	59.40	57.05	70.11	62.90	50.90	70.80	59.22	56.6	71.93	63.34
24	57.35	70.48	63.24	57.36	69.75	62.95	57.01	72.30	63.75	57.07	71.78	63.58
25	59.35	71.13	64.71	56.81	69.63	62.56	58.83	72.84	65.09	56.61	71.59	63.21
26	56.75	71.54	63.29	58.04	70.83	63.8	56.45	73.4	63.82	57.57	72.61	64.22
27	49.23	63.24	55.36	54.47	68.57	60.65	48.91	64.91	55.79	53.99	70.26	61.00
28	56.34	69.83	62.36	54.74	68.59	60.88	55.74	71.19	62.52	54.41	70.51	61.42
29	58.54	71.7	64.46	56.84	70.64	62.97	58.27	73.61	65.05	56.28	72.43	63.33
30	55.43	68.12	61.12	56.08	69.29	61.97	54.98	69.73	61.48	55.68	71.22	62.48
31	57.96	71.74	64.12	48.12	66.29	55.32	57.64	73.43	64.58	47.81	67.99	55.71
32	59.07	72.35	65.04	55.56	69.67	61.78	58.63	74.08	65.46	55.14	71.43	62.2
33	59.47	71.91	65.10	54.98	69.27	61.29	59.32	74.22	65.94	54.48	70.97	61.63
34	58.85	72.58	65.00	47.02	64.39	54.13	58.74	74.62	65.73	46.89	66.09	54.67
35	59.84	72.85	65.71	53.15	68.6	59.84	59.29	74.64	66.09	52.67	70.19	60.14
36	60.18	72.61	65.81	47.71	65.46	55.08	59.88	74.56	66.42	47.25	67.08	55.34
37	60.05	72.91	65.86	35.55	47.96	40.23	59.64	74.97	66.43	35.31	49.34	40.55
38	46.48	65.70	54.44	53.73	68.65	60.24	46.24	67.16	54.77	53.37	70.46	60.70
39	58.86	72.20	64.85	48.92	66.03	55.88	58.57	74.28	65.50	48.59	67.70	56.26
40	48.97	67.25	56.67	48.90	65.99	55.98	48.44	68.66	56.8	48.66	67.53	56.39
41	49.42	60.80	54.52	55.20	69.76	61.62	49.12	62.12	54.86	54.66	71.45	61.93
42	43.34	70.01	53.54	42.46	58.39	47.97	42.87	71.44	53.58	42.14	59.57	48.23
43	58.48	72.27	64.65	51.09	65.38	57.28	58.09	74.03	65.1	50.92	67.13	57.85

Table C1.15 (Continued)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
44	58.40	73.00	64.89	56.81	70.69	62.99	58.00	74.97	65.40	56.40	72.38	63.39
45	57.98	72.95	64.61	41.13	62.27	48.51	57.47	74.76	64.98	40.96	63.64	48.85
46	52.36	69.88	59.86	54.8	69.92	61.42	51.79	71.63	60.12	54.28	71.63	61.73
47	58.72	72.77	64.99	57.65	70.68	63.50	58.13	74.19	65.19	57.26	72.72	64.06
48	52.64	67.54	59.17	56.59	70.01	62.59	51.74	68.91	59.10	56.24	71.81	63.07
49	52.46	65.99	58.45	53.06	68.31	59.72	51.80	67.29	58.54	52.71	70.27	60.23
50	58.87	73.33	65.31	58.09	71.32	64.03	58.52	75.15	65.8	57.56	73.23	64.46
51	49.74	72.72	59.07	49.74	72.72	59.07	49.18	74.52	59.25	49.18	74.52	59.25

Table C1.16: Effect of BUS and RUS on Development and Test data using Feature F16

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	61.21	48.5	54.12	69.98	39.36	50.38	61.41	49.59	54.87	69.97	39.75	50.70
3	68.03	59.51	63.49	59.25	61.35	60.28	67.72	60.43	63.87	59.63	63.21	61.37
4	58.87	60.18	59.52	56.51	57.98	57.24	58.73	61.28	59.98	55.19	58.43	56.76
5	57.63	65.55	61.34	59.55	58.59	59.07	58.07	67.74	62.53	59.66	59.60	59.63
6	64.31	71.92	67.90	58.89	67.55	62.92	63.91	73.75	68.48	59.17	69.74	64.02
7	60.01	71.54	65.27	56.67	66.15	61.04	59.89	73.44	65.98	56.40	67.92	61.63
8	58.62	71.30	64.34	52.05	68.89	59.3	58.49	73.47	65.13	51.77	70.04	59.53
9	62.77	74.32	68.06	57.91	71.38	63.94	62.36	75.81	68.43	57.78	73.76	64.80
10	59.37	72.95	65.46	49.90	68.91	57.88	59.05	74.70	65.96	49.71	70.96	58.46
11	58.27	73.87	65.15	8.56	20.85	12.14	57.74	75.20	65.32	8.54	20.83	12.11
12	61.46	75.12	67.61	56.36	72.32	63.35	61.05	77.10	68.14	56.37	74.67	64.24
13	61.47	75.54	67.78	58.48	73.62	65.18	60.77	76.94	67.91	57.03	74.61	64.65
14	53.59	69.66	60.58	46.95	72.80	57.08	53.64	71.86	61.43	48.12	75.48	58.77
15	57.94	73.59	64.83	55.59	68.87	61.52	57.83	75.98	65.67	55.45	70.50	62.08
16	61.31	76.55	68.09	58.08	73.98	65.07	60.74	78.08	68.33	57.66	75.89	65.53
17	58.88	75.57	66.19	56.04	71.25	62.74	58.40	77.10	66.46	55.77	73.00	63.23
18	47.55	71.19	57.02	42.84	71.64	53.62	47.43	73.25	57.58	42.46	72.88	53.66
19	61.16	76.01	67.78	53.74	72.09	61.58	60.69	77.95	68.25	53.69	74.37	62.36
20	54.95	74.76	63.34	51.42	72.10	60.03	54.74	76.15	63.69	51.16	74.06	60.52
21	60.23	74.02	66.42	55.23	74.02	63.26	60.00	75.81	66.98	54.87	76.43	63.88
22	58.74	76.86	66.59	50.23	71.77	59.10	58.20	78.65	66.90	50.18	73.85	59.76

Table C1.16 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
23	60.15	75.90	67.11	57.13	75.10	64.89	59.54	77.53	67.35	55.60	75.86	64.17
24	55.10	72.47	62.6	42.67	70.42	53.14	55.14	74.56	63.4	41.90	70.71	52.62
25	50.37	70.83	58.87	49.14	63.19	55.29	50.23	72.89	59.47	48.27	63.61	54.89
26	58.64	76.41	66.36	53.06	74.63	62.02	58.01	77.66	66.41	52.74	76.96	62.59
27	47.61	72.62	57.51	55.16	34.87	42.73	46.96	73.23	57.22	54.90	35.65	43.23
28	59.97	77.64	67.67	46.50	71.83	56.45	59.65	79.37	68.11	45.95	72.94	56.38
29	57.06	75.69	65.07	35.26	72.15	47.37	56.59	77.55	65.43	35.32	73.71	47.76
30	58.59	77.65	66.79	54.81	75.11	63.37	58.11	79.49	67.14	54.44	77.30	63.89
31	59.22	77.73	67.22	53.93	74.41	62.54	58.77	79.36	67.53	53.58	76.58	63.05
32	54.03	74.64	62.68	52.10	72.22	60.53	53.63	76.67	63.11	51.56	73.53	60.62
33	58.79	77.59	66.89	48.47	72.47	58.09	58.14	79.36	67.11	48.33	74.57	58.65
34	55.78	75.40	64.12	54.41	69.79	61.15	55.27	77.36	64.48	54.04	71.78	61.66
35	59.55	76.52	66.98	38.12	66.31	48.41	59.04	77.91	67.17	37.91	67.74	48.61
36	49.44	73.60	59.15	46.41	62.84	53.39	49.29	76.1	59.83	45.48	62.84	52.77
37	55.32	74.57	63.52	34.01	64.97	44.65	54.85	76.31	63.82	34.23	67.10	45.33
38	52.39	74.21	61.42	37.27	66.07	47.66	51.71	75.91	61.52	37.12	67.32	47.85
39	59.00	77.73	67.08	48.68	71.80	58.02	58.39	79.14	67.2	48.11	73.78	58.24
40	56.57	75.71	64.76	50.17	70.74	58.71	56.07	77.90	65.21	50.03	72.17	59.09
41	59.20	76.96	66.92	55.87	75.29	64.14	58.83	78.55	67.27	55.38	77.33	64.54
42	60.22	76.80	67.51	50.20	70.84	58.76	59.86	78.63	67.97	49.65	72.67	58.99
43	58.85	76.84	66.65	54.75	74.68	63.18	58.38	78.39	66.92	54.31	76.58	63.55
44	60.06	76.57	67.32	9.01	31.61	14.02	59.56	77.89	67.50	8.39	31.16	13.22
45	56.64	75.74	64.81	38.24	61.52	47.16	56.19	77.85	65.27	38.02	62.72	47.34
46	58.61	75.98	66.17	55.6	74.95	63.84	58.35	77.90	66.72	55.19	76.77	64.22
47	55.47	74.02	63.42	51.75	73.16	60.62	54.71	76.93	63.94	52.21	74.25	61.31
48	59.32	77.42	67.17	33.91	62.96	44.08	58.66	78.84	67.27	33.94	64.90	44.57
49	53.00	74.65	61.99	7.07	36.66	11.85	52.61	76.69	62.41	6.95	37.01	11.70
50	58.56	76.85	66.47	44.70	70.96	54.85	58.25	78.35	66.82	44.17	72.12	54.79
51	56.66	79.57	66.19	56.66	79.57	66.19	56.34	81.34	66.57	56.34	81.34	66.57

Table C1.17: Effect of BUS and RUS on Development and Test data using Feature F17

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	73.09	41.83	53.21	58.24	45.24	50.92	61.18	48.08	53.84	71.16	40.76	51.83
3	69.90	57.33	62.99	60.65	53.69	56.96	69.74	58.33	63.53	60.68	55.25	57.84
4	64.28	57.95	60.95	60.62	54.76	57.54	64.02	58.84	61.32	60.57	56.45	58.44
5	57.41	61.58	59.42	51.83	45.03	48.19	57.27	63.13	60.06	51.68	45.53	48.41
6	64.44	70.16	67.18	56.86	63.86	60.16	64.55	72.41	68.25	57.24	66.28	61.43
7	62.86	68.77	65.68	51.09	55.82	53.35	62.54	70.28	66.18	50.66	56.60	53.47
8	61.54	70.51	65.72	57.51	68.54	62.54	61.17	72.22	66.24	57.60	71.02	63.61
9	59.91	72.29	65.52	57.69	67.89	62.38	59.16	73.89	65.71	57.54	70.33	63.3
10	62.60	73.17	67.47	59.07	72.07	64.93	62.56	75.50	68.42	59.34	74.79	66.18
11	57.27	70.64	63.26	55.16	68.10	60.95	57.06	72.80	63.98	54.9	69.84	61.48
12	61.05	71.11	65.70	48.64	67.87	56.67	60.65	72.62	66.10	48.08	69.35	56.79
13	58.86	73.57	65.4	40.79	48.70	44.40	58.51	75.53	65.94	40.62	49.67	44.69
14	58.30	72.70	64.71	53.57	73.30	61.90	58.16	75.27	65.62	54.57	75.00	63.17
15	59.00	73.71	65.54	53.31	71.96	61.25	58.20	75.16	65.6	53.21	74.28	62.00
16	62.13	75.78	68.28	59.02	73.26	65.37	61.64	77.65	68.72	58.91	76.01	66.38
17	58.22	73.47	64.96	53.54	73.61	61.99	58.00	76.09	65.82	54.69	75.63	63.48
18	60.69	75.76	67.39	56.56	73.92	64.09	60.52	77.96	68.14	56.3	76.44	64.84
19	55.57	72.56	62.94	52.93	69.90	60.24	55.30	74.54	63.49	52.60	72.32	60.9
20	59.03	73.69	65.55	56.56	71.33	63.09	58.85	75.60	66.18	56.50	73.88	64.03
21	59.26	76.09	66.63	57.12	70.65	63.17	58.6	78.02	66.93	56.44	72.82	63.59
22	61.14	75.91	67.73	57.52	74.29	64.84	60.76	77.95	68.29	57.36	76.88	65.70
23	58.97	75.63	66.27	55.70	72.93	63.16	58.55	77.52	66.71	55.67	75.42	64.06
24	59.16	76.30	66.65	56.50	73.73	63.98	59.06	78.37	67.36	56.21	76.11	64.66
25	51.67	69.74	59.36	47.37	70.29	56.6	51.39	71.70	59.87	46.91	72.28	56.89
26	44.47	65.57	53.00	29.29	47.80	36.32	44.15	67.44	53.36	28.56	47.93	35.79
27	57.49	76.32	65.58	54.90	73.68	62.92	56.99	78.06	65.88	54.76	76.18	63.72
28	60.97	75.58	67.49	55.91	74.55	63.90	60.72	77.86	68.23	55.36	76.99	64.41
29	58.31	74.94	65.59	55.22	73.27	62.98	58.15	77.15	66.32	56.09	75.29	64.29
30	54.01	73.93	62.42	51.24	71.14	59.57	53.80	76.55	63.19	51.97	74.10	61.09
31	57.49	73.94	64.69	56.24	69.56	62.19	57.62	76.75	65.82	56.08	71.64	62.91
32	60.54	75.77	67.3	47.98	68.31	56.37	60.21	77.65	67.83	47.69	70.19	56.79
33	60.11	76.62	67.37	56.34	73.51	63.79	60.07	79.00	68.25	56.22	75.92	64.60
34	52.24	73.47	61.06	33.99	61.66	43.82	51.89	75.26	61.43	34.13	63.20	44.32
35	55.79	73.16	63.31	35.99	68.15	47.10	55.93	75.75	64.35	35.27	68.89	46.65

Table C1.17 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
Rs	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
36	52.60	73.46	61.30	52.53	65.69	58.38	52.15	75.79	61.79	52.28	66.83	58.67
37	60.14	76.37	67.29	53.29	73.06	61.63	59.70	78.08	67.66	52.92	75.51	62.23
38	60.56	76.49	67.60	54.06	73.29	62.22	60.29	78.55	68.22	53.93	76.02	63.10
39	57.67	74.79	65.12	48.6	71.04	57.72	57.54	77.23	65.95	48.06	72.12	57.68
40	59.36	75.55	66.48	54.69	72.63	62.4	58.89	77.53	66.94	54.27	75.04	62.99
41	59.22	77.08	66.98	45.81	72.47	56.14	59.01	79.12	67.60	45.39	74.47	56.4
42	60.15	76.48	67.34	55.62	73.19	63.21	59.81	77.98	67.70	55.78	76.21	64.41
43	55.85	73.72	63.55	45.26	67.27	54.11	55.82	76.19	64.43	44.59	68.41	53.99
44	59.88	77.10	67.41	54.03	72.39	61.88	59.36	78.77	67.70	53.95	74.64	62.63
45	60.36	76.9	67.63	48.37	67.87	56.48	59.71	78.74	67.92	48.71	69.90	57.41
46	55.78	74.98	63.97	37.64	67.66	48.37	55.19	76.94	64.27	37.38	69.24	48.55
47	59.02	76.98	66.81	55.61	74.24	63.59	58.30	78.56	66.93	55.06	76.20	63.93
48	57.19	74.24	64.61	53.60	74.83	62.46	57.02	76.91	65.49	53.19	76.75	62.83
49	60.30	76.91	67.60	36.78	67.23	47.55	60.05	79.03	68.24	36.74	69.13	47.98
50	59.94	76.04	67.04	31.87	66.10	43.01	59.54	77.84	67.47	31.58	66.97	42.92
51	51.30	79.02	62.21	51.30	79.02	62.21	50.33	80.53	61.95	50.33	80.53	61.95

Table C1.18: Effect of BUS and RUS on Development and Test data using Feature F18

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	76.44	41.17	53.52	66.37	41.86	51.34	45.88	66.98	54.46	41.75	69.56	52.18
3	71.40	49.25	58.29	64.24	50.05	56.26	41.80	73.28	53.23	37.16	72.10	49.04
4	64.14	55.00	59.22	65.17	49.23	56.09	39.56	75.18	51.84	35.16	74.59	47.79
5	66.1	57.31	61.39	56.97	60.58	58.72	38.89	77.36	51.76	36.78	75.24	49.41
6	63.92	61.58	62.73	46.45	45.86	46.15	36.87	77.95	50.06	26.80	68.32	38.50
7	62.45	64.17	63.30	53.43	56.04	54.70	37.70	77.49	50.72	30.94	74.04	43.64
8	60.16	65.56	62.74	56.06	65.77	60.53	34.06	78.19	47.45	32.62	74.67	45.40
9	59.64	65.27	62.33	55.38	64.84	59.74	34.35	78.66	47.82	31.22	76.98	44.42
10	59.91	68.72	64.01	53.96	63.49	58.34	36.20	79.60	49.77	31.46	74.83	44.3
11	55.64	68.71	61.49	50.42	62.26	55.72	34.73	77.85	48.03	30.45	77.92	43.79
12	57.61	69.87	63.15	54.93	67.38	60.52	34.55	79.16	48.10	31.87	77.41	45.15
13	58.11	71.44	64.09	55.21	70.15	61.79	35.22	79.92	48.89	32.40	76.64	45.55

Table C1.2 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	76.44	41.17	53.52	66.37	41.86	51.34	45.88	66.98	54.46	41.75	69.56	52.18
3	71.4	49.25	58.29	64.24	50.05	56.26	41.80	73.28	53.23	37.16	72.10	49.04
4	64.14	55.00	59.22	65.17	49.23	56.09	39.56	75.18	51.84	35.16	74.59	47.79
5	66.1	57.31	61.39	56.97	60.58	58.72	38.89	77.36	51.76	36.78	75.24	49.41
6	63.92	61.58	62.73	46.45	45.86	46.15	36.87	77.95	50.06	26.80	68.32	38.5
7	62.45	64.17	63.30	53.43	56.04	54.7	37.70	77.49	50.72	30.94	74.04	43.64
8	60.16	65.56	62.74	56.06	65.77	60.53	34.06	78.19	47.45	32.62	74.67	45.4
9	59.64	65.27	62.33	55.38	64.84	59.74	34.35	78.66	47.82	31.22	76.98	44.42
10	59.91	68.72	64.01	53.96	63.49	58.34	36.20	79.60	49.77	31.46	74.83	44.3
11	55.64	68.71	61.49	50.42	62.26	55.72	34.73	77.85	48.03	30.45	77.92	43.79
12	57.61	69.87	63.15	54.93	67.38	60.52	34.55	79.16	48.10	31.87	77.41	45.15
13	58.11	71.44	64.09	55.21	70.15	61.79	35.22	79.92	48.89	32.40	76.64	45.55
14	55.73	69.34	61.79	52.66	67.37	59.11	32.8	78.98	46.35	30.19	77.93	43.52
15	58.07	71.66	64.15	54.26	69.11	60.79	35.93	79.83	49.56	31.81	77.78	45.15
16	57.66	71.95	64.02	53.85	68.44	60.27	35.21	80.01	48.90	33.23	77.18	46.46
17	57.68	72.02	64.06	30.64	50.14	38.04	35.33	80.22	49.06	18.91	70.81	29.85
18	56.40	71.08	62.89	53.43	67.30	59.57	34.04	78.82	47.55	31.29	76.48	44.42
19	52.67	67.39	59.13	38.02	41.54	39.7	27.64	75.13	40.41	21.44	63.20	32.02
20	56.82	72.89	63.86	53.86	70.96	61.24	33.99	79.79	47.67	31.67	78.18	45.08
21	56.92	73.44	64.13	45.78	66.20	54.13	34.72	80.44	48.50	26.61	77.59	39.63
22	55.51	70.43	62.09	50.48	69.20	58.38	33.39	78.36	46.83	29.13	76.82	42.24
23	58.08	65.14	61.41	51.38	69.43	59.06	34.34	78.01	47.69	31.69	78.37	45.13
24	53.07	68.04	59.63	50.66	64.04	56.57	31.19	77.63	44.50	28.08	76.40	41.07
25	56.83	73.65	64.16	39.11	58.95	47.02	36.44	80.3	50.13	23.78	75.70	36.19
26	55.69	73.51	63.37	52.83	71.12	60.63	35.03	79.76	48.68	30.14	78.21	43.51
27	54.81	73.48	62.79	49.07	70.47	57.85	34.25	79.95	47.96	30.63	77.08	43.84
28	54.08	66.61	59.69	35.96	62.46	45.64	32.44	78.94	45.98	23.31	76.95	35.78
29	52.54	71.41	60.54	50.48	66.08	57.24	31.33	78.5	44.79	29.54	77.00	42.70
30	56.16	71.28	62.82	52.89	69.72	60.15	32.78	79.51	46.42	28.93	77.79	42.18
31	55.36	73.91	63.30	37.73	52.48	43.9	35.92	80.73	49.72	21.77	72.04	33.44
32	52.77	72.08	60.93	49.18	67.83	57.02	29.95	77.00	43.13	28.04	76.95	41.10
33	55.83	73.11	63.31	51.50	69.21	59.06	34.61	79.59	48.24	27.94	77.82	41.12
34	53.78	71.14	61.25	28.79	50.30	36.62	32.86	79.88	46.56	18.89	68.28	29.59
35	54.39	73.15	62.39	51.54	71.62	59.94	31.79	79.76	45.46	29.40	78.29	42.75
36	53.71	72.24	61.61	50.52	67.28	57.71	30.46	79.20	44.00	28.02	77.48	41.16

Table C1.18 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
37	45.05	69.39	54.63	33.97	50.97	40.77	29.16	79.97	42.74	20.19	67.31	31.06
38	54.63	70.76	61.66	47.82	58.37	52.57	31.85	79.86	45.54	27.63	75.37	40.44
39	52.82	71.66	60.81	48.75	68.95	57.12	31.97	80.38	45.75	28.81	77.36	41.98
40	55.19	73.79	63.15	41.32	52.98	46.43	33.13	80.29	46.91	25.37	73.57	37.73
41	55.37	73.03	62.99	49.41	68.40	57.37	31.94	80.06	45.66	29.81	78.72	43.24
42	55.12	73.89	63.14	52.67	72.07	60.86	32.52	79.93	46.23	30.26	79.07	43.77
43	54.73	72.19	62.26	45.38	62.37	52.54	32.86	79.70	46.53	26.74	77.44	39.75
44	49.30	68.20	57.23	39.10	60.82	47.60	27.03	78.18	40.17	24.24	75.38	36.68
45	55.04	74.48	63.3	52.83	71.97	60.93	32.35	80.57	46.16	29.24	77.45	42.45
46	53.99	73.07	62.1	50.14	70.80	58.71	30.19	79.10	43.70	27.23	76.96	40.23
47	54.16	73.44	62.34	52.07	70.35	59.85	31.33	79.68	44.98	29.12	77.15	42.28
48	54.02	71.03	61.37	48.13	66.60	55.88	29.88	78.46	43.28	27.63	77.43	40.73
49	55.79	73.90	63.58	52.09	71.54	60.29	33.25	80.64	47.09	31.6	77.91	44.96
50	53.77	74.16	62.34	51.85	70.1	59.61	31.45	80.04	45.16	28.96	77.67	42.19
51	50.19	76.19	60.52	50.19	76.19	60.52	30.48	79.96	44.14	30.48	79.96	44.14

Table C1.19 Effect of BUS and RUS on Development and Test data using Feature F19

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
2	75.45	26.24	38.94	64.83	40.21	49.63	75.57	26.28	39.00	64.74	40.93	50.14
3	73.00	32.71	45.18	60.56	47.09	52.90	72.53	32.91	45.28	60.30	47.95	53.33
4	68.16	49.86	57.59	48.53	42.99	44.90	67.85	50.89	58.16	48.09	43.93	45.19
5	63.80	55.34	59.27	61.42	60.55	60.98	63.18	56.31	59.55	61.07	62.24	61.64
6	64.57	61.58	63.04	60.99	62.39	61.68	64.35	63.65	64.00	60.85	64.28	62.51
7	58.63	58.16	58.39	60.05	64.67	62.27	58.13	59.48	58.80	60.00	66.93	63.27
8	26.58	23.79	25.11	57.65	64.73	60.95	25.92	23.63	24.72	57.15	66.77	61.56
9	62.45	68.30	65.24	59.93	66.87	63.21	62.15	70.66	66.13	59.91	69.16	64.20
10	61.23	66.34	63.68	55.87	62.87	59.12	60.62	68.32	64.24	55.58	64.87	59.82
11	61.63	69.95	65.53	58.86	66.86	62.60	61.44	72.13	66.36	58.91	69.32	63.69
12	61.19	69.49	65.08	55.62	65.72	60.24	60.71	71.84	65.81	55.34	67.68	60.88
13	60.65	70.73	65.30	58.31	68.21	62.87	60.20	72.90	65.94	58.17	70.55	63.76
14	59.32	70.62	64.48	55.12	64.60	59.45	58.65	72.49	64.84	54.84	66.70	60.16

Table C1.19 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
15	61.25	69.81	65.25	48.35	57.03	52.31	60.73	71.86	65.83	48.02	58.70	52.80
16	56.87	59.4	58.11	59.09	70.22	64.18	56.27	60.68	58.39	58.93	72.52	65.03
17	42.32	62.99	50.63	57.04	69.02	62.46	42.02	64.70	50.95	56.55	71.07	62.98
18	60.75	72.30	66.02	56.58	69.18	62.24	60.38	74.46	66.68	56.14	71.19	62.76
19	61.41	71.37	66.02	51.68	63.77	56.99	61.17	73.58	66.8	51.27	65.76	57.52
20	55.9	70.25	62.26	53.46	67.76	59.72	55.42	72.13	62.68	52.88	69.57	60.04
21	60.47	71.77	65.64	54.58	68.09	60.54	60.10	74.08	66.36	54.34	70.13	61.19
22	59.82	72.59	65.59	58.30	70.72	63.91	59.56	74.97	66.38	57.85	72.88	64.50
23	34.09	51.85	41.13	57.04	70.09	62.89	33.96	53.43	41.53	56.52	72.10	63.36
24	60.9	71.52	65.78	58.14	70.50	63.73	60.43	73.62	66.38	57.67	72.58	64.27
25	55.65	68.89	61.57	52.64	68.09	58.90	55.04	70.79	61.93	52.37	70.21	59.51
26	29.84	38.22	33.51	52.97	67.91	59.40	30.07	39.84	34.27	52.52	69.86	59.86
27	60.5	73.03	66.18	57.34	70.29	63.15	59.94	74.98	66.62	56.83	72.35	63.65
28	60.72	72.79	66.21	45.36	61.27	51.85	60.73	75.11	67.16	44.73	62.86	51.99
29	57.29	69.93	62.98	50.65	67.00	57.64	56.34	71.57	63.05	49.92	68.75	57.79
30	59.42	72.87	65.46	54.94	67.65	60.63	58.85	74.78	65.87	54.56	69.77	61.24
31	58.85	72.77	65.07	55.32	70.07	61.79	58.71	75.17	65.93	54.69	71.89	62.09
32	60.58	72.93	66.18	57.78	69.52	63.11	60.30	75.07	66.88	57.31	71.63	63.68
33	59.47	72.37	65.29	57.97	71.20	63.90	58.96	74.49	65.82	57.54	73.40	64.51
34	59.67	73.04	65.68	55.19	69.27	61.38	59.25	75.43	66.37	54.65	71.31	61.82
35	60.61	72.6	66.07	55.05	68.21	60.92	60.41	74.79	66.84	54.44	70.19	61.32
36	60.74	72.66	66.17	56.49	69.08	62.15	60.31	74.68	66.73	56.05	71.13	62.69
37	60.51	72.39	65.92	57.22	70.99	63.35	60.24	74.75	66.72	56.83	73.22	63.99
38	59.74	73.02	65.72	57.03	70.31	62.98	59.28	75.15	66.28	56.45	72.32	63.40
39	59.88	72.88	65.74	57.83	71.01	63.75	59.51	74.99	66.36	57.30	73.05	64.22
40	59.78	71.51	65.12	54.06	68.34	60.35	59.46	73.35	65.68	53.47	70.03	60.62
41	55.23	71.91	62.48	57.62	70.97	63.59	54.55	74.00	62.8	57.02	72.98	64.02
42	60.78	72.66	66.19	55.13	67.67	60.75	60.54	74.67	66.87	54.41	69.50	61.03
43	49.11	69.73	57.63	53.29	69.12	60.11	48.54	71.25	57.74	52.85	71.08	60.56
44	58.51	71.04	64.17	51.03	63.69	56.62	58.06	73.20	64.76	50.53	65.59	57.04
45	58.89	73.28	65.3	56.34	69.71	62.31	58.15	75.29	65.62	55.8	71.85	62.81
46	58.81	72.93	65.11	58.17	71.35	64.09	58.29	75.00	65.6	57.77	73.59	64.72
47	57.02	69.94	62.82	57.4	69.56	62.89	56.23	71.60	62.99	56.93	71.57	63.41
48	60.02	72.84	65.81	56.97	69.97	62.80	59.63	74.96	66.42	56.44	72.05	63.29
49	60.43	71.63	65.56	58.37	71.37	64.22	59.88	73.51	66.00	57.94	73.51	64.80

Table C1.19 (Continued.)

	Develop						Test					
	BUS			RUS			BUS			RUS		
R_s	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score	Recall	Precision	F score
50	59.15	73.25	65.45	56.29	69.72	62.29	58.72	75.51	66.06	55.84	72.07	62.92
51	56.83	73.12	63.96	56.83	73.12	63.96	55.75	76.08	64.35	55.75	76.08	64.35

C.2 Results of Stop Word Filtering using different feature sets ($F_{1\text{-}19}$) on Development and Test data

Table C2.1: Stop word Filtering Results

	Develop			Test		
F	Recall	Precision	F score	Recall	Precision	F score
1	72.95	63.56	67.94	73.60	62.35	67.51
2	73.42	60.82	66.53	76.34	59.98	67.18
3	73.31	65.98	69.45	71.38	67.85	69.57
4	70.63	41.66	52.41	66.29	41.23	50.84
5	67.33	48.54	56.41	69.60	42.03	52.41
6	71.49	56.28	62.98	72.73	55.73	63.10
7	61.86	62.24	62.05	69.08	56.91	62.41
8	71.34	56.84	63.27	60.35	59.32	59.83
9	68.73	60.87	64.56	72.64	56.10	63.31
10	71.69	57.59	63.87	72.44	53.78	61.73
11	74.69	61.26	67.31	74.57	56.61	64.36
12	71.87	52.85	60.91	72.44	59.79	61.73
13	66.38	39.40	49.45	67.49	38.75	49.33
14	72.35	55.92	63.08	70.36	52.04	59.83
15	58.01	11.56	19.28	58.87	11.30	18.96
16	70.68	49.55	58.25	70.79	44.45	54.61
17	71.56	40.51	51.73	71.36	37.11	48.83
18	69.84	40.61	51.36	61.73	26.76	37.34
19	72.06	49.30	58.55	73.94	49.05	58.97

Appendix D: List of Stop Word Used

Table D.1: Stop words used

a	ask	concerning	example	herein	latterly	no	please	she	thereby	useful	wish
able	asking	consequently	except	hereupon	least	nobody	plus	should	therefore	uses	with
about	associated	consider	far	hers	less	non	possible	since	therein	using	within
above	at	considering	few	herself	lest	none	presumably	six	theres	usually	without
according	available	contain	fifth	hi	let	noone	probably	so	thereupon	uucp	wonder
accordingly	away	containing	first	him	like	nor	provides	some	these	value	would
across	awfully	contains	five	himself	liked	normally	que	somebody	they	various	would
actually	be	corresponding	followed	his	likely	not	quite	somehow	think	very	yes
after	became	could	following	hither	little	nothing	qv	someone	third	via	yet
afterwards	because	course	follows	hopefully	look	novel	rather	something	this	viz	you
again	become	currently	for	how	looking	now	rd	sometime	thorough	vs	your
against	becomes	definitely	former	howbeit	looks	nowhere	re	sometimes	thoroughly	want	yours
all	becoming	described	formerly	however	ltd	obviously	really	somewhat	those	wants	yourself
allow	been	despite	forth	ie	mainly	of	reasonably	somewhere	though	was	yourselves
allows	before	did	four	if	many	off	regarding	soon	three	way	zero
almost	beforehand	different	from	ignored	may	often	regardless	sorry	through	we	
alone	behind	do	further	immediate	maybe	oh	regards	specified	throughout	welcome	
along	being	does	furthermore	in	me	ok	relatively	specify	thru	well	
already	believe	doing	get	inasmuch	mean	okay	respectively	specifying	thus	went	
also	below	done	gets	inc	meanwhile	old	right	still	to	were	
although	beside	down	getting	indeed	merely	on	said	sub	together	what	
always	besides	downwards	given	indicate	might	once	same	such	too	whatever	
am	best	during	gives	indicated	more	one	saw	sup	took	when	
among	better	each	go	indicates	moreover	ones	say	sure	toward	whence	
amongst	between	edu	goes	inner	most	only	saying	take	towards	whenever	
an	beyond	eg	going	insofar	mostly	onto	says	taken	tried	where	
and	both	eight	gone	instead	much	or	second	tell	tries	whereafter	
another	brief	either	got	into	must	other	secondly	tends	truly	whereas	
any	but	else	gotten	inward	my	others	see	the	try	whereby	
anybody	by	elsewhere	greetings	is	myself	otherwise	seeing	than	trying	wherein	
anyhow	came	enough	had	it	name	ought	seem	thank	twice	whereupon	
anyone	can	entirely	happens	its	namely	our	seemed	thanks	two	wherever	
anything	cannot	especially	hardly	itself	nd	ours	seeming	thanx	un	whether	
aside	comes	exactly	hereby	latter	nine	placed	shall	thereafter	used	willing	