BIOMEDICAL NAMED ENTITY RECOGNITION

A Thesis Submitted for the Degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

By

Hamada Ali Mohamed Ali Nayel

Under the Supervision of

Dr. H. L. Shashirekha

Department of Post-Graduate Studies and Research in Computer Science

Mangalore University, Mangalagangothri-574199

Mangalore, Karnataka, India

July - 2018

MANGALORE UNIVERSITY

Department of Post-Graduate Studies and Research in Computer Science
Mangalore University, Mangalagangothri-574199
Mangalore, Karnataka, India

CERTIFICATE

This is to certify that the thesis entitled "**BIOMEDICAL NAMED ENTITY RECOGNITION**" submitted by **Mr. Hamada Ali Mohamed Ali Nayel** for the award of the degree of the Doctor of Philosophy in Computer Science is based on the results of the study carried out by him under my supervision. The thesis or part thereof has not been previously submitted for any other degree or diploma.

Dr. H.L Shashirekha
Research Supervisor
Professor, Department of Computer Science
Mangalore University, Mangalore

MANGALORE UNIVERSITY

Department of Post-Graduate Studies and Research in Computer Science
Mangalore University, Mangalagangothri-574199
Mangalore, Karnataka, India

DECLARATION

I hereby, declare that this thesis entitled "**BIOMEDICAL NAMED ENTITY RECOGNITION**", embodies the results of bonafide research work carried out by me, under the supervision of Dr. H. L. Shashirekha, Professor, Department of Post-Graduate Studies and Research in Computer Science, Mangalore University. I also certify that the work in the thesis is my own and original and has not been submitted to this University or any other University wholly or in part for the award of a degree.

Hamada Ali Mohamed Ali Nayel
Research Scholar
Department of Computer Science
Mangalore University, Mangalore

ACKNOWLEDGEMENT

I take this opportunity with much pleasure to thank all the people who have helped me through the course of my journey towards producing this thesis. Foremost, I sincerely thank my supervisor, **Prof H. L. Shashirekha**, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. Her wide knowledge and logical way of thinking have been of great value for me. Apart from the subject of my research, I learnt a lot from her, which I am sure, will be useful in different stages of my life.

I would like to express my sincere gratitude to the faculty members, **Prof. Manjaiah D.H, Prof. B.H.Shekar, Prof. Doreswamy** and **Mr. Prakash** for their support and encouragements and critical comments during the research work. I also thank, the Guest Faculties and non-teaching staff members of the Department of Computer Science. I would like to thank research scholars in the Department of Computer Science for their support and kind help.

I would like to express my deepest gratitude to **Prof. K. Byrappa**, Former Vice Chancellor, **Prof. B.S. Nagendra Prakash**, Registrar and **Prof. A. M. Khan**, Registrar (Evaluation) of Mangalore University, and also all other administrative staff of Mangalore University for helping me to work towards my Doctoral degree. I also would thank **Prof. Ravishankar Rao**, Director of International Students Centre, Mangalore University, for his support and help during my research work.

I would also like to thank **Indian Council for Cultural Relations (ICCR), Government of India, and Government of Arab Republic of Egypt** for their support and giving me the opportunity to do my doctoral research work in India. It will be endless to record and express to each and every one who has directly or indirectly co-operated for my success in all endeavors.

I wish to extend my warmest thanks to **Prof. Yuji Matsumoto**, Head of Computational Linguistics Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan, for his guidance during the research visit to his laboratory.

Also, I'd like to thank **Hiroyuki Shindo**, Assistant Professor, NAIST, for the collaborative research work carried out during my visit to NAIST.

Lastly, I'd like to thank my parents for supporting me spiritually throughout my life. Above all I'd like to thank my loving, supportive, encouraging and patient wife Mrs Amira, son Ali and daughter Salma. Without their encouragement and understanding it would have been impossible for me to finish this work. I dedicate this thesis to them.

Finally, I would like to thank everybody who helped me directly or indirectly for the successful realization of my thesis as well as express my apology for not mentioning them personally.

Hamada Ali Mohamed Ali Nayel

Abstract

Named Entity Recognition (NER) is a crucial Natural Language Processing (NLP) task which extracts Named Entities (NE) from the text. Names of persons, places, date and time are examples of NEs in general domain texts, while names of genes, proteins and diseases are examples of NEs in biomedical domain termed as BioNE. NER in Biomedical domain (BioNER) is an important preprocessing task for many further tasks such as relation extraction between entities, knowledge discovery and hypothesis generation. The tremendous growth of publications in biomedical research area makes it vital to apply BioNER as it is tough to extract NEs manually. Furthermore, BioNEs pose several challenges related to ambiguous names, synonyms, variations, multi-word NEs and nested NEs.

Different approaches have been used for BioNER, such as dictionary approaches, rule-based approaches, Machine Learning (ML) approaches and hybrid approaches. Of late ML approaches specially Artificial Neural Network based models are popularly being used for BioNER. Annotating the dataset for training the models to recognize and classify NEs is a crucial task in BioNER. There are many methods used for annotating the datasets such as XML format, BioNEs offset and Segment Representation (SR). SR is an efficient way of annotating BioNEs within a sentence in order to differentiate them from non-BioNEs. Different SR schemes such as IO, IOE2, IOB2, IOBE and IOBES are used to annotate the dataset to develop efficient BioNER systems.

Support Vector Machines (SVM) and Conditional Random Fields (CRF) have been used to train different BioNER models with different SR schemes. The Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004) shared task dataset, National Center for Biotechnology Information (NCBI) dataset, BioCreative II Gene Mention (BC2GM) recognition shared task dataset, BioCreative V Chemical-Disease Relation (BC5CDR) task dataset and i2b2/VA 2010 shared task[1] dataset are used to assess the performance of BioNER systems.

[1] https://www.i2b2.org/NLP/Relations/

Ensemble approach which combines the output of base classifiers to get better performance from a pool of classifiers than an individual classifier has achieved meaningful and encouraging results. The generalization capability of ensemble classifier which is usually much better than that of base classifiers is the strength of ensemble classifier. It has been applied for different Natural Language Processing (NLP) tasks such as BioNER, word segmentation, word sense disambiguation and Part-of-Speech (PoS) tagging. An ensemble based BioNER system using two different classification algorithms CRF and SVM and with IO, IOB2, IOE2, IOBE and IOBES SR schemes is proposed. To study the impact of ensemble approach on other NLP task an ensemble-based system is developed for Native Language Identification (NLI) - an important NLP task that has got the attention recently by research community.

In recent years, Deep Learning (DL) models are becoming important due to their demonstrated success at overcoming complex learning problems. DL models have been applied effectively for different NLP tasks such as PoS tagging and Machine Translation. A DL model for Disease Named Entity Recognition (Disease-NER) using dictionary information is proposed and evaluated on NCBI disease corpus and BC5CDR dataset. Pre-trained word embeddings trained over general domain texts as well as biomedical texts have been used to represent input to the proposed model. This study also compares two different SR schemes, namely IOB2 and IOBES for Disease-NER. The results illustrate that using dictionary information, pre-trained word embeddings, character embeddings and CRF with global score improves the performance of BioNER system.

An extension of IOBES SR scheme is proposed to improve the representation of multi-word entities and hence the performance of BioNER. A Bidirectional Long Short-Term Memory (BiLSTM) network is used to design a baseline system for BioNER and the new SR model is evaluated on i2b2/VA 2010 challenge dataset and JNLPBA 2004 shared task dataset. Results obtained illustrate that the proposed SR model outperforms IOB2 and IOBES schemes for multi-word entities with length greater than two. Further, the outputs of different SR models combined using majority voting ensemble method illustrate that ensemble method outperforms the performance of baseline models.

Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
BioNE	Biomedical Named Entity
BioNER	Biomedical Named Entity Recognition
BioNLP	Biomedical Natural Language Processing
CBOW	Continuous Bag of Words
CRF	Conditional Random Fields
Disease-NER	Disease Named Entity Recognition
DL	Deep Learning
DT	Decision Tree
FIRE	Forum of Information Retrieval Evaluation
FN	False Negative
FP	False Positive
HMM	Hidden Markov Model
ICLE	International Corpus of Learner English
IDF	Inverse Document Frequency
INLI	Indian Native Language Identification
IoT	Internet of Things
IR	Information Retrieval
LM	Language Modeling
LSTM	Long Short-Term Memory
MEDIC	MErged DIsease voCabulary
MEDLINE	Medical Literature Analysis and Retrieval System Online
ML	Machine Learning
MT	Machine Translation

MUC	Message Understanding Conference
NCBI	National Center for Biotechnology Information
NE	Named Entity
NER	Named Entity Recognition
NLI	Native Language Identification
NLM	National Library of Science
NLP	Natural Language Processing
NP-chunking	Noun Phrase Chunking
PMC	PubMed Central
PoS	Part-of-Speech
QA	Question Answering
RBF	Radial Basis Function
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SLA	Second Language Acquisition
SR	Segment Representation
SVM	Support Vector Machine
TF/IDF	Term Frequency / Inverse Document frequency
TM	Text Mining
TP	True Positive
WSD	Word Sense Disambiguation

Chapter 1

Introduction

1.1 Preamble

Huge amount of data is generated on daily basis from various sources such as newspapers, internet blogs, social media, literature, digital libraries, satellite data, videos and publications. It has been estimated that nearly 80% of the worlds digital data are represented in an unstructured textual form such as scientific papers, academic articles, journals and patents [1]. These data which involve a considerable amount of information has attracted the attention of many researchers to explore or analyse for various applications such as search engines, semantic web, e-commerce and Internet of Things (IoT). The reduction in the size and price of storage devices and the technology to transfer greater volumes of data at a much faster rate also adds to the accumulation of large volumes of data. However, analysing this large ever-growing amount of data manually is not only time consuming and labour intensive but is also error-prone. This creates a demand for the techniques and tools which aim at automatically analysing the data at a much faster rate. A group of techniques which deal with automatically analysing data for various applications is termed as Text Mining (TM).

TM has emerged to discover useful and meaningful information such as patterns, associations and relationships among entities in any unstructured natural language documents [2]. Typical TM tasks include text categorization, clustering, information extraction, document summarization and entity relation identification. TM tasks help users to get better insight into the unstructured text documents. For example, document clustering provides an efficient way of organizing web search results, and document summarization can gener-

ate information snippets to help users in exploring the web. Further, TM tasks are applied to different categories of documents such as news, research publications, discharge summaries and clinical reports.

Texts belonging to various domains such as banks, insurance and financial markets, legal documents and health care can be analyzed using TM techniques. TM has specific applications in life science and health care industries where large amount of textual data regarding patients records, diseases, medicines, symptoms and treatments of diseases and many more are generated frequently. It is difficult and challenging to extract useful information from biomedical texts due to ever growing repositories and usage of complex and technical vocabularies [3, 4]. To overcome these problems, researchers have developed several automated tools for TM in biomedical field such as BioIE[1], Knowledge EXtraction (KEX)[2], GoPubMed[3], A Biomedical Named Entity Recognizer (ABNER)[4] and LingPipe[5] for various applications. These tools provide an opportunity to extract valuable information, their association and inferring relationship among various diseases, species, and genes from biomedical texts. Use of appropriate TM tools in medical field help to evaluate the effectiveness of medical treatments by comparing different diseases, symptoms and their course of treatments [5]. TM tools are also used in biomarker discovery, pharmaceutical industry, clinical trade analysis, preclinical safe toxicity studies, patent competitive intelligence and landscaping, mapping diseases genes and exploring the targeted identifications [6].

Natural Language Processing (NLP) spans over the techniques of multiple disciplines such as Artificial Intelligence, Computer Science, Statistics and Linguistics applied for the discovery of interactions between computers and natural languages. It involves the understanding of natural languages and thus enable computers to derive meaning from natural languages input as well as to generate natural languages and to translate between different languages. As NLP aims to extract a comprehensive meaningful representation from free text, NLP techniques can be used roughly in TM.

The tasks of NLP include morphological analysis, syntactic and semantic analysis, sen-

[1] http://130.88.97.239/bioie/
[2] http://hgc.jp/service/tooldoc/KeX/intro.html
[3] http://www.gopubmed.org/web/gopubmed/
[4] http://pages.cs.wisc.edu/ bsettles/abner/
[5] http://alias-i.com/lingpipe/

timent analysis, information retrieval, Question Answering (QA), Machine Translation (MT), document summarization or speech recognition. Understanding morphological structure and syntactic rules of the text helps analysing a text semantically. Most of these NLP tasks can be divided into related sub-problems such as Part-of-Speech (PoS) tagging, Word Sense Disambiguation (WSD) and Named Entity Recognition (NER).

With the copious biomedical literature the need for a means to automatically distill information of interest from unstructured data arises. Further, it has become extremely difficult for biologists to keep up with the relevant publications in their own discipline due to the massive volume of biomedical text [7]. In response to these needs, many researchers in the field of NLP have taken interest in developing methods specifically for the biomedical domain, which is now popularly known as Biomedical NLP (BioNLP). While the conventional NLP techniques address the needs of text processing in generic domain, BioNLP techniques are customized to address the needs of processing biomedical texts.

1.2 Problem Description

A Named Entity (NE) is the name of a person, place, date and time in general newswire domain. NER is an important preprocessing technique for many applications such as Information Retrieval (IR) and QA [8, 9]. For example, in the QA competition in TREC-8 [6], 80% of the evaluation questions asked were for a NE [10]. NER is used to improve semantic search for enabling the Semantic Web [11]. In MT, accurate translation of NEs plays an important role in the translation of the overall text [12]. Also in many domain specific contexts, domain specific NER is the key technology for constructing terminology resources [13, 14, 15].

In addition to the general newswire documents, there are text documents in biomedical field where NEs refers to names of genes, proteins and diseases. NER in Biomedical domain termed as (BioNER) is an important preprocessing task for many further tasks such as relation extraction between NEs, knowledge discovery and hypothesis generation. In addition, the tremendous growth of publications in biomedical domain makes it vital to apply BioNER techniques as it is tough to extract NEs manually.

[6] https://trec.nist.gov/data/qa/t8_qadata.html

1.3 Motivation

BioNLP is a highly challenging research area and has been applied in several domains such as Health Informatics, Biological Analysis and Biomedical Knowledge Extraction for information representation and extraction. BioNER is a crucial preprocessing task in BioNLP pipeline as it is a prerequisite for extracting huge amount of high-valued biomedical knowledge deposited in unstructured text. The primary role of BioNER and the explosive growth of published biomedical research makes it vital to apply NER for biomedical literature.

BioNER is a tricky task as the linguistic and orthographical structures of NEs in biomedical texts are different from that in the general newswire domain. There are no specific rules or conventions that researchers in biomedical area can following while naming biomedical entities. Moreover, the usage of symbols and Latin letters in Biomedical NEs (BioNEs) add more complexity to BioNER. All these challenges demand the need for designing efficient approaches for BioNER systems.

Traditional NER techniques which are applied for newswire documents cannot be used directly to extract BioNEs due to the complexity of BioNEs. Usually single classifiers are used to classify the BioNEs. But, a single classifier may perform well on some data and may not give good results for some other data. Also, there is no mechanism to select a classifier that performs well. This problem may be overcome by using a pool of classifiers and combining their output. Developing BioNER system depends on the availability of annotated data which is used for training the classifiers. Hence, improving data annotation methods is an important factor for BioNER systems.

The key motivation factors that led us to pursue research in the area of BioNER are; lack of methods to annotate data [16] and less accurate models [17]. The existing state-of-the-art methods for BioNER are focused on tuning models to improve results instead of addressing different annotation techniques. Further, the performance of state-of-the-art models for BioNER systems are less compared to that of the general domain. In order to overcome the lacuna, there is need to explore different annotating techniques to improve the performance of BioNER system.

1.4 Thesis Contributions

The main contributions of the thesis are:

1. Study of the relation between Segment Representation (SR) schemes and the performance of BioNER systems
2. Development of an ensemble-based BioNER system, which considers SR schemes as base models.
3. Development of an ensemble approach for the Indian Native Language Identification (INLI) task
4. Development of an efficient deep Learning (DL) model for Disease-NER
5. Improving BioNER for multi-word BioNEs using extended SR scheme.

1.5 Thesis Outline

The remainder of the thesis is organized as follows: *Chapter 2* presents an overview of BioNER, focusing on the challenges of BioNER, approaches used for BioNER, datasets available and evaluation metrics for BioNER. *Chapter 3* compares the performance of BioNER with different SR schemes. A detailed overview of SR models and the two popular Machine Learning (ML) algorithms, Conditional Random Fields (CRF) and Support Vectors Machines (SVM) used for building the BioNER models have been presented. *Chapter 4* presented an ensemble based approach for BioNER system based on majority voting, weighted majority voting and stacking. These systems use different SR schemes and different classification algorithms as variations of base-classifiers. Implementations for INLI task based on SVM, Random Forest Trees, Multinomial Naïve Bayes and Ensemble approaches have been presented. *Chapter 5* introduces a DL based model for Disease-NER. Artificial Neural Networks (ANN) have been used in developing the model. *Chapter 6* introduces FROBES, an extended SR scheme designed for improving the performance of BioNER systems for multi-word BioNEs. A Bidirectional Long Short-Term Memory (BiLSTM) based model has been used for comparing the results of different schemes. *Chapter 7* concludes this thesis and discusses how work carried out in this thesis can be extended further.

1.6 Chapter Summary

In this chapter, the concepts and importance of TM and NLP have been introduced. Furthermore, the need for applying NLP techniques for biomedical texts has been discussed. The impact of BioNER for BioNLP, the importance of BioNER and research objectives have been discussed.

Chapter 2

Background for BioNER

BioNER is an important preprocessing task for many further tasks such as relation extraction between entities, knowledge discovery and hypothesis generation. In addition, the tremendous growth of publications in biomedical research area makes it vital to apply BioNER, as it is tough to extract NEs manually. In this chapter, we introduce BioNER task, challenges of BioNER, approaches for BioNER, available resources for BioNER and performance evaluation of BioNER systems.

2.1 Introduction

NER is the first and most important task which aims at identifying NEs in a given text for further tasks such as information extraction, knowledge discovery and hypothesis generation as shown in Figure 2.1. It is highly significant in TM and it has many applications such as intelligent search (searching for documents by entities occurring in them), MT (usually NEs are transliterated during translation), automatic summarization (NEs contained in a document are usually contained in it's summarization) and so on.

NER has applications in other NLP tasks such as improving the results of MT and QA systems. In document indexing, NER is used to properly find documents related to some person or organization and in sentiment analysis, it is used to relate outcomes to products.

Early work in NER systems started in "Message Understanding Conference 6" (MUC-6) [18] in 1995, where the main task was to recognize NEs, such as the name of persons, organizations, locations and artifacts, from newswire journal articles. Three classes of NEs were defined in MUC-6 as follows:

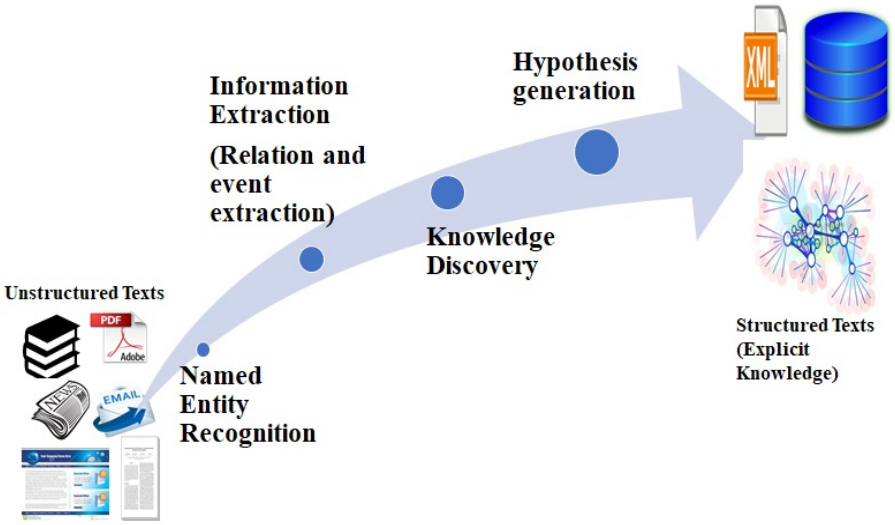


Figure 2.1: TM tasks

i `TIMEX` phrases for temporal expressions such as "*August 2*" and "*noon PST*"

ii `NUMEX` phrases for numeric expressions such as "2.7%" and "₹50 million"

iii `ENAMEX` phrases for proper names such as "*Amitabh Bachchan*", "*Mumbai*" and "*Indian Council for Cultural Relation*".

In addition to newswire documents there are several repositories for biomedical literature, such as Medical Literature Analysis and Retrieval System Online (MEDLINE[1]), PubMed[2] and PubMed Central (PMC)[3]. MEDLINE is a huge bibliographic database of journal citations and abstracts maintained by US National Library of Medicine (NLM). It is known as a reliable and comprehensive source of references into peer-reviewed fundamental biomedical studies, as well as clinical evidence. The scope of the database is biomedicine and health broadly defined to cover life sciences, behavioural sciences, chemical sciences, and bioengineering. MEDLINE contains over 24 million references to articles from approximately 5,000 journals dating back to 1950's[4] and recently the number of citations in MEDLINE is growing fast as shown in Figure 2.2. Basic information such as the title of the article, authors, journal and publication type, date of publication, language and a link to the publishers web site to request or view the full article is provided for each MEDLINE

[1]https://www.nlm.nih.gov/pubs/factsheets/medline.html
[2]https://www.ncbi.nlm.nih.gov/pubmed/
[3]https://www.ncbi.nlm.nih.gov/pmc/
[4]http://dan.corlan.net/medline-trend.html

citation. PubMed provides free access to full text open access articles from MEDLINE, life science journals and online books. It is a search engine that indexes titles, abstracts and metadata separately. These indices allow users to specify which fields or indices should be searched. PMC is a free full text archive of biomedical and life sciences journal literature. Regularly, the full text of scientific publications becomes available for research purposes through digital archives of biomedical literature like PMC archive developed by NLM. It is a part of the PubMed Central International Network, which currently includes the U.S. PubMed Central and UK PubMed Central. This network also includes PMC Canada, which provides access to all publications resulting from the Canadian Institutes of Health Research.

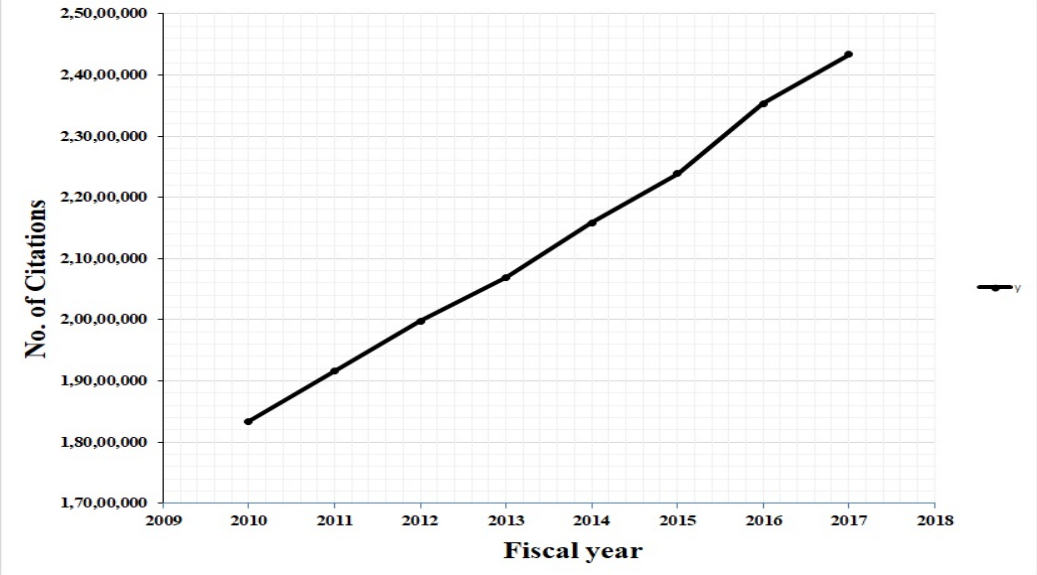


Figure 2.2: Total Number of citations in MEDLINE per Year

Literature in biomedical domain consists of the name of BioNEs such as genes (e.g *"Tp53"*, "*agaR*", *"IL-2 gene"*), proteins (e.g *"p53"*, *"NF-Kappa B"*, *"IL-2"*), cells (e.g *"T-cell"*, *"Human malignant mesothelioma"*, *"Saos-2"*), drugs (e.g *"Cyclosporine"*, *"herbimycin"*), chemicals (e.g *"5'-(Nethylcarboxamido) adenosine (NECA)"*), diseases (e.g *"colorectal cancer", "pendred syndrome", "thyroid goiter"*) and so on. The task of identifying NEs in biomedical domain is termed as BioNER. BioNER is an essential preprocessing task for various applications such as protein-protein interaction, gene-protein interaction, relation extraction, QA systems, hypothesis generation and intelligent document access (browsing

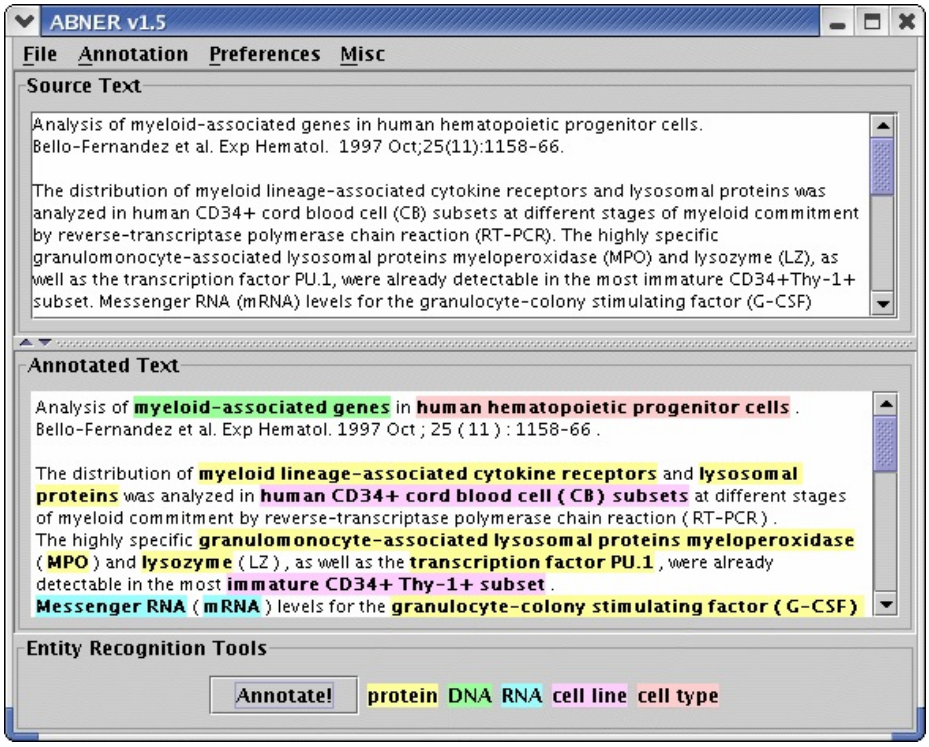


Figure 2.3: ABNER: A tool for BioNER

document collections by biomedical entities that occur in them). While NER in the general newswire domain has been well studied since the 1990s, BioNER is still in its infancy. For example, Figure 2.3 shows a BioNER task using ABNER [19].

2.2 Challenges in BioNER

The recognition of BioNEs in biomedical literature is not straightforward, because of several challenges related to ambiguous names, synonyms, variations, and newly issued names. Most of these problems arise as there are no solid naming conventions in the community, although guidelines for nomenclature do exist.

2.2.1 Ambiguity

Ambiguity is an open issue that affects the performance of BioNER systems. This problem occurs when the same word or phrase refers to different entities. For example, *"bcl-2"* can be a name of a protein or a DNA. While some BioNEs may be confused with common English words, such as *"can", "vamp"* and *"cycle"* [20], other names may denote biomedical

entities of different classes. For example, *"Friedrich atoxia"* refers to the name of a gene as well as a disease. Likewise, "*CD4*" can be a cell name, as well as a protein name.

2.2.2 Synonyms

While an ambiguous name may denote more than one entity, an entity also can be denoted by various names in a synonymy relation, also called as aliases. For example, *"caspase-3"*, *"CASP3"*, *"apoptosis-related cysteine protease"*, and *"CPP32"* denote the same entity [21]. Using synonyms causes ambiguity, as any of the aliases can be used to denote the same entity.

2.2.3 Variations in BioNEs

Variations in BioNEs denote the same entities by definition. There are many levels of variations including character-level variations, word-level variations, word order variations, syntactic variations and variations with abbreviations [21].

- The character-level variations in NEs are due to the presence or absence of special characters, or exchange of indices, such as digit and single alphabet. Some examples are: *"D(2)"* or *"D2"*, *"SYT4"* or *"SYT IV"*, *"IGA"* or *"IG alpha"*.
- A word in a name can be replaced by another word, or omitted in a variant of the name. For example, *"RNase protein"* or *"RNase P"* refer to the same entity.
- The strings of words in word-order variations show a different word order. For example, *"intergin alpha 4"* or *"alpha4 intergin"* refer to the same entity.
- A subsequence of a full name can be replaced with its abbreviation. For example, *"Placental anticoagulant protein I"* or *"PAP-I"*.

The variations in BioNEs make it difficult to create inclusive dictionaries, where same entity has different variations.

2.2.4 Newly discovered BioNEs

Another major source of problems for effective recognition of BioNEs is the overwhelming growth rate in the discovery of new names frequently [22]. In this case, dictionaries and lexicons have to be updated frequently to create effective BioNER systems. BioNER

system must be scalable in the sense that it should be able to recognize most if not all BioNEs in biomedical articles in plausible time and should also own the ability to resolve previously unseen BioNEs.

2.2.5 Language Phenomena

Most of the challenges of BioNER spring from a diversity of ambiguities involved in biomedical articles. From language perspective, much ambiguity exists as a result of different linguistic phenomena like comprehensive lexical variations (i.e., names having several spelling variations representing the same concept, for example, "*SELL*" and "*selection L*"), homonymy or polysemy (i.e., one name has several meanings denoting several concepts, for example, the medical device *cat* and the noun *cat*). Some of the most important language phenomena commonly seen in biomedical text are as follows:

i **Abbreviations**

Abbreviations, concise forms of concepts or names are common in biomedical texts. Abbreviations are one of the main sources of ambiguity. Usually, an abbreviation can be interpreted as several different full forms, if it were not explicitly defined in the context. For example, *"ACE"* can be either *"Angiotensin Converting Enzyme"* or *"Affinity Capillary Electrophoresis"*, while *"EGFR"* might correspond to "*Epidermal Growth Factor Receptor*" or *"Estimated Glomerular Filtration Rate"*. In some cases, an abbreviation refers to different concepts even if it refers to the same complete form. For example, *"NF2"* refers to a name of a gene, the protein it produces, as well as the disease resulting from its mutation. Similarly, *"CAT"* can refer to a protein, a medical device or an animal.

ii **Tokenization**

Tokenization is the first phase of NER that includes splitting the articles into tokens. Usually a white space is used to separate token in most languages. There are different linguistic phenomena that can cause problems for tokenization such as abbreviations (e.g., *"major histocompatibility complex(MHC)"* or *"major histocompatibility complex (MHC)"*), apostrophes (e.g., *"IL-10's activity"*), hyphenation (e.g., *"IL-10"* or *"IL 10"*), different formats (e.g., "*123,456.235*" and "*123456.235*") and various sentence boundaries (e.g., ".", ":", ";", "!" or "?").

iii **Multi-word NEs**

Most of NEs consist of multiple words (e.g., *"CD28 surface receptor"*), and shorter

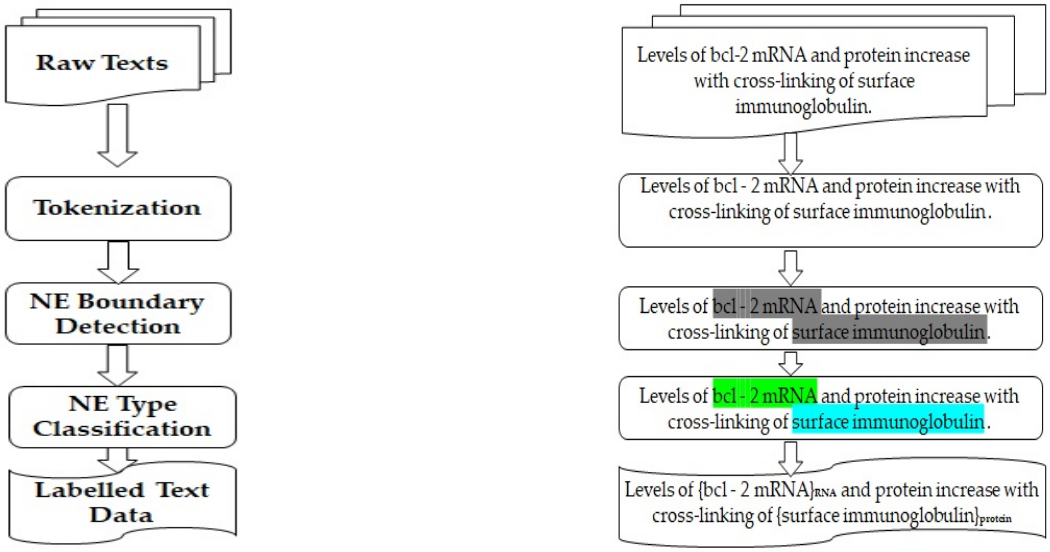
NEs can compound together to form longer NE (e.g., "*tumor*", "*VHL*" are single word NEs where as "*VHL tumor*" is a two word NE). This problem is more complicated as the name boundaries need to be determined and then overlapping of candidate names have to be resolved.

iv **Nested NEs**
One NE may occur within a longer NE (as a proper string). For example, a shorter NE *"T cell"* is nested within longer NE "*nuclear factor of activated T cells family protein*". In the GENIA corpus, about 17% of all NEs are nested with another NE [23]. Meanwhile, it is reported that about two-thirds of "Gene Ontology" (GO) terms contain another GO terms as a proper substring [24].

Many efforts have been put by researchers to tackle these challenges in the development of efficient BioNER systems.

2.3 General framework of BioNER system

A typical BioNER system is shown in Figure 2.4(b) and an example of a sentence from a research paper and the results of different phases of BioNER system are shown in Figure 2.4(a). Generally, BioNER system contains three phases, *tokenization*, *NE boundary detection*, and *NE type classification*. In tokenization phase, input text is separated into tokens/words which are contiguous text of meaningful words. It is one of the most important tasks of the BioNER system, since all the other tasks will be based on the tokens/words. Simple approach using white space or N-gram method is used for tokenization. The NE boundary detection phase decides whether a single token or several adjacent tokens represent an NE without considering the type of NE, thus distinguishing NEs from non-NEs. The basic problem in this phase is the ambiguity which occurs when the same word exists more than once in a biomedical text as a BioNE and as a regular word. For example, the word "*monocytes*" is an entity of `cell_type` class in addition to a regular word. Nesting of BioNEs is another snag for boundary detection. The NE type classification is a typical multi-class classification problem of assigning a specific class label from the predefined set of labels to each NE identified in the boundary detection phase. Type classification phase is usually carried out using ML approaches in combination with dictionary approach or rule-based approach. Practically, most of BioNER merges the boundary detection and type classification steps into one step.



(a) A typical BioNER system

(b) An example

Figure 2.4: A typical BioNER system with an example

2.4 Approaches for BioNER

Over the years, many approaches have been explored by researchers to develop efficient BioNER systems. The approaches for BioNER generally fall into five categories:-

2.4.1 Dictionary-based approaches

BioNEs in a text are identified by looking up a provided list of known BioNEs in the form of a dictionary, typically compiled from existing lexicons or databases [25]. There are some well-managed lexicons or databases for BioNEs such as *ChemIDplus*[5], *GenBank*[6] for genes and *UniProt*[7] for proteins. Limitations of this approach include false positive recognition caused by ambiguity, false negative recognition caused by synonyms and lack of unified source that covers newly discovered names [26]. Most lexicons used in this approach are constructed by experts in a specific domain and will usually have limited coverage in terms of size, due to the growth of new BioNEs. As the dictionaries are domain specific, dictionary based approaches are not portable to other domains.

[5]https://chem.nlm.nih.gov/chemidplus/ for chemicals
[6]https://www.ncbi.nlm.nih.gov/genbank/
[7]http://www.uniprot.org/

Dictionary based approaches use string matching algorithms to identify BioNEs. Many fast exact matching algorithms have been proposed by several researchers [27, 28, 29, 30, 31, 32]. However, spelling variations of an entity makes exact matching less attractive. For example, a protein name "*EGFR-1*" has the following variation "*EGR-1*", "*EGR 1*", "*Egr-1*", "*Egr 1*", "*egr-1*", and "*egr 1*". This problem is overcome by approximate string matching algorithms for NER.

2.4.2 Rule-based approaches

Rule-based approach makes use of hand-crafted rules by domain experts to directly match the candidate BioNEs in a given text. They deal with a broader range of variations, even covering a few of the word order variations and systematic variations [33]. Regular expressions are used to define the patterns that match with BioNEs. Highly specific rules lead to high precision, but in the utmost case, a new rule must be designed for every NE, causing low recall due to unseen NEs. On the other hand, rules that are too general lead to better recall but low precision. Limitations of rule based approach include domain specificity, domain experts to construct rules, strong trade off between precision and recall and lack of portability to other domains.

A classic example of rule-based system for BioNER proposed by Fukudo et al. [34] distinguishes between two components of protein names; "core terms" and "feature terms". Core terms contains sentence-medial capital letters, numbers and non-alphanumeric character, e.g. *"p53", "Src"* or *"Brca-1"*. Feature terms are fixed set of keywords describing the function or nature of a core term, e.g. "*receptor*" or "*protein*", as in "*EGF receptor*". The rules are described as context free grammar. A sample rule to identify a protein is as follows:

```
rule1:  protein --→ core_terms,feature_terms
rule2:  core_terms --→ medial_capital_letter | Numbers | Symbols
rule3:  feature_terms --→ receptor | protein | ...
```

2.4.3 Machine Learning approaches

ML approach uses a classification rule/function or a classifier to detect boundaries of BioNEs and classify them into one of the predefined categories. A classification rule $g : x \rightarrow Y$ can be formally defined as the task of estimating the label y of k-dimensional

input vector x where $x \in \mathbb{R}^k$ (note that, for most ML algorithms, input variables have to be real-valued) and $y \in Y = \{C_1, C_2, \ldots, C_m\}$. If the number of classes, $m = 2$, the classification is called as binary classification and for $m > 2$, the classification is called as multi-class classification. Intuitively, the classification rule g partitions the input space x into m disjoint decision regions whose boundaries are called decision boundaries. Several classifiers such as Naïve Bayes, k-Nearsest Neighbors, SVM [35], Hidden Markov Models (HMM) [36, 37], CRF [38] have been used for different applications.

A general ML framework for classification is shown in Figure 2.5. The framework accepts the training data which is a very large reliable annotated data. Distinguishable features are extracted from the training data depending upon the application. These features are then used to build the classification rule or a function which is validated using the test data.

BioNER can be thought of a special case of classification problem or a sequence labeling problem as it involves assigning a sequence of labels to a sequence of input words respectively. In this case, one of the predefined labels have to be assigned to a single word or a sequence of words in a sentence where a label may be the name of a class to which a word or a sequence of words belongs or a non-NE (the word is not a NE). The main reason for considering NEs in a sentence is because of the reason that NEs do not span over two subsequent sentences.

Formally, BioNER system can be defined as a classifier that accepts a sentence S, having n words in a sequence expressed as $S = < w_1, w_2, \ldots, w_n >$ and assigns one of the predefined labels $\{C_1, C_2, \ldots, C_m\}$ to each word or a sequence of words based on the characteristics of the words captured during the training phase. The classifier model is built by features extracted from the training data which is expressed as $T = \{< w_1, y_1 >, < w_2, y_2 >, \ldots < w_m, y_m >\}$ where w_i is the word and y_i is the corresponding tag of that word.

The main strength of ML approach is the improved performance than that of rule based and dictionary approach. Moreover, adaption to a new domain is easier as compared to rule based and dictionary based approach. Further, they do not require manual framing of rules and also can identify new BioNEs not covered by the rules or included in standard dictionaries. However, the major requirement of these approaches is very large reliable annotated resources. The process of annotating biomedical resources is time-consuming and

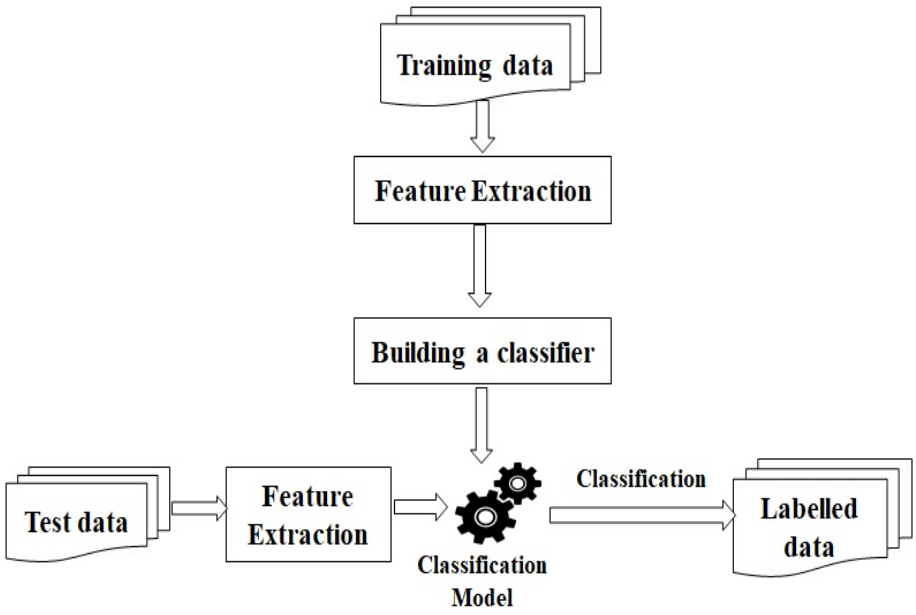


Figure 2.5: ML framework for classification

it should be done by domain experts.

Of late DL algorithms based on ANN [39] are being used to a larger extent for BioNER [17, 40]. DL is about learning multiple levels of representation and abstraction. This approach requires suitable word representations called word embeddings, a vector representation for words which are used as inputs for ANN model [41]. Word embeddings are static word representations that capture distributional syntactic and semantic information of the word. More information about word embeddings is given in Chapter 5.

Feature Extraction

Feature extraction plays a crucial role for the success of ML technique. A feature can be any piece of information that can be derived from the token and any other outside resource, barring the expected output tags. The classifier model exploits features of several various types (a detailed description is given in the corresponding chapter) including:

- **Dictionaries**: a collection of some NEs or English words that helps in determining a NE type
- **Orthographical features**: length of the word, capitalization, common bit information about the word form (contains a special character or not, has an hyphen inside

the word and so on)

- **Frequency information**: frequency of the word, the ratio of the words capitalized and lowercase appearances, the ratio of capitalized and sentence start frequencies of the word
- **Contextual information**: PoS tag, sentence and trigger words (the most frequent tokens in a window around the NEs)
- **Dynamic features**: include predicted class of a few preceding words.

Choosing the appropriate discriminative features helps the classifier to classify the words accurately.

2.4.4 Ensemble approach

Most of the applications use a single classifier. However, for some data, some classifier may give good results while other classifier may not perform well. Further, there is no generic rule which helps to choose a classifier for a particular application and data. So, instead of experimenting on single classifiers one by one in search of good results it will be beneficial to pool several such classifiers and then take the collective decision similar to the decision taken by a committee rather than an individual. This technique which overcomes the weakness of some classifiers using the strength of other classifiers is termed as ensemble and is gaining importance for various applications such as word segmentation, PoS tagging and word sense disambiguation. Ensemble classification uses a set of preferably weak, diverse and heterogenous classifiers as base classifiers and combines the output of these base classifiers in different ways to get the final output.

There are different approaches such as bagging, boosting and stacking to create ensemble classifiers [42]. In bagging, different training subsets are drawn with replacement from the entire training data and each training data subset is used to train each base classifier. The outputs of base classifiers are combined using majority voting. Boosting is similar to bagging, but the selection process of training subsets subsequently gives more weight to misclassified samples. In boosting, a sequence of weak classifiers has to be fit to weighted version of data and more weight is given to samples that were misclassified by earlier rounds. Stacking uses outputs of base classifiers to train a new model which is known as

meta-classifier [43] and the meta-classifier is used for final classification. In stacking, different classification algorithms can be used for training the base classifiers such as SVM, CRF and HMM. The classification algorithm used for training the meta-classifier should be different from those classification algorithms used for training the base models.

2.4.5 Hybrid approaches

Hybrid approaches mainly aim at combining two or more approaches into a single approach by exploiting the strength of each approach to achieve better performance when compared to individual methods [33]. In most cases, ML approaches are merged either with rule based approach or dictionary based approach. Dictionaries, lexicons or hand crafted rules are usually used subsequent to ML-based systems to improve results by resolving variations in BioNEs such as abbreviations and synonyms.

Patrick J. and Yefeng W. [44] have proposed a BioNER system using hybrid approach. They used a small set of features containing PoS tags, affixes, orthographical features, head nouns and N-gram features to build a maximum entropy based classifier. This classifier is followed by a set of rules used for BioNEs boundary corrections. They used GENIA corpus and the system reported a low f1-measure of 68.20%. Blau B. et al. [45] proposed a hybrid system for recognizing disease entities using variations of CRF algorithm to build a stacked ensemble system, which is followed by a fuzzy matching algorithm. The proposed system is designed to extract disease names only.

ML approaches are preferable to use as they can easily adopt to new domains as well as identify unseen entities. However, the major requirement of ML approaches is the large data set annotated by domain experts.

2.5 Datasets

To facilitate development, evaluation and comparison of algorithms in the area of BioNER, considerable efforts have been invested by researchers in creating standardized datsets. A brief introduction of some of the important resources are as follows:

2.5.1 JNLPBA 2004 Shared Task Dataset

JNLPBA shared task is an open challenge task of bio-entity recognition of technical terms in the domain of molecular biology [46]. It was held as a joint workshop of BioNLP/NLPBA 2004 [47]. The training set of the shared task originated from GENIA corpus v3.02 [48] contains a set of MEDLINE abstracts extracted using the MeSH search terms "*human*", "*blood cell*" and "*transcription factor*". These abstracts were annotated manually into 36 semantic classes. Among these classes, 5 classes namely *DNA*, *RNA*, *protein*, *cell_line* and *cell_type* were selected in JNLPBA shared task. The test set has been extracted using the same MeSH search terms of training set. The publication years for training set ranges over 1990 $\sim$ 1999, while for test set it ranges over 1978 $\sim$ 2001. Statistics of JNLPBA shared task dataset is shown in Table 2.1

	Training set	**Test set**	**Total**
No. of Abstracts	2000	404	2404
No. of Sentences	18546	3856	22402
No. of Words	472006	96780	568786
No. of Named Entities			
`protein`	30269	5076	35345
`DNA`	9533	1056	10589
`RNA`	951	118	1069
`cell_type`	6718	1921	8639
`cell_line`	3830	500	4330

Table 2.1: Statistics of JNLPBA 2004 shared task dataset

2.5.2 i2b2/VA 2010 Shared Task Dataset

The 2010 i2b2/VA challenge presented three tasks - entity extraction (extract problem, test and treatment), assertion classification (assign assertion type for medical problem concepts) and relation classification (assigning relation types that hold between medical problems, test and treatment) [49].

The dataset was created for entity and relation extraction purposes at i2b2/VA2010 challenge including 826 discharge summaries for real patients from the University of Pittsburgh

Medical Centre, Partners Health Care and Beth Israel Deaconess Medical Centre. Pittsburgh notes was used as a test set in i2b2/VA 2010 challenge, while other two sources were used as training set. Both test and training sets are manually annotated into three different entities "*treatment*", "*test*" and "*problem*". Statistics of the dataset is shown in Table 2.2.

	Training set	**Test set**	**Total**
No. of Documents	349	477	826
No. of Named Entities			
`Problem`	11968	18500	30468
`Treatment`	8500	13560	22060
`Test`	7369	12899	20268

Table 2.2: Statistics of i2b2 dataset

2.5.3 NCBI Dataset

NCBI disease corpus was introduced for disease name recognition and normalization [50]. It is the most inclusive publicly available dataset annotated with disease mentions. The corpus was manually annotated by 14 medical practitioners. General statistics of the dataset is given Table 2.3.

	Training set	**Development set**	**Test set**	**Total**
No. of Abstracts	593	100	100	793
No. of Sentences	5661	939	961	7261
Total Disease mentions	5145	787	960	6892
Unique Disease mentions	1710	368	427	2136

Table 2.3: Statistics of NCBI dataset

2.5.4 BC2GM dataset

BC2GM dataset is used for BioCreative II Gene Mention (BC2GM) task and it consists of 20,000 sentences extracted from PubMed abstracts. It is divided into training and test set

which contains 15,000 and 5,000 sentences respectively [51]. A BioNE class label named `GENE` and `O` (for non-BioNEs) are used to annotate the dataset.

2.5.5 BC5CDR dataset

BC5CDR dataset was created for BioCreative V Chemical Disease Relation (CDR) task and consists of 1500 PubMed articles with 5818 disease mentions [52]. The corpus was randomly split into three subsets: 500 each for training, testing and development sets. A BioNE class label named `DISEASE` and `O` (for non-BioNEs) are used to annotate the dataset.

2.6 Performance Evaluation

The performance of BioNER system is reported in terms of f1-measure [53]. F1-measure is a harmonic mean of Precision (P) and Recall (R). Denoting TP as the number of true positives, FP number of false positives and FN as the number of false negatives P, R and f1-measure are calculated as follows:

$$P = \frac{TP}{TP+FP}$$

$$R = \frac{TP}{TP+FN}$$

$$f1-measure = \frac{2*P*R}{P+R}$$

There are some other useful metrics, such as Full-match, Left-match and Right-match proposed by Tsai *et. al.* [54]. These metrics are defined as follows:

Full-match:	The predicted NE is reported as full-match, if its boundaries match with the boundaries of tagged NE on both sides
Left-match:	The predicted NE is reported as left-match, if its boundaries match with the boundaries of tagged NE on left side
Right-match:	The predicted NE is reported as right-match, if its boundaries match with the boundaries of tagged NE on right side

2.7 Chapter Summary

BioNER is a crucial task in BioNLP pipeline, because of explosive growth of research articles in biomedical area. In this chapter, we have introduced task and challenges of BioNER, different approaches for BioNER, the datasets available for BioNER and the performance metrics used for BioNER evaluation.

Chapter 3

Segment Representation Schemes

Annotating the dataset for training the classifier to recognize and classify NEs is a crucial task in BioNER. There are many methods used for annotating the datasets such as XML format, BioNEs offset and SR. SR is an efficient way of annotating BioNEs within a sentence in order to differentiate them from non-BioNEs. In this chapter, different SR schemes and relationship between them and the performance of BioNER systems using these schemes are presented. SVM and CRF have been used to train different BioNER models with the benchmark datasets JNLPBA 2004, NCBI, BC2GM, BC5CDR and i2b2 2010 shared task dataset using different SR schemes. Results obtained illustrates that CRF outperforms SVM for all SR schemes and IOBES SR scheme outperforms other SR schemes for JNLPBA 2004, BC2GM and BC5CDR datasets, while IOBE and IOE2 reported the highest performance for NCBI and i2b2 respectively.

3.1 Data Annotation

The main requirement for any ML algorithm is the large annotated corpus for training. In BioNER, the corpus is a collection of research articles, medical discharge summaries or patent abstracts. These corpora has to be annotated manually by domain experts by assigning suitable tags to BioNEs in order to differentiate them from non-BioNEs. Many methods such as XML format, standoff format and SR are used to annotate the corpus. XML format is an approach to annotate NEs in text using XML tags. An example of annotating the segment fragment "*Human APC2 maps to chromosome 19p13.3.APC and APC2 may therefore*

Some parts of the material of this chapter have appeared in the following research paper:
H. L. Shashirekha and Hamada A. Nayel: "A Comparative Study of Segment Representation for Biomedical Named Entity Recognition", In 2016 *International Conference on Advances in Computing, Communications and Informatics* (ICACCI), pages 1046-1052, Sep 2016.

Human APC2 maps to chromosome 19p13.3.APC and APC2 may therefore have comparable functions in development and <category="SpecificDisease">cancer</category>. The frequency, origin, and phenotypic expression of a germline MSH2 gene mutation previously identified in seven kindreds with <category="SpecificDisease">hereditary non-polyposis cancer syndrome</category> (<category="SpecificDisease">HNPCC</category>) was investigated .

Figure 3.1: Example of annotating text fragment using XML format

Interferon-gamma potentiates the antiviral activity and the expression of interferon-stimulated genes induced by interferon-alpha in U937 cells. Binding of type I interferon (IFN-alpha/beta) to specific receptors results in the rapid transcriptional activation, independent of protein synthesis, of IFN-alpha-stimulated genes (ISGs) in human fibroblasts and HeLa and Daudi cell lines. The binding of ISGF3 (IFN-stimulated gene factor 3) to the conserved IFN-stimulated response element (ISRE) results in transcriptional activation. This factor is composed of a DNA-binding protein (ISGF3 gamma), which normally is present in the cytoplasm, and other IFN-alpha-activated proteins which pre exist as latent cytoplasmic precursors (ISGF3 alpha). We have found that ISG expression in the monocytic U937 cell line differs from most cell lines previously examined. U937 cells express both type I and type II IFN receptors, but only IFN-alpha is capable of inducing antiviral protection in these cells. Pre-treatment with IFN-gamma potentiates the IFN-alpha-induced protection, but IFN-gamma alone does not have any antiviral activity.

(a) abstract01.txt

ID	Type	Begin	End	BioEntity
T1	Protein	0	16	Interferon-gamma
T2	Protein	583	594	ISGF3 gamma
T3	Protein	1014	1023	IFN-gamma
T4	Protein	1074	1083	IFN-gamma
T5	Protein	1128	1133	ISG15
T6	Protein	1275	1284	IFN-gamma
T7	Protein	1323	1332	IFN-gamma
T8	Protein	1798	1809	ISGF3 gamma
T9	Protein	1834	1843	IFN-gamma

(b) abstract01.ann

Figure 3.2: Example of annotation using standoff format

have comparable functions in development and cancer. The frequency, origin, and phenotypic expression of a germline MSH2 gene mutation previously identified in seven kindreds with hereditary non-polyposis cancer syndrome (HNPCC) was investigated." using XML format is shown in Figure 3.1, which contains three NEs of `SpecificDisease` class. This representation is appropriate for applying dictionary-based and rule-based approaches, but not suitable to apply ML-based approaches for BioNER, because ML-based framework for BioNER requires that each word should be assigned with a class label. Standoff format is a method used to represent BioNEs and relations between them in terms of their character offsets. In standoff format, raw texts and annotations are provided in separate files. For each raw text document, there is an associated annotation file. Within the document, individual annotations are connected to specific spans of text through character offsets. Example of using standoff format for annotating a research abstract is shown in Figure 3.2. In this example, for a raw text is given in Figure 3.2(a) the annotations are given in Figure 3.2(b). Standoff format was introduced to add more information about relation extraction and event extraction [55].

SR schemes are used to annotate NEs in the text by assigning each word a tag which differentiates between NEs and non-NEs. Figure 3.3 shows an example of annotating the sentence "*IL-2 gene expression and NF-kappa B activation through CD28 requires reactive oxygen production by 5-lipoxygenase.*" using SR approach. SR is widely used for tagging the word with one of the predefined types, as it is more appropriate to represent multi-word NEs.

IL-2	B-DNA
gene	I-DNA
expression	O
and	O
NF-kappa	B-protein
B	I-protein
activation	O
through	O
CD28	B-protein
requires	O
reactive	O
oxygen	O
production	O
by	O
5-lipoxygenase	B-protein
.	O

Figure 3.3: Example of using SR for annotating text fragment

Most of the SR schemes introduced by researchers are originally designed for PoS tagging or NP chunking (dividing sentences into non-overlapping noun phrases) [56, 57], while some techniques are designed to reduce ambiguity in these applications. However, SR schemes for NER have not received a similar degree of research attention in terms of their design and development [58], although handling ambiguity is one of the major challenges of NER [59]. As a result, most of the NER approaches use the existing SR schemes originally developed for other types of applications without any customization.

SR [58] is the process of assigning suitable class labels to the words in a given text. SR model comprises set of tags, which determine the position of a word in NE, combined with the class label that NE belongs to. The commonly used tags in SR techniques are B, E, I, S and O, the description of these tags is shown in Table 3.1. For example, if a word's tag is B-XXX, it means that the word is the first word of a NE of class XXX. The advantage of SR is that it can be used to represent multiple word NEs and nested NEs. Different SR

Tag	Abbreviation	Description
BEGIN	B	The first word of a multi-word NE.
INSIDE	I	An inner word of a multi-word NE.
END	E	The last word of a multi-word NE.
SINGLE	S	A single word NE.
OUTSIDE	O	A non-NE word.

Table 3.1: Different tags used for annotations and their description

models are used by researchers to tag the corpus. The primary SR model IO [60], assigns the tag I for the words inside the entity and the tag O for the words outside the entity, but is not able to represent the boundaries of two consecutive entities of the same class. IOB1 model has been introduced to solve this problem [61], by assigning the tag B to the first word of consecutive NEs of same class, while IOB2 model assigns the tag B for the first word of each NE [57]. IOE1 and IOE2 models use the same concepts of IOB1 and IOB2 respectively, but assigns the tag E to the last word of a NE [62]. Sun et al. [63] introduced IOBE model which concerns with the beginning and end of a NE. IOBE model assigns the tags B and E for the first and last word of all NEs respectively. IOBES model is a modified version of IOBE model that is concerned with single word NEs. In addition to IOBE tags, the IOBES model assigns the tag S to the single word NE. This model differentiates between a single word and multiple words NE. Example of tagging the text fragment "*Treatment / stay IHSS AF ESRD on HD, IgA nephropathy on...*" with IO, IOB1, IOB2, IOE1 IOE2, IOBE and IOBES models is shown in Table 3.2. This text fragment contains four NEs of class `problem` and a NE of class `treatment`.

The complexity of SR model can be viewed in terms of the number of tags it contains. More the number of tags higher will be the complexity. IO model is the most primitive model because it contains only two tags and the most complex model is IOBES because it contains maximum number of tags. Number of classes that words can be assigned depends on the number of tags of SR model. More tags results in more classes. The process of learning a classifier for IO model is faster than other models as it uses less labels and classification problem becomes easier, while IOBES is slower than others, because of more number of tags. The relationship between different SR models is shown in Figure 3.4.

	IO	IOB1	IOB2	IOE1	IOE2	IOBE	IOBES
Treatment	O	O	O	O	O	O	O
/	O	O	O	O	O	O	O
stay	O	O	O	O	O	O	O
IHSS	I-problem	B-problem	B-problem	E-problem	E-problem	B-problem	S-problem
AF	I-problem	B-problem	B-problem	E-problem	E-problem	B-problem	S-problem
ESRD	I-problem	B-problem	B-problem	E-problem	E-problem	B-problem	S-problem
on	O	O	O	O	O	O	O
HD	I-treatment	I-treatment	B-treatment	I-treatment	E-treatment	B-treatment	S-treatment
,	O	O	O	O	O	O	O
IgA	I-problem	I-problem	B-problem	I-problem	I-problem	B-problem	B-problem
nephropathy	I-problem	I-problem	I-problem	I-problem	E-problem	E-problem	E-problem
on	O	O	O	O	O	O	O

Table 3.2: An example of using different SR models

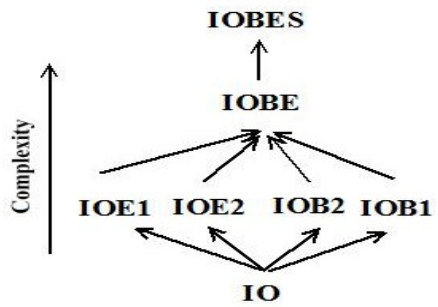


Figure 3.4: Relations between different SR schemes

3.2 Related Work

Many efforts have been made by researchers to improve the performance of existing BioNER systems and to develop new efficient BioNER systems in terms for boundary detection, NE type classification and enhancing SR models for BioNER. Konkol and Konopík [64] applied two ML based algorithms; CRF and maximum entropy and compared the performance of NER for different SR schemes. The general domain corpora from CoNLL02 [65] and CoNLL03 [66] shared tasks were used for English, Dutch, and Spanish languages, while for Czech language Czech NE corpus 1.1 [67] was used. This study may not be applicable for Biomedical domain directly due to the complexity of BioNEs. Han-Cheol

Cho et al. [58] studied various SR schemes for NLP tasks such as word segmentation, NER and shallow parsing. They presented a feature generation method for integrating multiple SR schemes into a CRF model. CoNLL03 and BC2GM shared tasks corpora were used to evaluate the proposed method for general and biomedical domain respectively [66, 51]. This study compares between different SR schemes for NER. However, some SR schemes are not entailed in this study. The corpus used for biomedical domain contains one class only and this may not reveal the conclusions accurately. An enhanced IOBES model to resolve the problem of ambiguity by adding new tag for ambiguous words was carried out by Sara Keretne et al. [16]. They used i2b2 dataset with multiple classifiers and reported that kNN classifier reduced the performance while C4.5 classifier improved the performance of BioNER.

Leman and Lu [68] designed a joint model based on semi-Markov model based linear classifier with a rich feature set approach for BioNER. Their model also performs normalization for BioNEs using semantic indexing. Nobata et al. [69] developed first ML approach for BioNER that used Naïve Bayes, decision trees and HMM approaches in both boundary detection and classification phases. Kazama et al. [70] used PoS, prefixes and suffixes features to learn the SVM based BioNER system. Their approach relies on splitting the non-entity class into sub-classes using PoS information. Lee et al. [35] developed a two-phase BioNER model based on SVM by adopting relevant features for each phase. They used GENIA corpus, and reported a f1-measure of 74.8% and 66.7% for boundary detection phase and semantic classification phase respectively. Keretna et al. [71] proposed a technique to extract drug names from unstructured and informal texts using hybrid model of lexicon-based and rule-based techniques. They used i2b2 2009 medication challenge dataset to evaluate the model and is able to achieve an f1-measure of 66.97%. Tsai [72] combined dictionary based approach with CRFs to improve the performance and reported f1-measure of 78.5%.

In this work, ML approaches, namely SVM and CRF have been used to study the performance of BioNER systems using different SR models, namely, IO, IOB2, IOE2, IOBE and IOBES.

3.3 Methodology

Framework of ML approach used for BioNER is shown in Figure 2.5, in Section 2.3. The model accepts the annotated data to build classifiers which will be used to tag the raw corpus with BioNEs tags.

3.3.1 Feature Extraction

Features, the properties or characteristics of words, are the keystones of ML algorithms. Features carry useful information about a single word or adjacent words. Various types of features such as word level features and character level orthographical features have been used in ML approaches. Word level features are those features which describe the characteristics of whole word or multiple words together. These features include word length, PoS tags, context words, chunk information, dynamic features, stopwords, non-English word, head nouns and inverse document frequency. Character level features describe the orthographical features of the word and include word affixes and word normalization.
Details of the features extracted are as follows:

1. **Word length**: This is a numeric value that determines the length of the current word. It is a good indicator for determining BioNEs with the observation that longer terms are likely to be BioNEs.

2. **PoS tags**: This is a very important feature as it determines the role of the word in the sentence and assigns a lexical category like `VB` (verb), `NN` (noun) or `JJ` (adjective) to the word. PoS feature helps in detecting BioNEs by detecting noun-phrases in the texts. For example, a word of "`NN`" PoS tag has higher probability of being an BioNE than a word of a "`VB`" PoS tag.

3. **Word affixes**: These are prefixes and suffixes of the current word. Prefix of length n is the first n characters of the word, while suffix of length n is the last n characters of the word. Affixes can suggest that the word is an BioNE or otherwise. For example, prefix "anti" in "antibiotic" suggests that the word is a medical treatment. All suffixes and prefixes up to length 5 have been used.

4. **Context words**:- These are the words surrounding the current word. This feature includes information about the prior and posterior words to the current word and is called as "*context window*". The context window of size n means n words before

the current word and n words after the current word, e.g. context window of size 3 is $w_{i-3}, w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, w_{i+3}$ where w_i is the current word. This feature is used under the principle that surrounding words carry effective information, such as PoS tag, length and affixes, for classifying BioNEs. It also can help in reducing the effect of ambiguity.

5. **Chunk Information**: Chunking means dividing the sentence into non-overlapping phrases. Chunk information is useful when determining the boundaries of multi-word BioNEs and is extracted using GENIA tagger V3.0.21. Figure 3.5 shows an example of chunk information feature for the sentence "*IL-2 gene expression and NF-kappa B activation through CD28 requires reactive oxygene production by 5-lipoxynease.*"

Word	Chunk Information
IL-2	B-NP
gene	I-NP
expression	I-NP
and	O
NF-kappa	B-NP
B	I-NP
activation	I-NP
through	B-PP
CD28	B-NP
requires	B-VP
reactive	B-NP
oxygene	I-NP
production	I-NP
by	B-PP
5-lipoxygenase	B-NP
.	O

Figure 3.5: Examples of chunk information feature

6. **Word Normalization**: Word normalization is the process of transforming a word into a uniform sequence. It assigns the same category to the words with same root and same shape and helps to overcome the problem variations in BioNEs. Two types of normalization namely stemming and word shape features are used. Stemming reduces the words to their morphological root. For example, the stem of "*receptors*" is "*receptor*" and the stem of "*activation*" is "*active*". There are two types of word shapes, general word shape and summarized word shape. In a general word, X is

Word	General word shape	Summarized word shape
receptors	xxxxxxx	x
Ca2+	XxdO	XxdO
1.5-fold	dOdOxxxx	dOdOx
UICC	XXXX	X

Table 3.3: Examples of word shape features

substituted for each capital letter, *x* for each lowercase character, *d* for consecutive digits and *O* for special characters. In a summarized word shape, consecutive capital letters are replaced by *X*, consecutive small letters by *x*, consecutive special characters by *O* and consecutive digits by *d*. Examples of word shape feature are given in Table 3.3

7. **Orthographic Features**:- An orthography is a set of rules used for writing in a specific language. Orthographic features are generalizations of the appearance of words and these features capture word formation information. Orthographic features such as capitalization, digit and special characters are entailed as field guides for discriminating texts. The set of all orthographic features extracted are shown in Table 3.4.

Feature	Example	Feature	Example
INITCAPS	Tonsillectomy	ALPHNUM	B12
ALLCAPS	MCV, RBC	GREEK	alpha
ENDCAPS	pH, proBNP	NUMBER	101.5
INCAPS	freeCa	HASATGC	LACTATE
CAPSMIX	cTropnT	PUNCT	INR(PT)
HASDIGIT	pO2,calHCO3	ROMAN	IV, CD
HASHYPHEN	hyper-CVAD		

Table 3.4: List of orthographic features and examples

8. **Dynamic Feature**: Denotes the predicted tags of the words preceding the current word. It is called dynamic feature because it is calculated during run time. Dynamic feature of size *n* denotes the predicted tags of *n* words preceding the current word.

If the current word is w_i, then dynamic features of size 3 are the tags $t_{i-3}, t_{i-2}, t_{i-1}$ corresponding to the words w_{i-3}, w_{i-2}, w_{i-1}. These features will help in determining multi-word BioNEs.

9. **Stop Words**:- Stop-words are common words that normally exist with high frequency rates and they do not add much meaning to text. Examples of stop words include pronouns, determiners and propositions such as "*the*", "*and*" or "*it*". This feature is a binary value that checks whether the current word is a stop word or not.

10. **Non-English Word**:- This is a binary value which fires only if the word exists in the dictionary. In this work, we used Grady augmented dictionary in `qdapDictionaries` package in `R` software [73]. This dictionary contains Grady Ward's English words augmented with proper nouns and contractions[74]. This feature is helpful because most of BioNEs contains non-English terms.

11. **Head Nouns**: The noun phrase that describes the functionality or property of a BioNE is called head noun [75]. Head nouns are very important as these play a key role for correct classification of a BioNE class. Unigrams and bigrams are used as head nouns. For example, "*examination*" is the head noun of "*cardiovascular examination*".

12. **Inverse Document Frequency (IDF)**: IDF - an indicator for the word informativeness is a statistical measure used to evaluate the rareness of a word in a corpus [76]. BioNEs usually have lower frequency rates as compared with non-BioNEs words. IDF of all words occurring in training corpus is computed as follows:

$$IDF(t,D) = \log \frac{1+N}{1+|\{d \in D : t \in d\}|}$$

where, t is the word, D is the corpus and $N = |D|$.

3.3.2 Classification

SVM and CRF classifiers have reported a high performance as compared to other classifiers according to literature review. In this work, SVM and CRF classifiers have been used to build BioNER systems based on different SR schemes.

Support Vector Machines

SVM is a supervised learning technique that learns a discriminative function from the training data. It belongs to a family of generalized linear classifiers where the boundary function is a hyperplane in the feature space which is used for classification or regression. SVM achieve generalization by maximizing the margin and they support an efficient learning of non-linear functions using the kernel trick. It is a binary classifier which creates a hyperplane that discriminates between the two classes [77]. However, it can be extended to multi-class problems by creating several binary SVM and combining them using a one-vs-rest approach or one-vs-one approach [78]. In the one-vs-rest approach, for k classes included in the training data, there will be k binary SVM classifiers. The training set of i^{th} SVM composed of all samples of i^{th} class will be labeled as positive samples and all samples from other classes will be labeled as negative samples. Each binary SVM classifier will predict the label of the new input and the SVM classifier with the highest output determines the class of input data. In the one-vs-one approach, a multi-class model which is based on majority voting on the combined binary classifiers will be used. In total $k(k-1)/2$ individual binary SVM classifiers are required one for each pair [79], where k is the number of classes. Since, each classifier has one vote; the class with the highest number of votes will be selected.

Mathematical Formulation of SVM The training set is represented by data points $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ where x_i is an n-dimensional real vector, y_i is either 1 or -1, denoting the class to which the data point x_i belongs. The SVM classification function (or hyperplane) $f(x)$ takes the form

$$f(x) = w \cdot x + b \tag{3.1}$$

where, w is the weight vector and b is the bias, which will be computed by the SVM in the training process. For every point $x_i \in D$, the function f must return positive numbers for positive data points and negative numbers otherwise,

$$w \cdot x_i + b > 0 \quad \text{if} \quad y_i = 1 \text{ , and}$$
$$w \cdot x_i + b < 0 \quad \text{if} \quad y_i = -1$$

These conditions can be revised into:

$$y_i(w \cdot x_i + b) > 0, \ \forall (x_i, y_i) \in D \tag{3.2}$$

D is called linearly separable, if there exists such a linear function f that correctly classifies

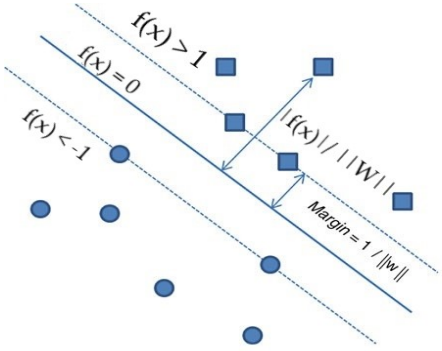


Figure 3.6: Example of hyperplane

every point on D or satisfies Equation 3.2. The hyperplane f needs to maximize the distance from the hyperplane to the closest data points, which are called margins. An example of such a hyperplane is illustrated in Figure 3.6. To achieve this, equation 3.2 is revised into the following:

$$y_i(w \cdot x_i + b) \geq 1, \ \forall (x_i, y_i) \in D \tag{3.3}$$

The distance from the hyperplane to vector x_i is formulated as $\frac{|f(x_i)|}{\|w\|}$. Thus, the margin becomes $\frac{1}{\|w\|}$, because $f(x_i)$ will return 1 for closest vectors x_i according to Equation 3.3. The closest vectors that satisfy Equation 3.3 with the equality sign are called "*support vectors*". Thus, the training problem in an SVM becomes a constrained optimization problem as follows:

$$\text{Minimize:} \quad Q(w) = \frac{1}{2} \parallel w \parallel^2 \tag{3.4}$$

$$\text{Subject to:} \quad y_i(w \cdot x_i + b) \geq 1, \ \forall (x_i, y_i) \in D \tag{3.5}$$

The formulation of SVM assumes that there is a separating hyperplane. However, in most real case problem there is no separating hyperplane due to noisy data labels as shown in Figure 3.7. For such cases, a *soft margin* SVM allows mislabelled data points while maximizing the margin. This is done by introducing a slack variable, ζ_i, which measures the degree of misclassification. Thus, the optimization problem for soft margin SVM will

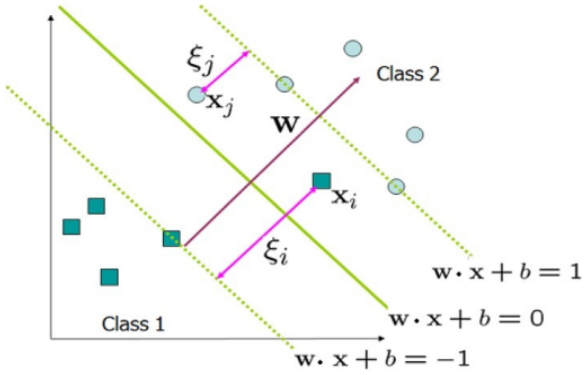


Figure 3.7: Linear SVM with soft margin

be as follows:

$$\text{Minimize:} \quad Q(w) = \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \zeta_i \tag{3.6}$$

$$\text{Subject to:} \quad y_i(w \cdot x_i + b) \geq 1 - \zeta_i, \ \ \forall (x_i, y_i) \in D \tag{3.7}$$

where $C \geq 0$ is a parameter that controls training errors allowed and ζ_i is a slack variable which measure the degree of misclassification of the instance x_i.

After training, the hyperplane or discriminative function f is obtained. At application time, previously unseen data are mapped onto the same space and the predictions are based on side of the hyperplane the new instances fall on.

The SVM model as discussed so far works well for data which is linearly separable as in the case of the example in Figure 3.6. However, in some cases, the data are not separable linearly. To fit such data usually a kernel trick is used to replace every dot product by a non-linear kernel function such as Polynomial kernel, Gaussian kernel and Radial Basis Function (RBF)[80]. This allows the algorithm to fit the hyperplane in a transformed feature space which is nonlinear. Thus although the classifier will be a hyperplane in the higher dimensional feature space, it will be nonlinear in the original input space.

Masayuki A. and Yuji M. introduced SVM to NER in 2003 [81]. The classifier is trained using words, extracted features and the tags (class labels) as data points and is used to predict the tag (class label) for each word in the given input.

Conditional Random Fields

CRF [82] are undirected graphical models, a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, CRF can easily incorporate a large number of arbitrary, non-independent features while having efficient procedures for non-greedy finite-state inference and training. CRF is used to calculate the conditional probability of values on designated output nodes given values on the designated input nodes. The conditional probability of a state sequence $S = <s_1, s_2, \ldots, s_T>$ given an observation sequence $O = <o_1, o_2, \ldots, o_T>$ is calculated as:

$$P(S|O) = \frac{1}{Z_0} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, O, t)\right) \tag{3.8}$$

where

t	is an iterator over all the words related to the current state
T	is the length of the sentence
k	is an iterator over all features of the current word
λ_k	is the weight of k-th feature
$f_k(s_{t-1}, s_t, O, t)$	is the feature function

When defining the feature functions $f_k(s_{t-1}, s_t, O, t)$, a set of real-valued functions $b(O, t)$ of the observation sequence has to be constructed in order to express some characteristic of the training data. An example of such a function is

$$b(O,t) = \begin{cases} 1, & \text{if the observation at position } t \text{ is the word ``anti-CD4''} \\ 0, & \text{otherwise} \end{cases}$$

Each feature function takes one of these real-valued functions $b(O, t)$ if the current state or previous and current states take particular values. For example, consider the following state function:

$$g(s_t, O, t) = \begin{cases} b(O,t), & \text{if } s_t = \text{B-PROTEIN} \\ 0, & \text{otherwise} \end{cases}$$

Then, feature function takes the form:

$$f_k(s_{t-1}, s_t, O, t) = \begin{cases} b(O,t), & \text{if } s_{t-1} = \text{O and } s_t = \text{B-PROTEIN} \\ 0, & \text{otherwise} \end{cases}$$

The model assigns an individual weight λ_k to each feature function f_k. These weights are learnt from the training data. If $\lambda > 0$, whenever function f is active (i.e., we see the word "anti-CD4" at a current position and we assign it label "B-PROTEIN", it increases the probability of the label sequence y. This is another way of saying the model should prefer the label "B-PROTEIN" for the word "anti-CD4". On the other hand, if $\lambda < 0$, then the model avoids assigning the label "B-PROTEIN" for "anti-CD4"

The values of the feature functions may range between $-\infty, \ldots, \infty$, but typically they are binary. To make all conditional probabilities sum up to 1, the normalization factor

$$Z_0 = \sum_s \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k \times f_k(s_{t-1}, s_t, O, t)\right) \tag{3.9}$$

is calculated efficiently by dynamic programming.
To train a CRF, the objective function L to be maximized is the penalized log-likelihood of the state sequences given the observation sequences:

$$L = \sum_{i=1}^{N} \log(P(s^i|o^i)) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \tag{3.10}$$

where

$<s^i, o^i>$	is the labeled training data
i	is an iterator over the words
k	is an iterator over all features of the current word
λ_k	is the weight of k-th feature
$\sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$	is the Gaussian prior on λ to regularize the training.

The predicted tag sequence $\hat{s}$ is calculated as follows:

$$\hat{s} = \textbf{argmax}\{P(\mathbf{S}|\mathbf{O})\}$$

When implementing CRF for the NER task, the sequence of the words in a sentence is assumed as an observation sequence and the state sequence as its corresponding tag sequence. The context window plays an important role in building the classification model, because CRF assumes that the observations (words) are dependent. This assumption makes CRF a suitable classification model for NER.

3.4 Experiments and Results

JNLPBA 2004 shared task, NCBI, BC2GM, BC5CDR and i2b2 2010 shared task benchmark datasets (described in Section 2.5) are used to study the performance of different SR models. SVM and CRF are applied to study the performance of different SR models for BioNER. GENIA tagger V3.0.21[1] is used to extract PoS, chunk information and word stem. Orthographic features such as capitalization and digitization are extracted using regular expressions in `R` programming language [73]. The list of stop words provided by `tm` package in `R` for English is used to determine stop words in the dataset.

3.4.1 Training

SVM classifiers are designed using YamCha[2] toolkit along with TinySVM-0.09[3] for the five SR models namely, IO, IOB2, IOE2, IOBE and IOBES. A context window of size 3 is used and dynamic features are set at three. The toolkit used for training the classifiers implements polynomial kernels only. In our experiments, we have applied second degree polynomial kernel $k(x,y) = (1+x^{\top}y)^2$ using one slack variable. The number of class labels for each dataset is greater than two, thus one-vs-one method is used for multi-class classification.

The CRF classifiers are designed using CRF++[4] for the five SR models namely, IO, IOB2, IOE2, IOBE and IOBES. Context window is set at 3 and the features parameters of CRF++ are given in a template file, which follows the settings from CRFs++ toolkit. The template file provides unitary templates and binary templates. The templates are given in $[position, column]$ format, where $position$ stands for the relative position of the word in focus and $column$ stands for the feature column number in the input file. Examples of using feature template is given below:

U00:%x[0,0]	A unary template for using the current word
U02:%x[1,5]	A binary template for using 5^{th} feature of the next word
U40:%x[0,2]/%x[-1,1]	A binary template for using the conjunction of 2^{nd} feature of the current word and 1^{st} feature of the previous word

The templates used for our experiments are given in Table 3.5.

[1] http://www.nactem.ac.uk/GENIA/tagger
[2] `http://chasen.org/~taku/software/yamcha/`
[3] `http://chasen.org/~taku/software/TinySVM/`
[4] https://taku910.github.io/crfpp/

U000:%x[0,0]	U052:%x[0,0]/%x[1,0]
U001:%x[0,1]	U053:%x[0,0]/%x[-1,0]
⋮	U054:%x[0,0]/%x[2,0]
U0025:%x[0,25]	U055:%x[0,0]/%x[-2,0]
U026:%x[1,0]	U056:%x[0,0]/%x[1,0]/%x[-1,0]
U027:%x[1,1]	U057:%x[0,0]/%x[1,0]/%x[-1,0]/%x[2,0]/%x[-2,0]
⋮	U058:%x[0,0]/%x[0,1]/%x[0,2]/%x[0,4]
U051:%x[1,25]	U059:%x[0,0]/%x[0,1]/%x[0,3]/%x[0,4]
U026:%x[-1,0]	U060:%x[0,0]/%x[0,2]/%x[0,3]/%x[0,4]
U027:%x[-1,1]	U061:%x[0,0]/%x[0,1]/%x[0,2]/%x[0,3]/%x[0,4]
⋮	
U051:%x[-1,25]	

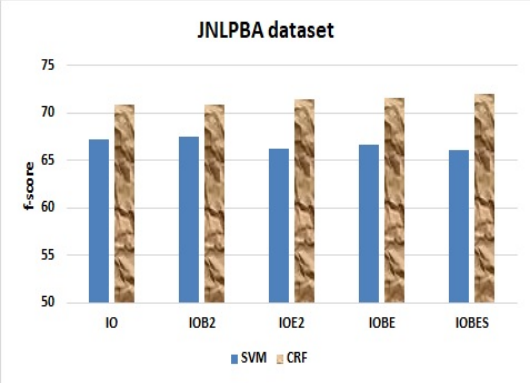
Table 3.5: Template features used for CRF++

3.4.2 Results and Discussion

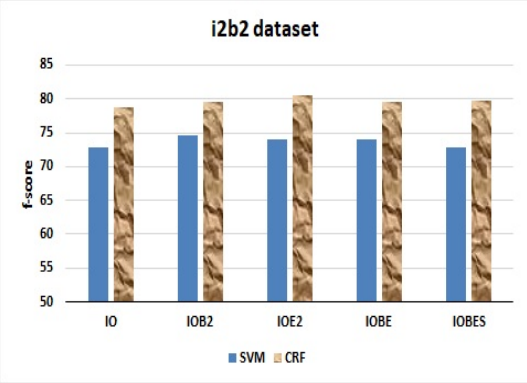
The results in terms of f1-measure using SVM and CRF for different SR schemes are shown in Figure 3.8. These results illustrate that CRF classifiers perform better than SVM classifiers for all SR schemes. The detailed results in terms of Precision (P), Recall (R) and f1-measure for left match, right match and exact match are given in Table 3.6, Table 3.7, Table 3.8, Table 3.9 and Table 3.10.

It is clear from the results that IO model has not given best performance for any dataset. As it uses only I and O tags, there is no strict identification for both boundaries. Complex SR schemes models such as IOBE and IOBES were introduced for better representations for multi-word NEs. These schemes recorded the best f1-measure for NCBI, JNLPBA, BC2GM and BC5CDR datasets, while IOE2 scheme recorded the highest f1-measure for i2b2 dataset.

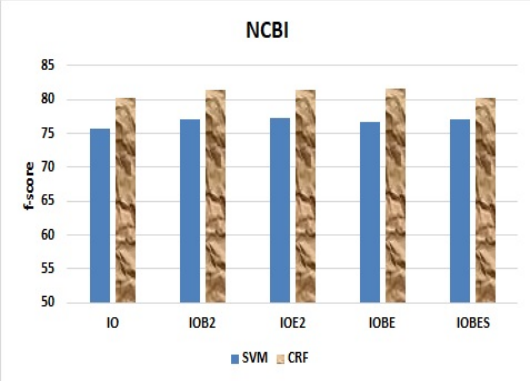
From the results, it is clear that the simpler SR techniques give lower performance, while the more complex one give high performance. IO scheme fails to represent consecutive BioNEs, while IOBES scheme can represent consecutive BioNEs and distinguish between single and multi-word BioNE. Performance of other schemes ranges between the performance of IO and IOBES models.



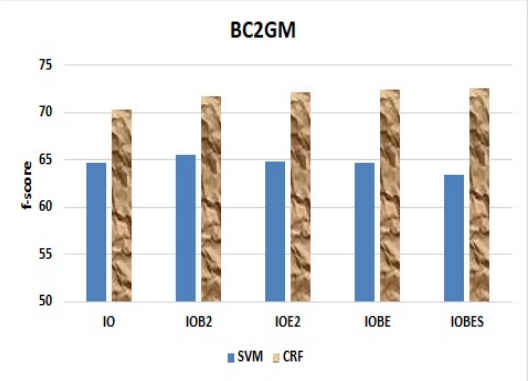
(a) Results of JNLBPA 2004 dataset



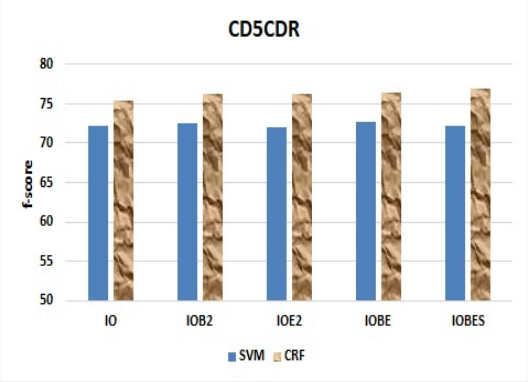
(b) Results of i2b2 2010 dataset



(c) Results of BC2GM dataset



(d) Results of NCBI dataset



(e) Results of BC5CDR dataset

Figure 3.8: f1-measure of CRF and SVM

SR scheme	Boundary	SVM			CRF		
		R	P	f1-measure	R	P	f1-measure
IO	Exact match	66.17%	68.41%	67.27%	72.21%	69.52%	70.84%
	Left match	70.05%	72.42%	**71.22%**	76.45%	73.59%	74.99%
	Right match	74.11%	76.61%	**75.34%**	80.34%	77.34%	78.81%
IOB2	Exact match	66.74%	68.35%	**67.54%**	72.74%	69.18%	70.92%
	Left match	70.35%	72.05%	71.19%	76.67%	72.91%	74.74%
	Right match	74.31%	76.11%	75.20%	80.76%	76.80%	78.73%
IOE2	Exact match	65.29%	67.35%	66.30%	73.33%	69.76%	71.50%
	Left match	69.31%	71.50%	70.39%	77.13%	73.37%	75.20%
	Right match	73.23%	75.54%	74.37%	80.78%	76.84%	78.76%
IOBE	Exact match	66.01%	67.46%	66.73%	73.45%	69.91%	71.64%
	Left match	70.13%	71.67%	70.90%	77.11%	73.40%	75.21%
	Right match	73.77%	75.39%	74.57%	80.93%	77.03%	78.93%
IOBES	Exact match	64.85%	67.46%	66.13%	73.83%	70.23%	**71.98%**
	Left match	68.99%	71.78%	70.36%	77.43%	73.65%	**75.50%**
	Right match	72.71%	75.64%	74.15%	80.95%	77.00%	**78.93%**

Table 3.6: Results of JNLPBA shared task dataset

SR scheme	Boundary	SVM			CRF		
		R	P	f1-measure	R	P	f1-measure
IO	Exact match	68.02%	78.43%	72.85%	75.49%	82.37%	78.78%
	Left match	71.98%	83.00%	77.10%	0.80.04%	87.34%	83.53%
	Right match	73.46%	84.71%	78.69%	0.80.40%	87.73%	83.91%
IOB2	Exact match	69.69%	80.08%	**74.52%**	76.07%	83.24%	79.49%
	Left match	72.73%	83.58%	**77.78%**	79.61%	87.12%	83.19%
	Right match	73.79%	84.80%	**78.91%**	80.18%	87.75	83.80
IOE2	Exact match	68.96%	80.06%	74.09%	77.28%	84.12%	**80.55%**
	Left match	71.99%	83.57%	77.35%	80.79%	87.94%	**84.21%**
	Right match	73.38%	85.19%	78.85%	81.30%	88.49%	**84.74%**
IOBE	Exact match	69.12%	79.57%	73.98%	75.98%	83.59%	79.60%
	Left match	72.49%	83.46%	77.59%	79.46%	87.41%	83.25%
	Right match	73.37%	84.46%	78.53%	79.97%	87.98%	83.79%
IOBES	Exact match	67.53%	79.21%	72.91%	76.10%	83.63%	79.69%
	Left match	71.05%	83.34%	76.70%	79.47%	87.34%	83.22%
	Right match	71.82%	84.24%	77.54%	80.14%	88.07%	83.92%

Table 3.7: Results of i2b2 shared task dataset

SR scheme	Boundary	SVM			CRF		
		R	P	f1-measure	R	P	f1-measure
IO	Exact match	70.52%	81.76%	75.73%	76.67%	84.31%	80.31%
	Left match	73.02%	84.66%	78.41%	79.17%	87.06%	82.92%
	Right match	77.92%	90.34%	83.67%	84.06%	92.44%	88.05%
IOB2	Exact match	71.77%	83.31%	77.11%	77.92%	85.29%	81.44%
	Left match	73.54%	85.37%	79.02%	79.69%	87.23%	83.29%
	Right match	77.92%	90.45%	83.72%	84.38%	92.36%	88.19%
IOE2	Exact match	72.08%	83.17%	**77.23**%	77.81%	85.47%	81.46%
	Left match	74.38%	85.82%	**79.69**%	79.48%	87.30%	83.21%
	Right match	78.23%	90.26%	83.82%	85.10%	93.48%	**89.09** %
IOBE	Exact match	71.35%	82.73%	76.62%	77.92%	85.58%	**81.57**%
	Left match	73.44%	85.14%	78.86%	79.69%	87.53%	**83.42**%
	Right match	77.81%	90.22%	83.56%	84.17%	92.45%	88.11%
IOBES	Exact match	71.77%	83.41%	77.16%	76.77%	84.23%	80.33%
	Left match	73.54%	85.47%	79.06%	78.44%	86.06%	82.07%
	Right match	78.33%	91.04%	**84.21**%	83.85%	92.00%	87.74%

Table 3.8: Results of NCBI dataset

SR scheme	Boundary	SVM			CRF		
		R	P	f1-measure	R	P	f1-measure
IO	Exact match	57.90%	73.30%	64.69%	67.04%	73.97%	70.33%
	Left match	67.51%	85.47%	**75.44**%	76.25%	84.14%	80.00%
	Right match	65.34%	82.73%	73.01%	74.80%	82.54%	78.48%
IOB2	Exact match	58.61%	74.21%	**65.49**%	68.49%	75.30%	71.73%
	Left match	67.48%	85.45%	75.41%	77.06%	84.72%	80.71%
	Right match	65.91%	83.46%	**73.66**%	75.46%	82.97%	79.04%
IOE2	Exact match	57.58%	73.98%	64.76%	69.04%	75.62%	72.18%
	Left match	66.06%	84.87%	74.29%	77.49%	84.87%	81.01%
	Right match	64.62%	83.02%	72.67%	76.09%	83.34%	**79.55**%
IOBE	Exact match	57.79%	73.33%	64.64%	69.38%	75.83%	72.46%
	Left match	67.27%	85.37%	75.25%	77.79%	85.02%	**81.24**%
	Right match	65.09%	82.60%	72.81%	75.92%	82.98%	79.29%
IOBES	Exact match	56.40%	72.60%	63.48%	69.09%	76.35%	**72.54**%
	Left match	65.69%	84.57%	73.95%	77.08%	85.17%	80.92%
	Right match	63.64%	81.93%	71.63%	75.46%	83.39%	79.23%

Table 3.9: Results of BC2GM dataset

SR scheme	Boundary	SVM			CRF		
		R	P	f1-measure	R	P	f1-measure
IO	Exact match	65.17%	80.78%	72.14%	69.64%	82.07%	75.35%
	Left match	67.02%	83.08%	**74.19**%	71.77%	84.58%	77.65%
	Right match	74.21%	91.99%	82.15%	78.64%	92.67%	85.08%
IOB2	Exact match	65.60%	81.04%	72.50%	70.57%	82.99%	76.28%
	Left match	66.98%	82.74%	74.03%	72.36%	85.09%	78.21%
	Right match	74.48%	92.01%	82.32%	78.73%	92.58%	85.10%
IOE2	Exact match	65.39%	80.29%	72.08%	70.75%	82.76%	76.29%
	Left match	66.89%	82.13%	73.73%	72.24%	84.51%	77.89%
	Right match	74.50%	91.48%	82.12%	79.32%	92.78%	**85.52**%
IOBE	Exact match	65.75%	81.21%	**72.67**%	70.61%	83.15%	7637%
	Left match	67.04%	82.80%	74.09%	72.45%	85.31%	78.35%
	Right match	74.59%	92.13%	**82.44**%	78.32%	92.23%	84.71%
IOBES	Exact match	65.57%	80.40%	72.24%	71.11%	83.78%	**76.93**%
	Left match	67.09%	82.26%	73.90%	72.51%	85.43%	**78.44**%
	Right match	74.59%	91.46%	82.17%	78.48%	92.46%	84.90%

Table 3.10: Results of BC5CDR dataset

3.5 Chapter Summary

In this chapter, the effect of different SR schemes on the performance of BioNER is studied. SVM and CRF have been used to build BioNER systems for this study. The results obtained illustrate that IO models reported low performance for all datasets, as it contains only two tags and it cannot differentiate between any of the boundaries. IOBES scheme outperforms other schemes for JNLPBA 2004 shared task, BC2GM and BC5CDR datasets. IOBE and IOE2 reported the highest f1-measure for NCBI and i2b2/VA 2010 shared task datasets respectively.

Chapter 4

Ensemble-Based Approach

Ensemble approach combines the output of base classifiers to get better performance from a pool of classifiers than an individual classifier and has achieved meaningful and encouraging results [83]. The generalization capability of ensemble classifier which is usually much better than that of base classifiers is the strength of ensemble classifier. It has been applied for different NLP tasks such as BioNER [75, 45], word segmentation [84], word sense disambiguation [85, 86] and PoS tagging [84, 87, 88]. In this chapter, we present an ensemble based system for BioNER using two different classification algorithms, CRF and SVM with IO, IOB2, IOE2, IOBE and IOBES SR schemes. An ensemble-based system developed for Native Language Identification (NLI) has been introduced in the second part of this chapter. NLI is an important NLP task that got attention recently by research community. As we wanted to study the impact of ensemble approach on other NLP task, we participated on a NLI task for Indian languages held in conjunction with FIRE2017 [89], IISc, Bangalore.

Some parts of the material of this chapter have appeared in the following research paper:

1. Hamada A. Nayel and H. L. Shashirekha: "Mangalore-University@INLI-FIRE-2017: Indian Native Language Identification using Support Vector Machines and Ensemble Approach", In Working notes of Indian Native Language Identification (INLI) task held in conjunction with Forum of Information Retrieval Evaluation (FIRE-2017), IISc, Bangalore, India, 8th - 10th December.
2. Hamada A. Nayel and H. L. Shashirekha: "Improving NER for Clinical Texts by Ensemble Approach using Segment Representations", In the Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017), pages 197-204, NLP Association of India, December-2017, Kolkata, India.

4.1 Preamble

Ensemble learning is a classification technique, which uses a set of different heterogenous and diverse classifiers as base classifiers and combines the output of them in different approaches to get the final result [42]. This technique which is gaining popularity tries to overcome the weakness of some classifiers using the strength of other classifiers.

Generalization is a property of a classifier, which characterizes how correctly the result can be applied to unseen data. The advantage of using ensemble classification is the generalization ability of an ensemble which is usually much stronger than that of a single classifier. Dietterich [90] explores how training data and classifier may affect the generalization of an ensemble classification. Experimentations showed that ensembles for classification problem are often much more accurate than the individual base classifiers that make them up [91, 92, 93], and different theoretical clarification have been proposed to rationalize the effectiveness of some ensemble methods [94, 95, 96].

The general framework of ensemble learning is shown in Figure 4.1. Two stages are required while building an ensemble based system; building base classifiers and combining the output of these base classifiers. Base classifiers can be built using homogenous or heterogenous classifiers. Homogeneous classifiers use different distributions of the training set but similar classifiers to build the base classifiers as in bagging and boosting. On the other hand, heterogenous classifiers use full training set but different classifiers to build the base classifiers. Combining outputs of base classifiers can be done using majority voting, weighted voting or Staking. Stacking uses outputs of base classifiers as inputs to train a classifier, which will be used for final output and is known as *meta-classifier* [43].

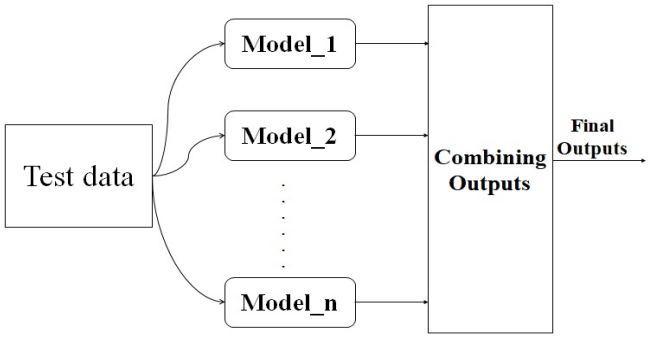


Figure 4.1: Framework of Ensemble Approach

4.2 Ensemble Based framework for BioNER

In this section, a two-stage ensemble algorithm for BioNER is proposed. In the first stage, SVMs and CRFs are used to create ten base classifiers with different SR models namely IO, IOB2, IOE2, IOBE and IOBES. Stacking using CRF as a meta-classifier, majority voting and weighted voting have been used separately to combine the results of base classifiers in the second stage. Up to our knowledge, this is the first work that uses SR models to achieve diversity of base classifiers.

4.2.1 Related Work

The research works carried out in ensemble approach uses different training data sets or different learning algorithms to create the base classifiers. An ensemble based system using majority voting and stacking has been designed for clinical concept extraction by Youngjun Kim et al. [97]. Four types of information extraction models have been used as base models: rule-based model, knowledge based system using MetaMap[1], SVM and CRF. They used majority voting and stacking for combining the outputs of base models. Zhou et al. [98] developed an ensemble based system for BioNER in which the base classifiers are SVM-based classifier and two others based on discriminative Hidden Markov Model (DHMM). They used majority voting technique to combine the outputs of the base classifiers. At final stage, three post-processing modules: an abbreviation resolution module, a gene name refinement module and a dictionary matching module are incorporated into the system for further improvement. GENETAG dataset[2] [99] was used to evaluate the approach and reported a 82.58% f1-measure. Bhasuran et al. [45], proposed a stacked ensemble approach for recognizing disease names. Backward CRF and forward CRF have been used as base classifiers. A fuzzy string matching algorithm was integrated with the system for detecting rare disease mentions. NCBI and BC5CDR datasets have been used to evaluate the system and they reported that the stacked ensemble approach outperforms the base classifiers. As they used the same classification algorithm with two variations to create the base classifiers, the classifiers diversity may not be achieved. Kang et al. [100] proposed an ensemble based system for NER in clinical records. They used a simple voting scheme to combine the outputs of two dictionary-based tools (MetaMap[3] and Peregrine) and five

[1] https://metamap.nlm.nih.gov/

[2] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENETAG.tar.gz

[3] https://metamap.nlm.nih.gov/

statistical-based tools (ABNER[4] [101], Lingpipe[5], OpenNLP Chunker[6], JNET [102] and StanfordNER[7]) and the combined system outperforms the best base classifier. Using i2b2 shared task dataset for system evaluation they have reported an f1-measure of 82.0%. It is reported that when excluding MetaMap tool[8] from the base classifiers the performance of proposed system is improved by 0.2%. Ekbal and Saha [75] used stacked ensemble approach to extract BioNEs. They used a genetic algorithm for feature selection in order to use CRF and SVM for learning the base classifiers and have evaluated their method on JNLPBA 2004 shared task dataset and GENETAG dataset, and reported 75.17% and 94.70% f1-measure respectively.

4.2.2 Methodology

The proposed system consists of two phases, the first phase is building the base classifiers and second phase is combining the outputs of these classifiers. Figure 4.2 shows the framework of first phase, where training data is used to learn the base classifiers. In this phase feature extraction has been done as mentioned in Section 3.3.1. A SR converter module is designed to convert the dataset which is available in IOB2 scheme into IO, IOE2, IOBE and IOBES schemes. SVM and CRF classification algorithms have been used to learn the base classifiers using different SR schemes.

Figures 4.3 and 4.4 show the framework of second phase using majority voting and stacking respectively. In this phase, SR converter module is used to convert the outputs of the base classifiers into IOB2 scheme again for combining purpose. In majority voting, we used both simple majority voting and weighted majority voting. For stacking, we used CRF to learn the meta-classifier.

4.2.3 Experiments and Results

The proposed method combines the outputs of base-classifiers using two different approaches namely majority voting and stacked generalization and is evaluated on JNLPBA 2004 shared task, NCBI, BC2GM, BC5CDR and i2b2 2010 shared task benchmark datasets (described in Section 2.5). For each dataset, there are five CRF-based base classifiers with

[4]http://pages.cs.wisc.edu/ bsettles/abner/
[5]http://alias-i.com/lingpipe/
[6]http://opennlp.apache.org/
[7]https://nlp.stanford.edu/software/CRF-NER.shtml
[8]https://metamap.nlm.nih.gov/

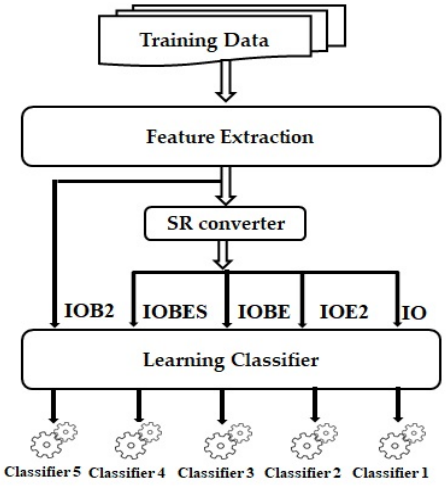


Figure 4.2: Learning base classifiers

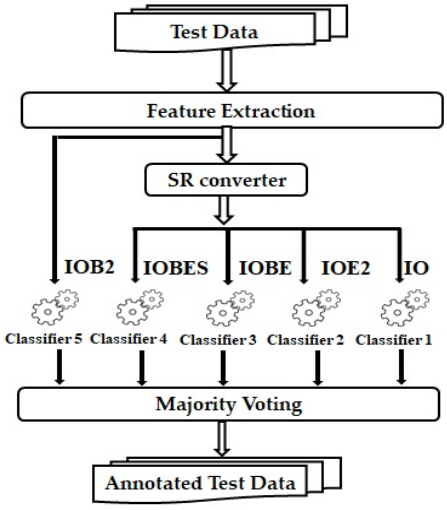


Figure 4.3: Combining base classifiers using majority voting

different SR schemes and five SVM-based base classifiers with different SR schemes. The setting and description of creating these base classifiers are given in Section 3.4.1.

In majority voting, the output of all base classifiers are combined together and the output of final system is decided based on majority voting. If majority voting fails then the highest performance output of the base classifiers is considered on final output. Another variation of majority voting is *weighted majority voting*. In weighted majority voting each

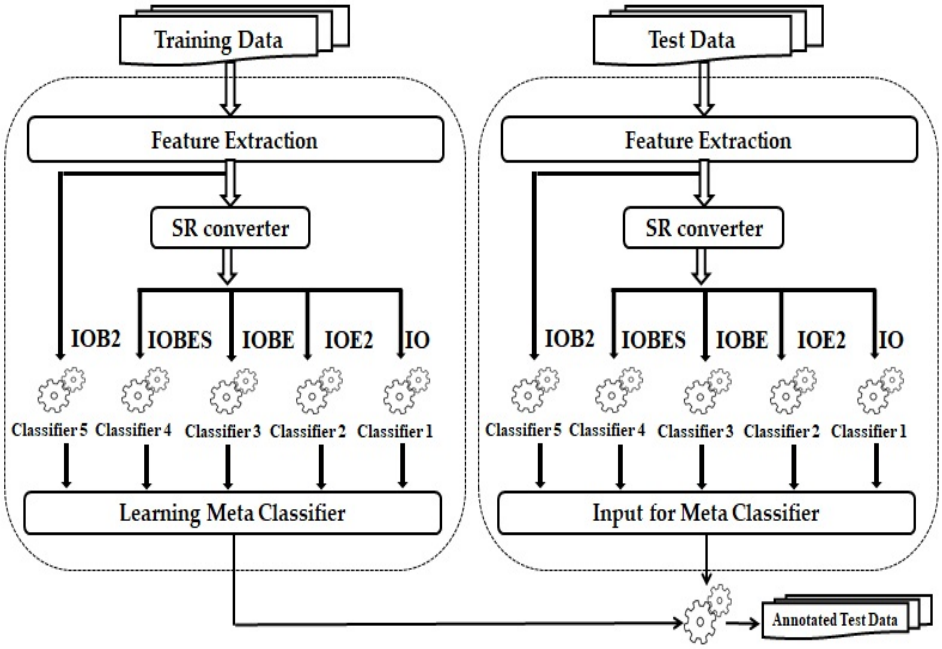


Figure 4.4: Combining base classifiers using stacking

base classifier is assigned a weight while calculating the majority voting. We have used the f1-measure of each base classifier as the weight of the corresponding classifier. Majority voting and weighted majority voting are implemented using R. An open source implementation of CRF, CRF++ package[9], has been used for constructing a CRF-based meta classifier for stacking.

The results of base classifiers and ensemble classifiers using majority voting (MV), weighted majority voting (weighted MV) and Stacking are shown in Table 4.1 in terms of f1-measure. The results show that ensemble using stacking outperforms ensemble using majority voting and weighted majority voting for all datasets. Ensemble using stacking approach reported the best f1-measure for NCBI, BC2GM and BC5CDR datasets. For JNLPBA and i2b2 datasets ensemble approach decreases the f1-measure by 0.06 and 0.56 respectively. Ensemble using weighted majority voting outperforms majority voting because the weights assigned to base classifiers effects positively according to the classifier performance. The detailed results in terms of P, R and f1-measure for exact matching, right match and left match for JNLPBA, i2b2, NCBI and BC2GM datasets are given in Table 4.2, and for BC5CDR dataset are given in Table 4.3.

[9]https://taku910.github.io/crfpp/

Classifiers	SR Model	Dataset				
		JNLPBA	**i2b2**	**NCBI**	**BC2GM**	**BC5CDR**
SVM Base Classifiers	IO	67.27%	72.85%	75.73%	64.69%	72.14%
	IOB2	67.54%	74.52%	77.11%	65.49%	72.50%
	IOE2	66.30%	74.09%	77.23%	64.76%	72.08%
	IOBE	66.73%	73.98%	76.62%	64.64%	72.67%
	IOBES	66.13%	72.91%	77.16%	63.48%	72.24%
CRF Base Classifiers	IO	70.84%	78.78%	80.13%	70.33%	75.35%
	IOB2	70.92%	79.49%	81.44%	71.73%	76.28%
	IOE2	71.50%	**80.55**%	81.46%	72.18%	76.29%
	IOBE	71.64%	79.60%	81.57%	72.46%	76.37%
	IOBES	**71.98**%	79.69%	80.33%	72.54%	76.93%
Ensemble Classifiers	MV	70.17%	69.59%	80.53%	72.16%	75.99%
	Weighted MV	71.88%	79.85%	81.65%	73.12%	76.85%
	Stacking	71.92%	79.89%	**82.08**%	**73.21**%	**77.03**%

Table 4.1: Results of base and ensemble classifiers

Dataset	Method	Boundary	SVM		
			R	P	f1-measure
JNLPBA	MV	Exact match	72.59%	67.91%	70.17%
		Left match	77.36%	72.37%	74.78%
		Right match	80.99%	75.76%	78.29%
	Weighted MV	Exact match	72.97%	70.82%	71.88%
		Left match	76.58%	74.32%	75.43%
		Right match	80.76%	78.38%	79.55%
	Stacking	Exact match	73.03%	70.85%	**71.92**%
		Left match	76.65%	74.35%	75.48%
		Right match	80.80%	78.39%	79.57%
i2b2	MV	Exact match	70.80%	68.41%	69.59%
		Left match	77.27%	74.67%	75.95%
		Right match	76.80%	74.21%	75.49%
	Weighted MV	Exact match	75.33%	84.94%	79.85%
		Left match	78.54%	88.56%	83.25%
		Right match	79.58%	89.73%	84.35%
	Stacking	Exact match	75.38%	84.97%	**79.89**%
		Left match	78.58%	88.58%	83.28%
		Right match	79.61%	89.74%	84.37%
NCBI	MV	Exact match	78.44%	82.75%	80.53%
		Left match	80.94%	85.38%	83.10%
		Right match	85.62%	90.33%	87.91%
	Weighted MV	Exact match	77.40%	86.40%	81.65%
		Left match	78.85%	88.02%	83.19%
		Right match	84.27%	94.07%	88.9%0
	Stacking	Exact match	78.02%	86.59%	**82.08**%
		Left match	79.48%	88.21%	83.62%
		Right match	84.79%	94.10%	89.21%
BC2GM	MV	Exact match	69.91%	74.56%	72.16%
		Left match	79.30%	84.57%	81.85%
		Right match	77.20%	82.33%	79.68%
	Weighted MV	Exact match	68.96%	77.81%	73.12%
		Left match	76.81%	86.66%	81.43%
		Right match	75.83%	85.55%	80.40%
	Stacking	Exact match	69.06%	77.90%	**73.21**%
		Left match	76.90%	86.75%	81.53%
		Right match	75.87%	85.59%	80.44%

Table 4.2: Results of JNLPBA, i2b2, NCBI and BC2GM datasets

Dataset	Method	Boundary	SVM		
			R	P	f1-measure
BC5CDR	MV	Exact match	72.72%	79.57%	75.99%
		Left match	75.47%	82.59%	78.87%
		Right match	81.56%	89.24%	85.22%
	Weighted MV	Exact match	70.80%	84.04%	76.85%
		Left match	72.20%	85.70%	78.37%
		Right match	78.84%	93.59%	85.58%
	Stacking	Exact match	71.02%	84.15%	**77.03**%
		Left match	72.42%	85.81%	78.55%
		Right match	79.00%	93.60%	85.68%

Table 4.3: Results of BC5CDR dataset

4.3 Native Language Identification (NLI)

Native Language Identification (NLI) aims at identifying the native language (L1) of users written in another or later learned language or speech (L2). The general framework of NLI is given in Figure 4.5. NLI is an important task that has many applications in different areas such as social-media analysis, authorship identification, second language acquisition and forensic investigation. In forensic analysis [103], NLI helps to glean information about the discriminant L1 cues in an anonymous text. Second Language Acquisition (SLA) [104] studies the transfer effects from the native languages on later learned language. In education, automatic correction of grammatical errors is an important application of NLI [105]. NLI can be used as a feature in authorship identification task [106] which aims at assigning a text to one of the predefined list of authors. Authorship identification is used in terrorists communications investigation [107] and digital crime investigation [108].

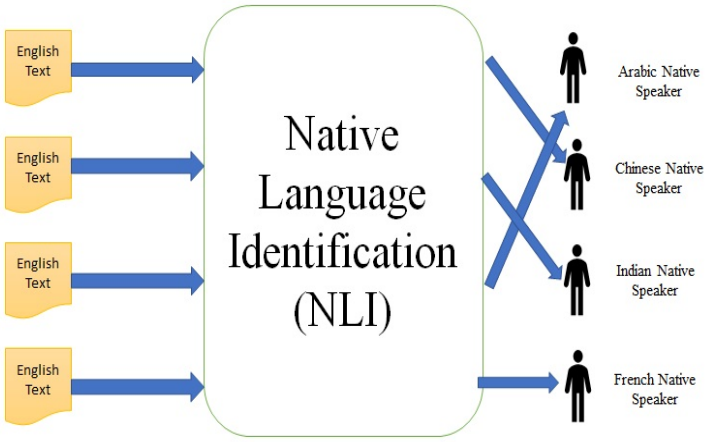


Figure 4.5: General framework of NLI task

4.3.1 Related Work

Supervised ML algorithms have been used for NLI by many researchers. Jarvis et al. [109], used SVM to create a model for NLI and reported an accuracy of 83.6%. Features such as N-grams of words, PoS tags and lemmas have been used to create feature space model for training the classifier. They tested the performance of their system using TOEFL11 dataset [110]. The TOEFL11 is a collection of 11,000 essays written by non-native English speak-

ers as part of a high-stakes test of general proficiency in academic English. The essays were written by learners from the following 11 L1 backgrounds: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. In this work, the feature set was not sufficient to cover the characteristics of different languages. Tetreault et al. [111] used ensemble approach to build a classifier to improve the performance of base classifiers. A wide range of features have been used to build an ensemble of logistic regression learners. Such features include word and character *n*-gram, PoS, function words, writing quality markers and spelling errors. In addition, a set of syntactic features such as Tree Substitution Grammars and dependency features extracted using the Stanford parser[10] has been used. The system has been evaluated using TOEFL11 and International Corpus of Learner English (ICLE). ICLE is a collection of 6,085 essays written by university undergraduates of advanced proficiency. The essays were written by learners from the following 11 L1 backgrounds: Bulgarian, Chinese, Czech, French, Japanese, Russian and Spanish. The system resulted state of the art accuracies of 90.1% and 80.9% on the ICLE and Toefl11 datasets respectively.

Malmasi et al. [112] used function words, PoS N-grams and a hybrid PoS/function word mixture N-gram model to train a SVM classifier for NLI task using data from learners of Norwegian language. Their system achieves an accuracy of 79% against a baseline of 13% for predicting L1 authors. The system has been applied to corpus of only one language, while most real world applications consists of more than one language. Wong et al. [113] investigated the use of Latent Dirichlet Analysis-based topic modelling to produce a feature set with lower dimensionality by clustering similar features. The proposed system did not outperform the baseline systems as the authors reported that Latent Dirichlet Analysis-induced classification models perform worse than the full feature-based models.

4.3.2 Task Description

India is a multilingual country with 22 languages mentioned in the 8th schedule of the constitution. According to article 343(1) of the Indian constitution "The official language of the Union shall be Hindi in Devanagari script". While Hindi is the most commonly used language in Northern India, the languages, Kannada, Tamil, Telugu and Malayalam are used in southern part of India. However, English is used as an official language throughout the country and it is the most commonly spoken language in India and certainly the most

[10]http://nlp.stanford.edu:8080/parser/

read and written language. The number of speakers of English as the second language increases frequently and this has also contributed to its rich variation. English is mixed with most of the Indian languages and is used as a second language frequently. Regional and educational differentiation distinguish the language usage and shows the stylistic diversity in English. But, identifying the native language of the users based on the context in English is a challenging task. In this era, social media is overwhelming our lives. Majority of people are communicating and discussing different topics using different platforms of social media such as Facebook, Twitter and Google+. While communicating with each other Indians prefer to use English because their native languages are different. In addition, most software and keyboards do not support input using Indian languages characters. So, people are using English keyboard to write words as transliterated words.

The task is to identify the native language of the writer from the given Text/XML file which contains a set of Facebook comments in English language. Six Indian languages - Tamil, Hindi, Kannada, Malayalam, Bengali and Telugu are considered for this shared task. This task was held in conjunction with Forum of Information Retrieval Evaluation (FIRE-2017), at Indian Institute of Science (IISc), Bangalore [89].

4.3.3 Task Formulation

Given a comment $I_j = <w_1, w_2, \ldots, w_N>$ where each w_i, $i = 1 \ldots N$ is either an English language word or a word of native language written in English (or transliterated to English language) for an individual social media user, the objective of the task is to identify the native language of the user. The comments may include English words in addition to the words of any one native language written in English. Considering the languages as a set of classes $C = \{Tamil, Hindi, Kannada, Malayalam, Bengali, Telugu\}$ and comments as individual instances $I = \{I_1, I_2, \ldots, I_n\}$ the task has been formulated as a classification problem that assigns one of the six predefined classes of C to a new unlabelled instance I_u.

4.3.4 Methodology

The three systems proposed for Indian Native Language Identification (**INLI**) [114] task submissions are described in this section. The general frame work of classifiers for INLI is shown in Figure 4.6. In preprocessing phase (also known as corpus cleaning) the inputs are tokenized and non-informative tokens and phrases are excluded followed by constructing

vector space model. These two steps are common for the proposed systems while the next step is creating a learning model using a ML algorithm. SVM, ensemble-based classifier using SVM, Multinomial Naïve Bayes and Random Forests Trees as base classifiers and Multinomial Naïve Bayes classifier are used for classification purpose for first, second and third submission respectively.

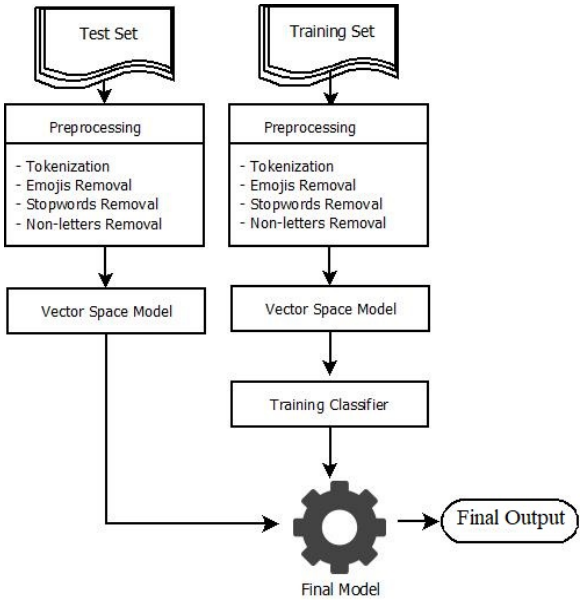


Figure 4.6: Framework of classifier

Pre-processing

In this phase, each comment I_j is tokenized into a set of words and uninformative tokens are removed to obtain bag of tokens as follows:

- ***Emoji removal***

  Emoji is a small image used as a visual presentation to express emotion. The first step in removing unrelated information is to remove Emojis automatically as they are not important for the identification of native language.

- ***Special characters and digits***

  Digits and special characters such as #, %, ... are the characters which appear frequently in the comments of all the languages. As such these characters do not contribute to the identification of native language, hence they are removed automatically.

- ***Modified stop words***

  Stop words are the words which appear frequently and do not contribute to the identification of native language. Hence, to remove stop words a union of different stop words lists, namely, i) Stop words list extracted from `nltk.corpus`[11] package, ii) Stop words list extracted from `stop_words`[12] package and iii) Manually compiled list of stop words which are commonly used but not mentioned in previous lists. (The complete list of manually written stop words is given in Appendix A)

Constructing Vector Space Model

After preprocessing, the informative tokens in the comments will be represented as vector space model. If $<t_1, t_2, \ldots, t_k>$ are the unique tokens/terms in a comment I_j, the vector space model for the comment I_j will be represented as $<w_{j1}, w_{j2}, \ldots, w_{jk}>$ where w_{ji} is the weight of the token/term t_i in comment I_j. For term weights, Term Frequency/Inverse Document Frequency (TF/IDF) is calculated as follows:-

$$t_j = tf_j * \log\left(\frac{N+1}{df_j+1}\right)$$

where tf_j is the total number of occurrences of term t_j in the current comment, df_j is the number of comments in which the token/term t_j occurs and N is the total number of comments.

Model Construction using SVM classifier

An SVM classifier is constructed as per the framework shown in Figure 4.6 to classify an instance into one of the six classes Tamil, Kannada, Malayalam, Bengali, and Telugu. SVM model as described in Chapter 3, can be formulated as an optimization problem as follows:

$$\begin{aligned} \text{Minimize:} \quad & Q(w) = \frac{1}{2} \parallel w \parallel^2 \\ \text{Subject to:} \quad & y_i(w \cdot x_i + b) \geq 1, \ \ \forall (x_i, y_i) \in D \end{aligned}$$

To solve this problem, we used Stochastic Gradient Descent (SGD) - an algorithm for solving optimization problems. SGD is one of the simplest and most popular linear methods to solve the optimization problems. In this task, the vector space model is represented as

[11] www.nltk.org/nltk_data/

[12] pypi.python.org/pypi/stop-words

continuous real-valued vectors (TF/IDF) and it is suitable to use SGD. The typical gradient descent algorithm updates the parameter w of the objective function $Q(w)$ as follow:

$$w = w - \eta \nabla_w E\left[Q(w)\right]$$

where η is step size and $E[Q(w)]$ is the expectation value of the function $Q(w)$ over full training data.

Instead of using full training data, SGD updates the parameter w using a few training examples. The new value of the parameter w is given by,

$$w = w - \eta \nabla_w \left(Q(w); x^{(i)}, y^{(i)}\right)$$

where $(x^{(i)}, y^{(i)}) \in B$ and $B \subset D$.

Model Construction for Multinomial Naïve Bayes classifier

In simple Naïve Bayes classification model the most probable native language c of the writer is calculated as follows:

$$c = \textbf{argmax}_{c \in C} P(c|w_1, w_2, \ldots, w_k)$$

where C is the set of available languages, and $P(c)$ is calculated by the frequency of c in the training data as follows:

$$P(c) = \frac{\text{No of comments written by user with native language } c}{\text{Total number of comments}}$$

and

$$P(c|w_1, w_2, \ldots, w_k) = \frac{P(w_1, w_2, \ldots, w_k|c)P(c)}{P(w_1, w_2, \ldots, w_k)}$$

where $P(w_1, w_2, \ldots, w_k) \neq 0$.
With assumption of uniformity of $(w_1, w_2, \ldots, w_k)$

$$P(c|w_1, w_2, \ldots, w_k) = P(w_1, w_2, \ldots, w_k|c)P(c)$$

Applying chain rule we have,

$$P(w_1, w_2, \ldots, w_k|c)P(c) = P(c)\prod_{i=1}^{k} P(w_i|c)$$

Then the problem becomes:

$$c = \textbf{argmax}_{c \in C} P(c)\prod_{i=1}^{k} P(w_i|c) \tag{4.1}$$

In Naïve Bayes model [115], a comment is assumed to be a set of features $\{w_1, w_2, \ldots, w_k\}$, while in multinomial Naïve Bayes model the comment is assumed to be a feature vector $<w_1, w_2, \ldots, w_k>$ and the value of $P(w_i|c)$ in Equation 4.1 is calculated as follows [116]:

$$P(w_i|c) = \frac{1 + w_{ic}}{k + k_c} \tag{4.2}$$

where

w_{ic}	the frequency of feature w_i in the comments written by natives of the language c
k	total number of features
k_c	total number of features in comments written by users with native language c

Model Construction for Random Forest Tree classifier

A Decision Tree (DT) is a supervised algorithm having a tree structure with internal nodes representing conditions or decisions and leaf nodes representing the class labels. DT is used for classification of large datasets and frequent pattern mining. An example of DT for name gender identification is shown in Figure 4.7.

Random forest classifier comprises of multiple decision trees and each tree depends on independently sampled random vector [117]. Majority voting is used to predict the class based on the output of decision trees.

Model Construction for Ensemble classifier

The framework of ensemble learning is shown in Figure 4.1. Multinomial Bayes, SVM and random forest tree classifiers are used as base classifiers and the results of these classifiers are combined by weighted voting. The base classifiers are designed as per the framework shown in Figure 4.6.

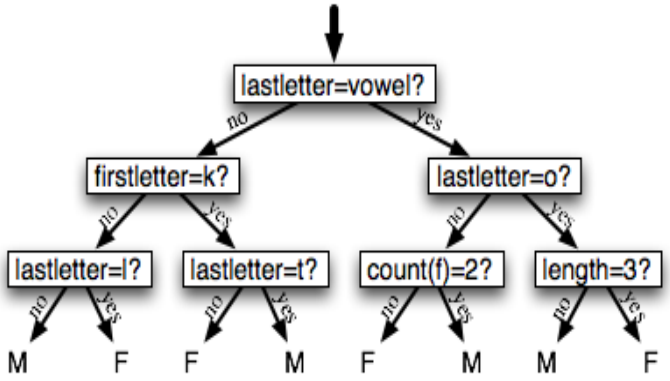


Figure 4.7: Example of Decision Tree for name gender identification

4.3.5 Dataset

The data sets provided by the organizers of INLI task are a collection of comments from different regional newspaper's facebook pages during April-2017 to July-2017. Training and test sets contain 1233 and 783 comments respectively. Table 4.4 shows a brief statistics about the training set.

Language	**# of comments**	**Ratio**
Tamil	207	16.79%
Hindi	211	17.11%
Kannada	203	16.46%
Malayalam	200	16.22%
Bengali	202	16.38%
Telugu	210	17.03%
Total	1233	100.00%

Table 4.4: Training set statistics

4.3.6 Performance Evaluation

Performance of INLI task is measured as the accuracy of a system in addition to class-wise accuracy which is calculated using Precision (P), Recall (R) and f1-measure[13]. For each class, P is the measure of the number of comments correctly classified over the total number of comments that the system classified as same class. R is the measure of the number of comments correctly classified over the actual number of comments of the class. F1-measure is the harmonic mean of P and R, which is calculated as follows:

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

4.3.7 Results and Discussion

Three systems were submitted to INLI [114]. Overall accuracy of SVM based submission is 47.60% and it ranks second among 27 systems submitted by 12 teams as shown in Table 4.5. Overall accuracy of Ensemble and Multinomial Naïve Bayes based submissions are 47.30% and 45.20% respectively, and they rank third and seventh among all the submissions respectively. Classwise accuracies in terms of P, R and f1-measure are shown in Table 4.6, Table4.7 and Table 4.8 for SVM based submission, Ensemble based submission and Multinomial Naïve Bayes submission respectively. The accuracies of all three submissions (using 10-fold cross-validation for all the classifiers) are given in Table 4.9. Results of all submissions illustrate that the performance of identifying Hindi is the worst. The reason may be most of the other languages' natives have knowledge of Hindi and systems depend essentially on the effective words for each language.

[13] http://www.nltk.org/_modules/nltk/metrics/scores.html

Rank	Team	Submission No	Accuracy	Organization
1	DalTeam	1	48.80%	Dalhousie University
2	MANGALORE	2	47.60%	Mangalore University
3	MANGALORE	1	47.30%	Mangalore University
4	SEERNET	3	46.90%	Seernet Technologies LLC
5	SEERNET	1	46.60%	Seernet Technologies LLC
6	SEERNET	2	46.40%	Seernet Technologies LLC
7	MANGALORE	3	45.20%	Mangalore University
8	Bharathi_SSN	1	43.60%	SSN College of Engineering (Bharathi)
9	BASELINE(TFIDF-SVM)	1	43.00%	Task Organizer
10	SSN_NLP_new	1	38.80%	SSN College of Engineering (Thenmozhi)
11	ClassyPy	1	38.20%	Erasmus University of Rotterdam
12	Anuj	1	38.20%	ANUJ
13	ClassyPy	2	37.90%	NIT-W, IDRBT-HYD
14	IDRBT	1	37.20%	IIT Hyderabad
15	Anuj	2	36.80%	ANUJ
16	DIG	1	36.70%	CEC, Kerala
17	DIG	2	36.70%	CEC, Kerala
18	team_CEC	1	32.20%	BMS, Bangalore
19	ClassyPy	3	28.90%	NIT-W, IDRBT-HYD
20	Bits_Pilani	2	28.00%	BITS
21	BMSCE_ISE	2	27.80%	JU
22	Bits_Pilani	1	26.90%	BITS
23	Bits_Pilani	3	26.70%	BITS
24	Anuj	3	25.50%	ANUJ
25	DIG	3	22.30%	CEC, Kerala
26	BMSCE_ISE	1	22.20%	JU
27	JUNLP	1	17.80%	JUNLP

Table 4.5: Overall accuracy of all teams participated in INLI

Class	P	R	f1-measure
Bengali	54.00%	**84.90**%	**66.00**%
Hindi	**60.00**%	7.20%	12.80%
Kannada	40.40%	54.10%	46.20%
Malayalam	42.70%	66.30%	51.90%
Tamil	58.00%	58.00%	58.00%
Telugu	32.50%	48.10%	38.80%
Overall Accuracy		47.60%	

Table 4.6: Class wise accuracy of SVM-based submission

Class P	R	F1-measure	
Bengali	56.50%	**79.50**%	**66.10**%
Hindi	**60.70**%	6.80%	12.20%
Kannada	38.40%	58.10%	46.20%
Malayalam	40.40%	70.70%	51.40%
Tamil	58.00%	58.00%	58.00%
Telugu	32.80%	49.40%	39.40%
Overall Accuracy		47.30%	

Table 4.7: Class wise accuracy of Ensemble-based submission

Class	P	R	f1-measure
Bengali	59.20%	**78.40**%	**67.40**%
Hindi	**66.70**%	4.80%	8.90%
Kannada	34.80%	64.90%	45.30%
Malayalam	32.60%	78.30%	46.00%
Tamil	54.40%	49.00%	51.60%
Telugu	39.40%	34.60%	36.80%
Overall Accuracy		45.20%	

Table 4.8: Class wise accuracy of Multinomial NB-based submission

Fold No.	Submission 1	Submission 2	Submission 2
1	88.09%	87.30%	87.30%
2	84.80%	84.80%	85.60%
3	90.32%	90.32%	88.71%
4	91.06%	91.06%	87.80%
5	89.43%	86.18%	86.18%
6	79.68%	80.49%	81.30%
7	86.18%	90.24%	86.18%
8	88.52%	89.34%	90.16%
9	90.98%	90.16%	90.16%
10	89.34%	91.80%	90.16%
Mean	87.84%	88.17%	87.36%
STD	3.32	3.33	2.61

Table 4.9: 10-fold cross-validation accuracy for all submissions

4.4 Chapter Summary

Ensemble approach is an effective learning methodology, as it tries to overcome the weakness of one classifier by the strength of another. In this chapter, Ensemble approach has been used for BioNER and INLI. For BioNER, three models - majority voting, weighted majority voting and stacking have been used separately to combine the output of base classifiers. Weighted majority voting has been used to combine the output of base classifiers of INLI. The performance of ensemble classifier of BioNER outperforms the performance of each of the base classifiers, while in INLI the performance of ensemble classifier is very close to the best base classifier and outperforms the classifiers submitted by other teams.

Chapter 5

A Deep Learning Model for Disease-NER

In recent years, Deep Learning (DL) models are becoming important due to their demonstrated success at overcoming complex learning problems. DL models have been applied effectively for different NLP tasks such as PoS tagging and MT. In this chapter, a DL model for Disease Named Entity Recognition (Disease-NER) using dictionary information is proposed and evaluated on National Center for Biotechnology Information (NCBI) disease corpus and BC5CDR dataset. Word embeddings trained over general domain texts as well as biomedical texts have been used to represent input to the proposed model. This study also compares two different SR schemes, namely IOB2 and IOBES for Disease-NER. The results illustrate that using dictionary information, pre-trained word embeddings, character embeddings and CRF with global score improves the performance of BioNER system.

5.1 Preamble

Disease is a principle BioNE, which has got attention by biomedical research due to increase in research in health and the impact of disease on public life. Disease-NER is a challenging problem due to the general challenges of BioNER listed in Section 2.2. Further, abbreviations which are used frequently in biomedical literature are the main sources of ambiguity. For example, "*AS*" may refer to "*Asperger Syndrome*" or "*Autism Spectrum*" or "*Aortic Stenosis*" or "*Ankylosing Spondylitis*" as well as "*Angleman Syndrome*". In such cases to which entity an abbreviation refers to has to be resolved depending on the context. Figure 5.1 shows an example of an abstract with disease mentions highlighted.

Identification of APC2, a homologue of the adenomatous polyposis coli tumour suppressor. The adenomatous polyposis coli (APC) tumour-suppressor protein controls the Wnt signalling pathway by forming a complex with glycogen synthase kinase 3beta (GSK-3beta), axin/conductin and betacatenin. Complex formation induces the rapid degradation of betacatenin. In colon carcinoma cells, loss of APC leads to the accumulation of betacatenin in the nucleus, where it binds to and activates the Tcf-4 transcription factor. Here, we report the identification and genomic structure of APC homologues. Mammalian APC2, which closely resembles APC in overall domain structure, was functionally analyzed and shown to contain two SAMP domains, both of which are required for binding to conductin. Like APC, APC2 regulates the formation of active betacatenin-Tcf complexes, as demonstrated using transient transcriptional activation assays in APC -/- colon carcinoma cells. Human APC2 maps to chromosome 19p13.3. APC and APC2 may therefore have comparable functions in development and cancer.

Figure 5.1: An abstract with disease mentions highlighted

Different approaches have been used for Disease-NER, such as dictionary-based approach, rule-based approach, ML approach and hybrid approach as discussed in Section 2.4. Deep Learning (DL) is a big trend in ML, which promises powerful and fast ML algorithms moving closer to the performance of Artificial Intelligence (AI) systems. It is about learning multiple levels of representation and abstraction that help to make sense of any data such as images, sound, and text. Automatically learning features at multiple levels of abstraction allow a system to learn complex functions mapping the input to the output directly from data, without depending on human-crafted features. DL employs the multi-layer Artificial Neural Networks (ANN) for increasingly richer functionality. While the concept of classical ML is characterized as learning a model to make predictions based on past observations, DL approaches are characterized by learning to not only predict but also to correctly represent the data such that it is suitable for prediction. Figure 5.2 shows the difference between classical ML flow and DL flow.

Given a large set of desired input-output mapping, DL approaches work by feeding the data into an ANN that produces consecutive transformations of the input until a final transformation predicts the output. These transformations are learnt from the given input-output mappings, such that each transformation makes it easier to relate the data to the desired label.

Of late, DL algorithms based on ANNs [118, 17] are being used to a larger extent for various NLP tasks such as PoS tagging [119], multi-task learning [120], relation extraction for biomedical texts [121] and biomedical event extraction [122, 123].

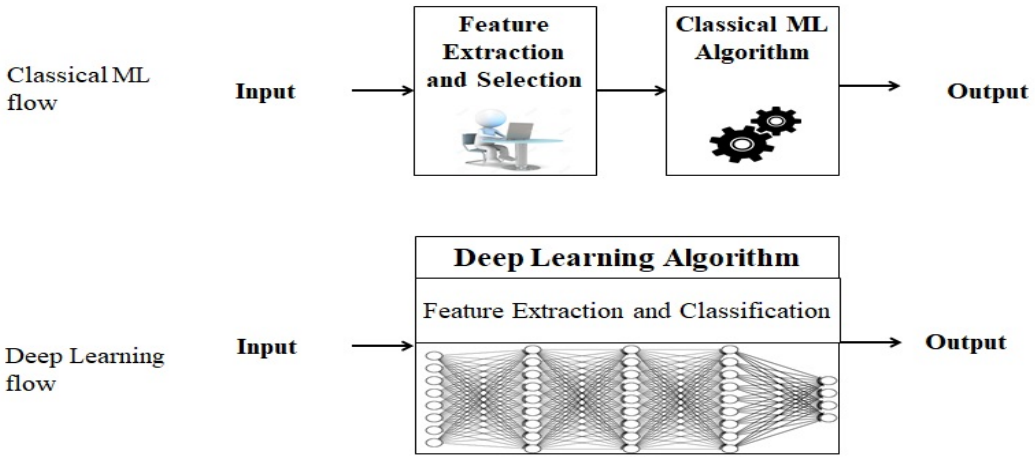


Figure 5.2: Classical and deep learning flows

In this chapter, an efficient DL model using ANN for Disease-NER has been developed. NCBI and BC5CDR datasets are used for evaluation of the proposed model and the results are reported in terms of f1-measure.

5.2 Background

5.2.1 Artificial Neural Networks (ANN)

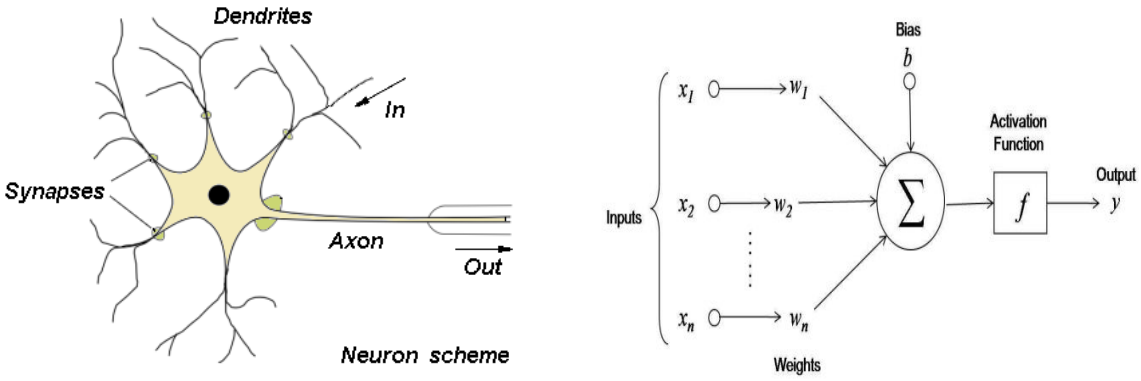
ANN are inspired by the mechanism of brain computation which consists of computational units called neurons. However, connections between ANN and the brain are in fact rather slim. In the metaphor, a neuron has scalar inputs with associated weights and outputs. The neuron multiplies each input by its weight, sums them and transforms to a working output through applying a non linear function called activation function. Figure 5.3 shows examples of some popular activation functions. The structure of a biological neuron and an artificial neuron model with n inputs and one output is shown in Figures 5.4(a), 5.4(b) respectively. In this example, a neuron receives simultaneous inputs $X = (x_1, x_2, \ldots, x_n)$ associated with weights $W = (w_1, w_2, \ldots, w_n)$, a bias b and calculates the output as:

$$y = f(W \cdot X + b) \tag{5.1}$$

where f is the activation function.

Function	Formula
Binary Step	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (sigmoid)	$f(x) = \frac{1}{1+e^{-x}}$
Tanh	$f(x) = \tanh x = \frac{2}{1-e^{-2x}} - 1$
ArcTanh	$f(x) = \tanh^{-1} x$
Rectified Linear Unit (ReLU)	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

Figure 5.3: Examples of activation functions



(a) Structure of a biological neuron

(b) A simple neuron model

Figure 5.4: A typical BioNER system with an example

ANN comprises of a large number of neurons within different layers with each layer having a specific task. An ANN model basically consists of three layers: an input layer, one or more hidden layers and an output layer. Input layer contains a set of neurons called input nodes, which receive raw inputs directly. The hidden layers receive the data from the input nodes and responsible for processing these data by calculating the weights of neurons at each layer. These weights are called connection weights and passed from one node to another. Number of nodes in hidden layers influences the number of connections as well as computational complexity. During learning connection weights are adjusted to be able to predict the correct class label of the input. Using multiple hidden layers helps in detecting more features while learning the model. Output layer receives the processed data and uses its activation function to generate final output. This kind of ANN where information flows

in one direction from input layer to output layer through one or more hidden layers is called feed-forward ANN. Figure 5.5 shows an example of a feed-forward ANN with two hidden layers. An ANN is called fully connected if each node in a layer is connected to all nodes in the subsequent layer.

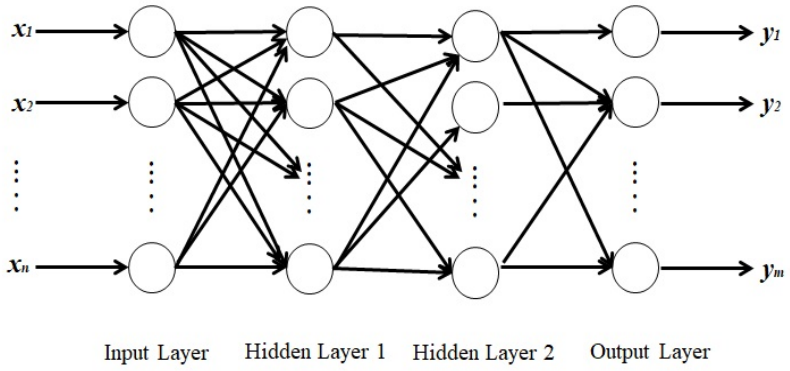


Figure 5.5: Structure of a simple feed-forward ANN

Recurrent Neural Networks (RNN) is a type of ANN in which hidden layer neurons has self-connections which means output depends not only on the present inputs but also on the previous neuron state. A simple form of RNN which contains an ANN with the previous set of hidden unit activations feeding back into the network along with the inputs is shown in Figure 5.6. The activations are updated at each time step t and a delay unit has been introduced to hold activations until they are processed at the next time step. The input vector x_0 at time stamp $t = 0$ processed using RNN structure is as follows:

$$h_t = f_W(h_{t-1}, x_t) \tag{5.2}$$

where, h_t is the output at time stamp t, h_{t-1} is the output at time stamp $t-1$, f_W is an activation function with parameter W and x_t is the input vector at the time stamp t.

In case of long sequences, RNNs are biased towards their most recent inputs in the sequence due to the gradient vanishing problem [124, 125]. While calculating weights in RNN for each time stamp t, the gradients of error get smaller and smaller as moving backward in the network and gradually vanish. Thus, the neuron in the earlier layers learns very slowly as compared to the neurons in the later layers. Earlier layers in the network are important as they are responsible to learn and detect patterns and are the building blocks of the RNN. Figure 5.7 shows an example of gradient vanishing problem. In this figure, the dark shade

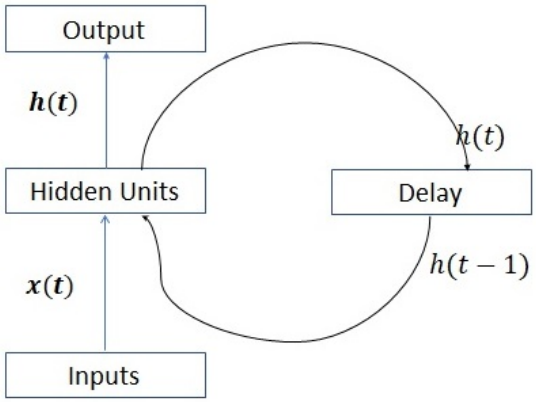


Figure 5.6: A Simple RNN Structure

of the node indicates the sensitivity over time of the network nodes to the input at first time stamp. The sensitivity decreases exponentially over time as new inputs overwrite the activation of hidden unit and the network forgets the input at first time stamp. To overcome the gradient vanishing problem, S. Hochreiter and J. Schmidhuber [126] introduced a new RNN architecture called Long Short-Term Memory (LSTM).

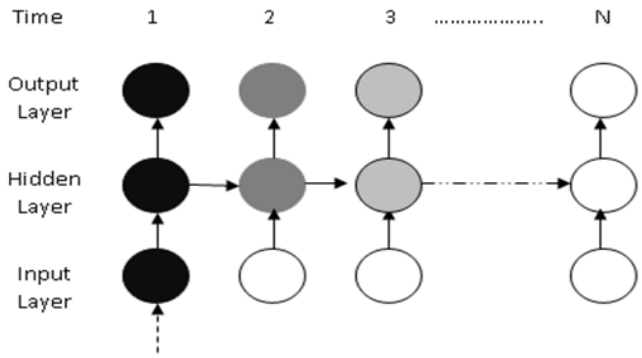


Figure 5.7: Gradient vanishing problem for RNN

Long Short-Term Memory

LSTM [126] is a kind of RNN which handles sequences of arbitrary length and is able to model dependencies between far apart sequence elements as well as consecutive elements. The LSTM architecture consists of a set of RNNs known as memory blocks. Each block contains self-connected memory cell (m_t) and three multiplicative units namely input (i_t),

output (o_t) and forget (f_t) gates, that provide continuous peers of write, read and reset operations for the cells as shown in Figure 5.8. These gates regulate the information in memory

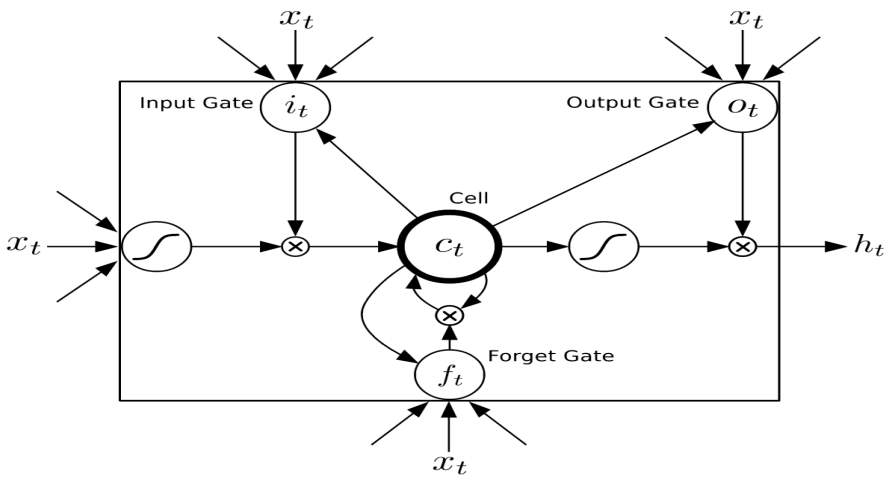


Figure 5.8: Structure of LSTM unit

cell and consists of a sigmoid function. The input gate regulates the proportion of history information that will be kept in memory cell and the output gate regulates the proportion of information stored in the memory cell which will influence other neurons. Forget gate can modify the memory cell by allowing the cell either to remember or forget its previous state. The complete details of LSTM architecture is described by S. Hochreiter and J. Schmidhuber [126].

In the sequence labeling problem, the typical input is a sequence of input vectors and the output is a sequence of output tags. NER can be considered as a sequence labeling problem where the sentence X consisting of words $(w_1, w_2, ..., w_n)$ is given as input and the required output is the sequence of tags $T = (t_1, t_2, ..., t_n)$ that represents the class labels of the words. LSTM is suitable to apply for NER as it can remember up to the first word in the sentence. However, one shortcoming of LSTM is that they process the input only in left context. But, in NER it is beneficial to have access to both left and right contexts as the output tag of a word depends on few previous and few next words (context window). This problem is overcome by a Bidirectional LSTM (BiLSTM) [127] where each sequence is presented in forward and backward direction to two separate hidden states to capture left and right context information respectively. Then the outputs of two hidden states are concatenated to form the final output as shown in Figure 5.9.

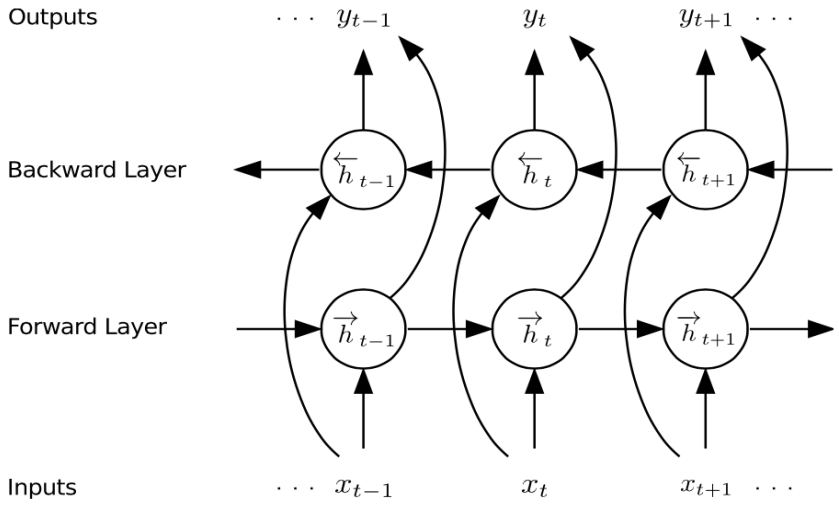


Figure 5.9: Structure of Bidirectional LSTM

To apply BiLSTM for NER, a sentence $X = (w_1, w_2, ..., w_n)$ has to be fed to a BiLSTM and each word has to be replaced by its vector representation. Applying BiLSTM structure on these representations outputs a vector $Y = (t_1, t_2, ..., t_n)$ representing the corresponding tags.

ANN in general can be applied for language understanding and NLP tasks such as speech recognition, MT and text summarization using the concept of language modeling (LM). LM is a probabilistic model that is able to predict the next word in the sequence given the words that precede it. The central component in ANN language modelling is the use of an embedding layer which maps discrete symbols (words, characters) to continuous vectors in a relatively low-dimensional space. Vector representation of words and documents has been a leading approach in information retrieval and computational semantics [128]. The primitive method of representing a document as a vector is the bag-of-words model. If m is the size of the vocabulary, a document is assigned a m-dimensional vector with non-zero entries for words corresponding to the entries occurred in the document and zero entries for rest of the words. These are used to learn n-dimensional real-valued vector representations of words in $\mathbb{R}^n$. After the rise in popularity of ANNs, a new approach for representing lexical semantics has become the new default for novel models. These real-valued vectors are called "word embeddings" and have become a primary part of models for various applications, such as information retrieval [129], sentiment analysis [130], automatic summarization [131], MT [132] and QA [133].

5.2.2 Word Embeddings

Word embeddings is a distributed representation of words in a vector space that captures semantic and syntactic information for words [134]. The basic idea behind word embeddings is to use distributional similarity based representations by representing a word by means of its neighbors. Distributed representations of words help to enhance the performance of learning algorithms in various NLP tasks by grouping similar words. Mikolov et al. [135] introduced the skip-gram model and Continuous Bag of Words (CBOW) model for learning word embeddings from huge amounts of text data. An ANN structure has been used for learning word embeddings which encode many linguistics regularities and patterns explicitly. Some of these patterns can be represented as linear operations, e.g. the result of vector calculation: $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman}$ is closer to $\overrightarrow{queen}$ than to any other word vector as shown in Figure 5.10.

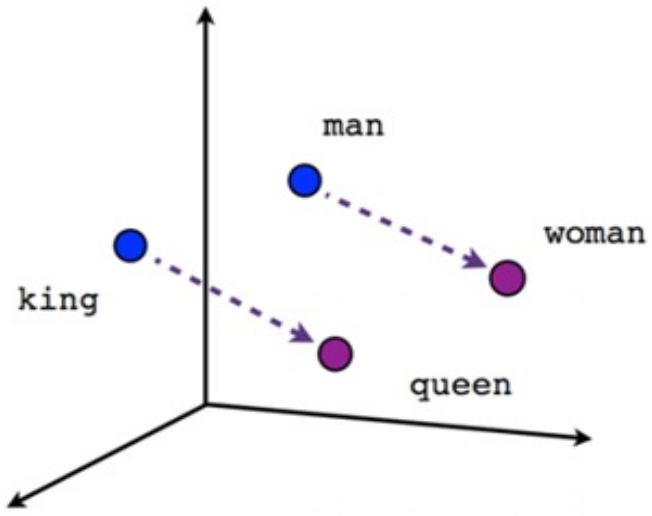


Figure 5.10: Example of word vector calculations

CBOW model trains the word embeddings by predicting a word in a sentence using its surrounding words, while Skip-gram model trains the word embeddings by predicting the surrounding words for a given word in the input layer. The structure of Skip-gram and CBOW models are shown in Figure 5.11.

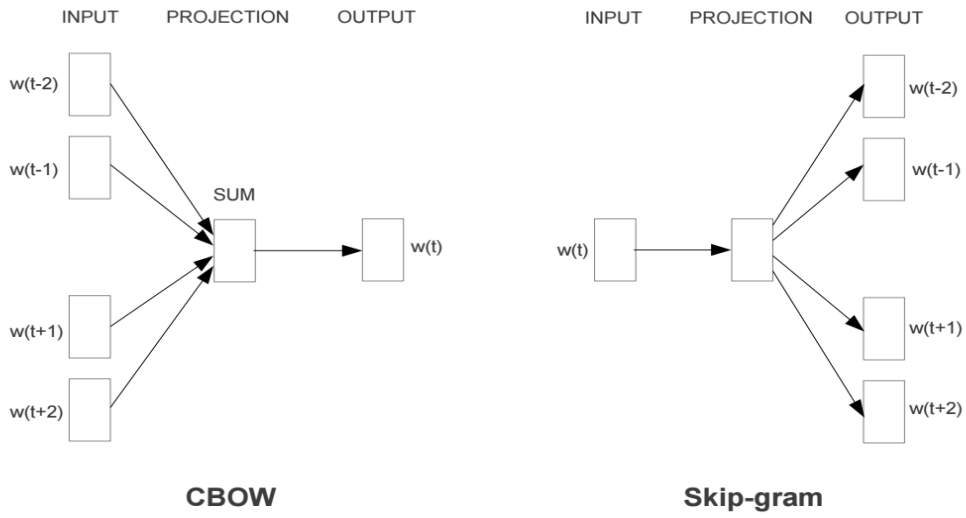


Figure 5.11: Structure of CBOW and Skip-gram models

The word embeddings e_t representing a word located at the i-th position in a sentence is calculated by maximizing the average log probability as follows:-

$$\text{In CBOW model} \qquad \frac{1}{T}\sum_{t=1}^{T} \log p(e_t | e_{t-\frac{n}{2}}, \ldots, e_{t-1}, e_{t+1}, \ldots, e_{t+\frac{n}{2}})$$

$$\text{In Skip-gram model} \qquad \frac{1}{T}\sum_{t=1}^{T} \log p(e_{t-\frac{n}{2}}, \ldots, e_{t-1}, e_{t+1}, \ldots, e_{t+\frac{n}{2}} | e_t)$$

where $e_{t-\frac{n}{2}}, \ldots, e_{t-1}$ are vectors for $\frac{n}{2}$ preceding words, $e_{t+1}, \ldots, e_{t+\frac{n}{2}}$ are vectors for $\frac{n}{2}$ subsequent words, and T is the number of tokens in the sentence.

Character-level Embeddings Word embeddings maintain semantic and syntactic information of words, but does not capture orthographical information at character level such as capitalization, numeric characters, special characters and hyphenation. However, such information plays an important role in Disease-NER as disease names can be characterized by the existence of combination of these information. Character-level word representation has been introduced to represent orthographical and morphological information [136].

5.3 Related Works

Researchers have explored many approaches for Disease-NER. Robert Leaman et al. [137] have addressed Disease-NER by learning the similarities between the disease mentions and concept names. They evaluated their approach using NCBI dataset and achieved a f1-measure of 80.9%. This was the first work to use pairwise learning to rank Disease NER approach but the performance was not high. A multi-task learning approach applied to BioNER using Neural Network architecture by Gamal C. et al. [40] has achieved a f1-measure of 80.73% for Disease-NER using NCBI dataset. In addition to NCBI dataset, 15 biomedical corpora have been used to train the system. These corpora have joint articles with NCBI test set and this affects the model evaluation. Leaman and Lu [68] used semi-Markov models to build TaggerOne, a tool for BioNER which reported a high competitive f1-measure of 82.9% for NCBI dataset. Sunil and Ashish [138] designed a RNN model for Disease-NER using Convolution Neural Network for representing character embeddings and bidirectional LSTM for word embeddings. A pre-trained word embeddings trained over a corpus of PubMed articles have been used for training the model and it is not enough to represent words in general domain. They evaluated their system on NCBI dataset and achieved f1-measure of 79.13%. Wei et al. [139] developed an ensemble-based system for Disease-NER. At the base level, they built CRF with a rule-based post-processing system and Bi-RNN based system. At the top level, they used SVM classifier for combining the results of the base systems. The proposed system uses manually extracted features for SVM and CRF as well as hand-crafted rules which depends on human experts. BC5CDR corpus has been used for evaluation and the model reported a f1-measure of 78.04%.

BANNER [140], a tool which implements CRF algorithm for BioNER using general features such as orthographical, linguistic and syntactic dependency features has reported a f1-measure of 81.8% on NCBI dataset. Xu et al. [141] designed a system (CD-REST) for chemical-induced disease relations from biomedical texts. A CRF-based module for NER as first step is designed using character level, word level features, context features and distributed word representation features learned from external un-annotated corpus. They evaluated the system using BC5CDR dataset and reported a f1-measure of 84.43%. Zhao et al. [142] developed a system for Disease-NER based on convolutional neural network. The proposed system integrated with dictionary information and a post-processing module has been used for performance enhancements. NCBI and BC5CDR datasets have been used for evaluation of the proposed system and reported a f1-measure of 85.17%

and 87.83% respectively. Haodi Li et al. [143] designed an end-to-end system used for chemical-disease relation extraction. The proposed system uses an ensemble approach using SVM and CRF as base classifiers with SVM as a meta-classifier to build a module for Disease-NER. BC5CDR dataset has been used to evaluate the system and reported a f1-measure of 86.93%. Hsin-Chun Lee et al. [144] presented an enhanced CRF-based system for Disease-NER. Rich feature set in addition to dictionary based features extracted from different lexicons have been used to train CRF. BC5CDR dataset has been used to evaluate has been used for system evaluation and reported f1-measure of 86.46%.

In this chapter, a DL model has been designed for Disease-NER using BiLSTM for learning the model and CRF for decoding the results with the following objectives:

- Using dictionary information for each token
- Using pre-trained word embeddings trained over a huge corpus of texts from biomedical domain as well as generic domain
- Learning a BiLSTM model for character-level word representations instead of using hand engineered features

5.4 Methodology

General structure of the proposed model is given in Figure 5.12. Proposed model accepts a sequence of words, for example, the sentence "*The risk of colorectal cancer was significantly high.*" and the associated tags "(O, O, O, B-Disease, I-Disease, O, O, O, O)" as input and gives a vector representation for each word containing information about the word itself and the neighbouring words within a sentence, denoted as contextual representation. In addition to word embeddings and character-level embeddings, dictionary information for each token has been extracted from MErged DIsease voCabulary (**MEDIC**) [145]. MEDIC is a comprehensive and publicly available dictionary for disease entities which provides information including disease names, concept identifiers, definitions of diseases and synonyms. Figure 5.13 shows the interface of MEDIC, and information resulted using the search key "Acute malaria". Dictionary information for each word is represented as a binary vector containing information about the existence of the token in the dictionary either solo or as a part of multi-word disease name or abbreviation or a synonym of a disease.

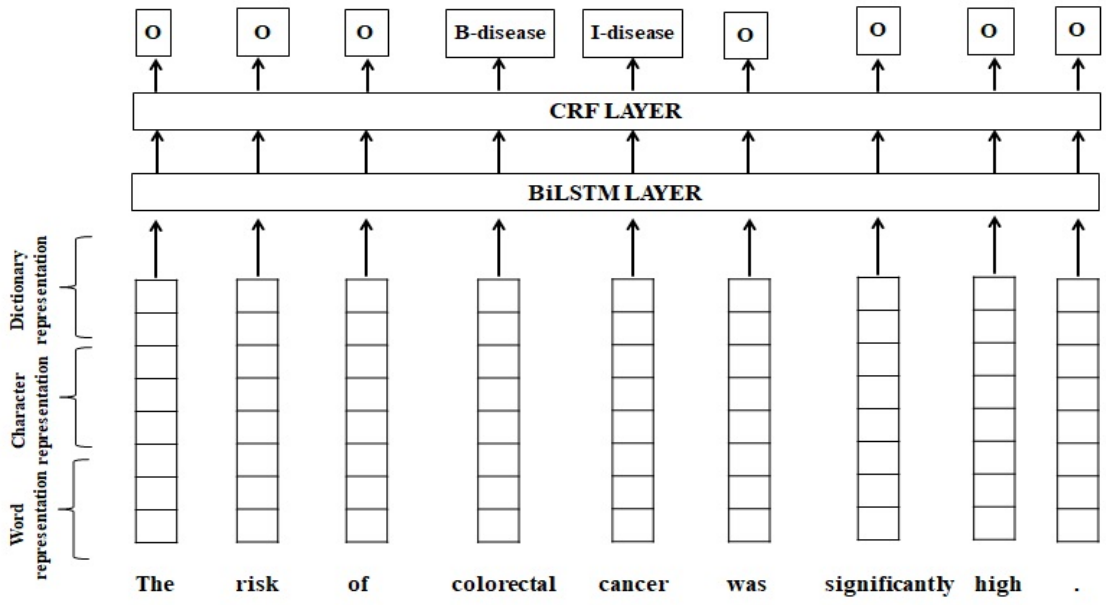


Figure 5.12: General structure of the proposed model

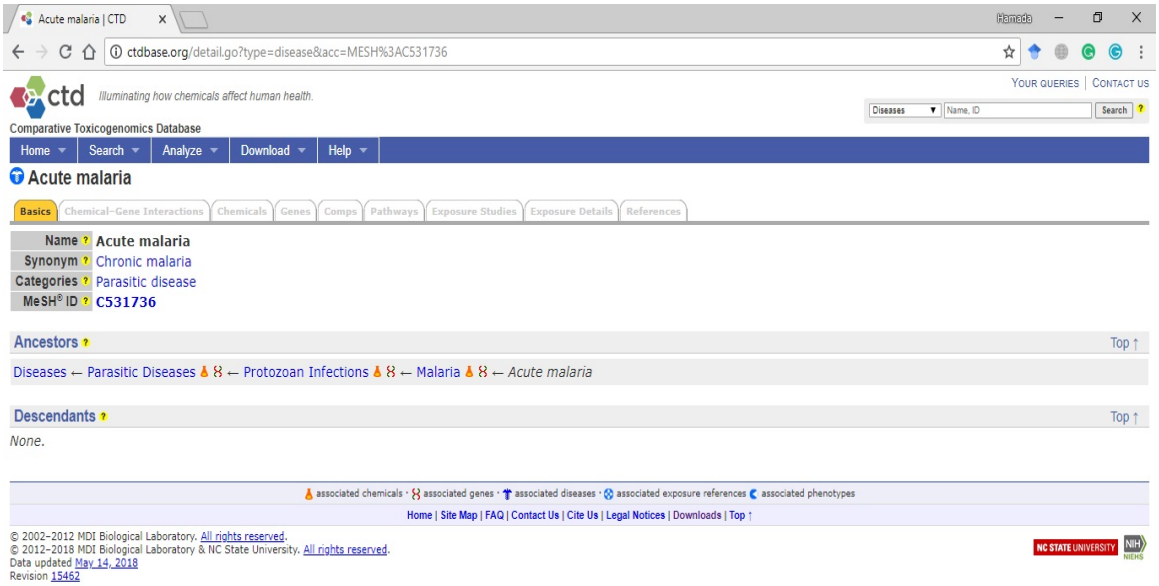


Figure 5.13: Dictionary interface for the entry “Acute malaria”

Character level representation is concatenated with word embeddings and the dictionary information. At this level, every word is represented as a vector comprising of character level information, word level information and dictionary information. Feeding the vector representation of word sequence of a sentence in direct and reverse order to a BiLSTM network will output a contextual representation for each word as shown in Figure 5.14.

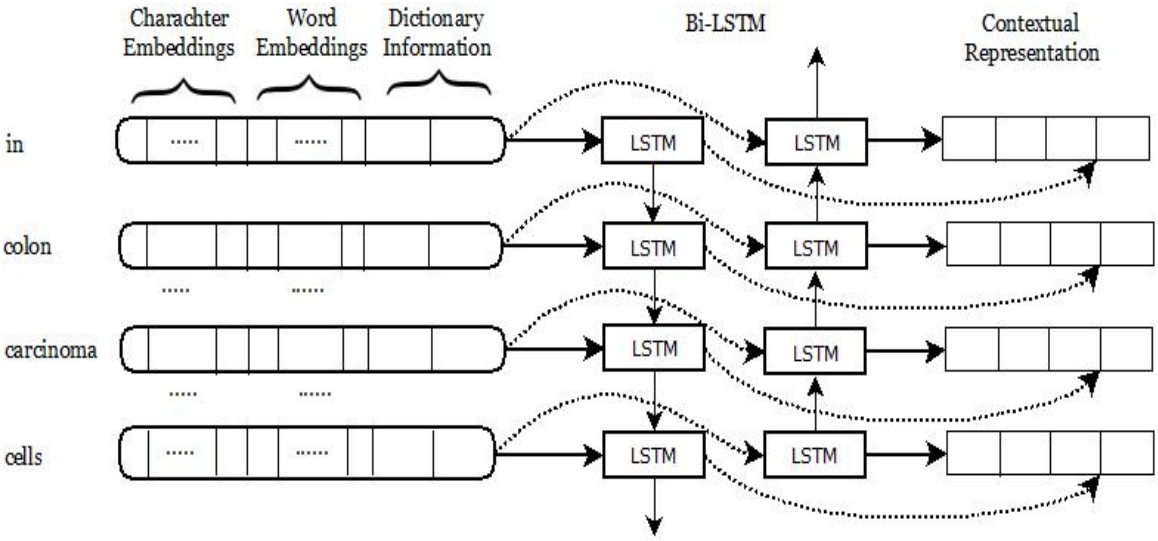


Figure 5.14: The contextual representation learning model

5.4.1 Decoding

Decoding is the final step, which converts the contextual representations of the tokens into corresponding tags. For sequence labeling problem, it is beneficial to consider the correlations between labels of the tokens in the neighbourhoods and jointly decode the best chain of labels for a given input sentence. CRF [82] model is used for decoding as CRF considers the contextual information for decoding the label for each token. There are two approaches for calculating the scores of output tags; local scores and global scores. Local scores use scores represented in the final contextual representations for each word, while global scores use transition scores as well as scores represented in the final contextual representations for each word.

A fully connected neural network has been used to convert the contextual representations of the tokens to a vector where each entry corresponds to a score for each output tag. For an input sentence, $\mathbf{X} = (x_1, x_2, ..., x_m)$, $\mathbf{Y} = (y_1, y_2, ..., y_m)$ is the sequence of output tags,

$S \in R^{T \times m}$ is the score matrix, where T is number of predefined tags and $s_{i,j} \in S$ is the score of i^{th} tag for the j^{th} word. A global score, $Score \in R$ of sequence of tags $\mathbf{Y} = (y_1, y_2, ..., y_m)$ is defined as:

$$Score(y_1, y_2, ..., y_m) = \sum_{t=1}^{m} \Big(s_{y_t,t} + Tr[y_t, y_{t+1}] \Big)$$

where, $Tr[y_t, y_{t+1}]$ is the score of assigning the tag y_{t+1} given the tag y_t.
The sum of scores of all possible sequences of the tags for an input sentence $\mathbf{X}$ is calculated as:

$$Z = \sum_{y'_i \in Y(x), 1 \leq i \leq m} e^{Score(y'_1, y'_2, ..., y'_m)}$$

Then given the sentence $\mathbf{X} = (x_1, x_2, ..., x_m)$, the conditional probability of a label sequence $\mathbf{Y} = (y_1, y_2, ..., y_m)$ is defined as:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{e^{Score(y_1, y_2, ..., y_m)}}{Z}$$

The predicted tag sequence $\hat{y} \in Y(\mathbf{x})$ is calculated as:

$$\hat{y} = \textbf{argmax}\{P(\mathbf{Y}|\mathbf{X})\}$$

Example of decoding a contextual representation for text fragment "*In colon carcinoma cells*" is given in Figure 5.15. In this example, numbers in columns are scores of assigning corresponding tags to the word. CRF is used to calculate the score of all paths and then the path with maximum score will be chosen. The global scores of two sequences are calculated as follows:

Score(O, B-Disease, I-Disease, O) = 4+3+2+5+4+7+2+8+1=36

Score(O, B-Disease, B-Disease, O) = 4+3+2+5+1+9+1+8+1=34

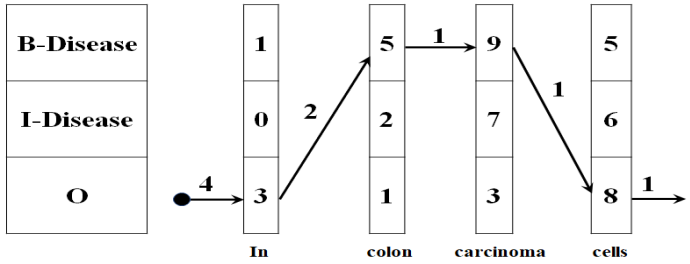
The first path having higher score will be selected by CRF.

In addition to this approach, a local score can be used to decode the output vectors to tags as follows:

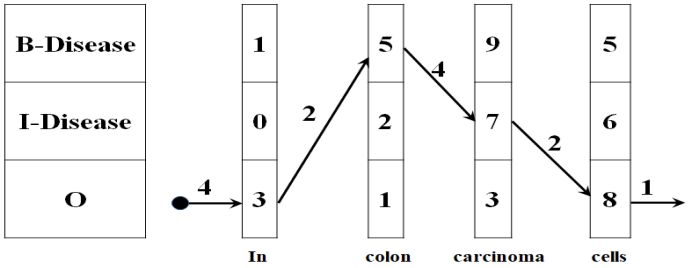
$$local\_score(y_1, y_2, ..., y_m) = \sum_{t=1}^{m} s_{y_t,t}$$

The predicted tag sequence $\hat{y} \in Y(\mathbf{x})$ is calculated as:

$$\hat{y} = \textbf{argmax}\{local\_score(y_1, y_2, ..., y_m)\}$$



(a) Path of the sequence ((O, B-Disease, B-Disease, O)



(b) Path of the sequence ((O, B-Disease, I-Disease, O)

Figure 5.15: A typical BioNER system with an example

5.5 Experiments

5.5.1 Pre-processing

Pre-trained word embeddings using Skip-gram model are used to represent tokens in the data set. This word embeddings[1] model combines domain-specific texts (PMC and PubMed texts) with generic ones (English Wikipedia dump) for better representation of tokens. We have used pre-trained word embeddings trained by Sampo Pyysalo et al. [146], as constructing word embeddings requires a huge corpus and machines with high specifications. Using skip-gram model for training word embeddings improves the semantic and syntactic representation. Character-level embeddings are created by initializing vector representations for every character in the corpus and then the character representation corresponding to every character in a word are given in direct and reverse order to BiLSTM as shown in Figure 5.16. All numbers in the input are replaced with the value `NUM` (number), all letters are converted to lowercase and words that are not represented in the pre-trained word embeddings are marked as `UNK` (unknown). As the pre-trained word embeddings are of large size, a look-up table containing the word embeddings of all words in the dataset are extracted from the pre-trained word embeddings which used as input. To evaluate the im-

[1] http://bio.nlplab.org/

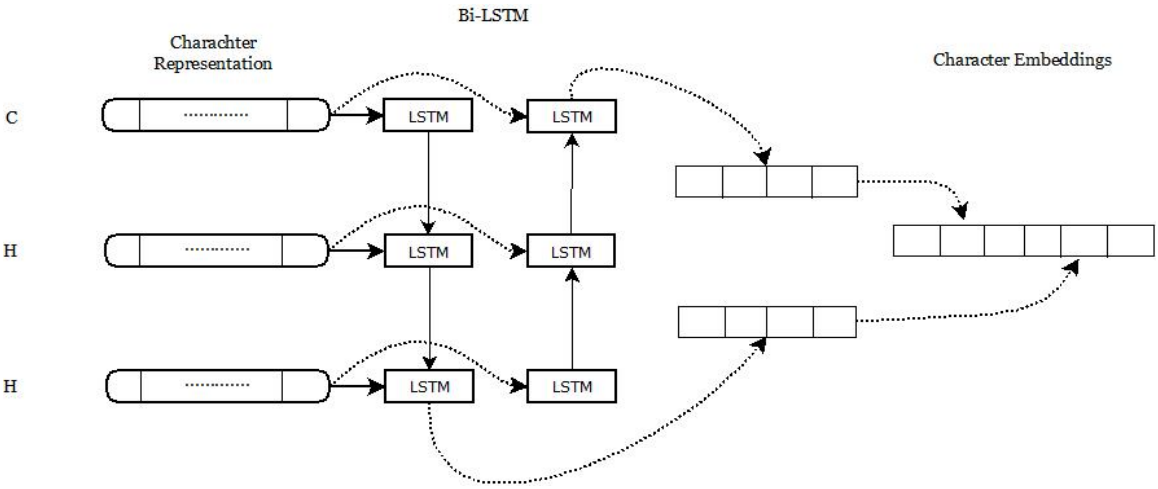


Figure 5.16: Character representation learning model

pact of the pre-trained word embeddings, randomly initialized word embeddings have been used separately.

5.5.2 Parameters

In this work, the dimension of character-level embeddings is set to 100 and that of the pre-trained word embeddings to 200. Different values for these parameters have been experimented and we set the values which reported the best performance for our experiments. Table 5.1 shows the parameters and their values used in our experiments.

5.6 Results and Discussion

NCBI and BC5CDR datasets described in Section 2.5.3 have been used to evaluate the proposed model. SR schemes namely IOB2 and IOBES have been used to train the model as these two schemes reported high performance among other schemes for BioNER as discussed in Chapter 3.

Results shown in Table 5.2 illustrate that the character embeddings significantly increase the performance for both datasets. Decoding the output vectors using CRF with global scores improves the performance rather than using local scores. The pre-trained word embedding results in better performance than randomly initialized word embeddings. The reason is that these pre-trained embeddings are trained over a collection of huge texts in-

Parameter	**Value**
No. of epochs	15
Dropout	0.5
Batch size	20
Learning method	Adam
Learning decay	0.9
No. of LSTM units for character embeddings	100
No. of LSTM units for word embeddings	300
No. of distinct words in NCBI dataset	8147
No. of distinct words in BC5CDR dataset	13427
No. of distinct characters in NCBI dataset	84
No. of distinct characters in BC5CDR dataset	80

Table 5.1: Parameters and their values used for training the proposed model

cluding texts from biomedical domain as well as generic domain. In both datastes, the best results are reported using character embeddings, pre-trained word embeddings, dictionary information and CRF with global scores. IOBES and IOB2 schemes do not show significant difference with NCBI dataset. However, using the same schemes with BC5CDR dataset shows a significant difference of 1.13 in the f1-measure.

Dataset	SR scheme	V1	V2	V3	V4	f1-measure
NCBI	IOB2	x	x	x	x	71.16%
		x	x	x	✓	80.46%
		x	x	✓	✓	83.13%
		x	✓	✓	✓	84.24%
		✓	✓	✓	✓	85.19%
	IOBES	x	x	x	x	73.23%
		x	x	x	✓	81.60%
		x	x	✓	✓	83.44%
		x	✓	✓	✓	84.40%
		✓	✓	✓	✓	**85.40**%
BC5CDR	IOB2	x	x	x	x	73.26%
		x	x	x	✓	75.81%
		x	x	✓	✓	78.30%
		x	✓	✓	✓	78.36%
		✓	✓	✓	✓	78.49%
	IOBES	x	x	x	x	73.64%
		x	x	x	✓	76.98%
		x	x	✓	✓	77.95%
		x	✓	✓	✓	79.33%
		✓	✓	✓	✓	**79.62**%

Table 5.2: Results of the proposed model with following variations:

V1: (x) without dictionary information (✓) with dictionary information
V2: (x) randomly initialized word embeddings (✓) pre-trained word embeddings
V3: (x) using local score for decoding (✓) using global score for decoding
V4: (x) without character embeddings (✓) with character embeddings

Table 5.3 and Table 5.4 give the comparisons between our model and the state-of-the-art models for NCBI and BC5CDR datasets respectively. For NCBI dataset, our model outperforms the state-of-the-art works.

BC5CDR dataset was originally created to extract the relationship between chemicals and diseases. Due to ambiguity nature of words, some of the NEs are considered as chemical NEs as well as disease NEs. This ambiguity has affected the performance of our model.

Model	**f1-measure**
ANN (BiLSTM + CNN) [138]	79.13%
Pairwise learning [137]	80.90%
ANN (ReLU + Softmax activiation) [40]	80.74%
TaggerOne (Semi-Markov model) [68]	82.90%
BANNER (CRF) [140]	81.80%
CNN (Dictionary + Postprocessing) [142]	85.17%
Our model	**85.40**%

Table 5.3: Comparison of our model with related work on NCBI dataset

Model	**f1-measure**
LSTM [147]	76.50%
Ensemble (RNN + CRF) [139]	78.04%
CD-REST (CRF + External resource) [141]	84.43%
CRF (dictionary Information)[144]	86.46%
Ensemble(SVM+CRF) [143]	86.93%
CNN (Dictionary + Postprocessing) [142]	87.83%
Our model	**79.62**%

Table 5.4: Comparison of our model with related work on BC5CDR dataset

5.7 Chapter Summary

In this chapter, an ANN-based model for Disease-NER is presented. Instead of hand engineering the features for tokens, character-level embeddings are used to represent orthographical features of tokens in addition to using word embeddings and dictionary information. Two different SR schemes namely IOB2 and IOBES are used for annotating the corpus. Results show that using character embeddings, pre-trained word embeddings, dictionary information and CRF with global scores improves the performance of BioNER. In addition, IOBES scheme outperforms IOB2 scheme.

Chapter 6

FROBES: An Extended SR Scheme for Multi-word BioNER

In this chapter, an extension of IOBES model is proposed to improve the performance of BioNER. The proposed SR model, FROBES, improves the representation of multi-word entities. A BiLSTM network is used to design a baseline system for BioNER and the new SR model is evaluated on i2b2/VA 2010 challenge dataset and JNLPBA 2004 shared task dataset. Results obtained illustrate that the proposed SR model outperforms IOB2 and IOBES schemes for multi-word entities with length greater than two. Further, the outputs of different SR models combined using majority voting ensemble method illustrate that ensemble method outperforms the performance of baseline models.

6.1 Preamble

SR schemes have been applied for different NLP tasks such as Noun Phrase chunking (NP-chunking) [60, 56], word segmentation [148, 149] and NER [150, 151, 152]. The research works carried out in BioNER are centred around developing efficient techniques and tools for BioNER. Improving SR schemes may enhance the performance of BioNER but are less explored. Different SR schemes such as IO, IOB2, IOE2, IOBE and IOBES are used to represent the data. For example, IOB2 scheme discriminates the left boundary of BioNEs, while IOE2 discriminates the right boundary of BioNEs. IOBE scheme combines both IOB2 and IOE2 schemes. IOBES scheme highlights both boundaries in addition to the

Some parts of the material of this chapter have appeared in the following research paper
Hamada A. Nayel, H. L. Shashirekha, Hiroyuki Shindo, Yuji Matsumoto: "Improving Multi-Word Entity Recognition for Biomedical Texts," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 16, pp. 301-319, 2018.

single-word BioNEs. Most of SR schemes concentrated on both boundaries of BioNEs in addition to single word BioNEs. For multi-word BioNEs containing more than two words, all intermediate words will be assigned the same tag I-XXX, where I represents "Intermediate" and XXX is the class name.

Multi-word BioNEs are the critical challenges that should be considered while developing BioNER systems as most of the BioNEs consist of multiple words. Statistics of lengths of NEs for JNLPBA and i2b2 datasets are given in Table 6.1.

Length of the NE	**Number of NEs**			
	JNLPBA Dataset		**i2b2 Dataset**	
	Test	**Train**	**Test**	**Train**
N = 1	3466	21646	14116	7497
	40.01%	42.19%	45.31%	45.38%
N = 2	2620	15442	8469	4441
	30.25%	30.10%	27.18%	26.88%
N = 3	1240	7530	4573	2365
	14.32%	14.68%	14.68%	14.32%
N > 3	1336	6683	3996	2216
	15.42%	13.03%	12.83%	13.42%
Total	8662	51301	31154	16519

Table 6.1: Statistics for lengths of NEs in JNLPBA and i2b2 datasets

In this chapter, a new SR model is proposed to enhance the representation and extraction of multi-word BioNEs. Further, the outputs of different SR models are combined using majority voting ensemble method and the performance of the proposed SR model is evaluated on i2b2/VA 2010 challenge dataset and JNLPBA 2004 shared task dataset.

6.2 Related Work

ML algorithms such as SVM [153], CRF [154] and ME [155] used for BioNER depend essentially on extracting feature set used for training. Haode et al. [118] have used Convolutional Neural Network (CNN) based model for BioNEs normalization. Xu et al. [17] designed a model using BiLSTM and CRF model for clinical NE extraction. A randomly initialized word embeddings have been used as input to the proposed model and using such embeddings does not represent the semantics of the inputs. They used NCBI disease corpus to evaluate their model and have reported a f1-measure of 80.22%.

Lots of research works have been carried out to study the performance of SR models on BioNER [140]. Han-Cheol Cho et al. [58] studied the performance of different SR models using linear chain CRFs to learn a base model for NER. Shashirekha and Nayel [152] studied the performance of BioNER using different SR models. Using CRFs and SVMs for learning the baseline systems for biomedical entity extraction they have compared different SR models on JNLPBA dataset and i2b2/VA 2010 shared task dataset. An extension of IOBES scheme has been introduced by Keretna et al. [16] to improve BioNER by introducing a new tag to resolve the problem of ambiguity and was evaluated on i2b2/VA 2010 shared task dataset. In their work, the same word may be assigned with three different classes which increases the complexity.

CRF algorithm with rich textual feature set has been used to train a model for i2b2/VA 2010 shared task by Gurulingappa et al. [156]. Proposed model integrated with a post-processing rule-based module reported 81.8% f1-measure. Performance of this approach depends essentially on the features extracted for CRF and post-processing rules which requires domain knowledge. Zhang and Elhadad [157] developed an unsupervised model for BioNER using IDF information and stem values using Unified Medical Language System (UMLS) meta-thesaurus. JNLPBA shared task dataset and i2b2/VA 2010 shared task dataset have been used to evaluate the proposed model and f1-measure of 15.20% and 26.50% respectively is reported.

6.3 Proposed Model

6.3.1 FROBES

FROBES, an extension of IOBES model is proposed to represent multi-word entities using the tags **F/R/O/B/E/S** for front, rear, outside, begin, end, single respectively. FROBES is designed to discriminate the first part and the second part of multi-word NEs. In FROBES, in addition to tagging intermediate words in multi-word NEs information about position of the word in an entity is also added. For all tokens at the rear/front of the entity, the tags **R/F** are used respectively. This information helps in improving the learning process.

In this model, the tag **I** of IOBES model is replaced by the tags **F** and **R** for entities of length greater than two words. This model describes both halves of the entities, the first half contains tags **B** and **F**, and the second half contains tags **R** and **E**. The representation of the proposed model and other models is shown in Figure 6.1.

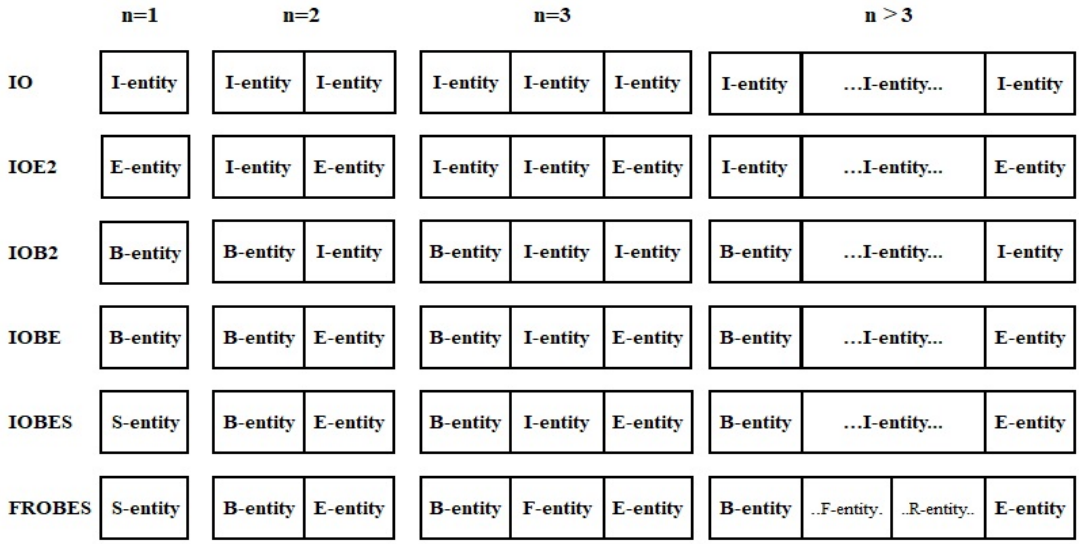


Figure 6.1: SR models

FROBES differentiates between the words at the beginning and ending of the multi-word entity. Some multi-word BioNEs have the property of common endings. In many cases, these common ending helps in determining the entity class. For example, many protein names have the common expression "*transcription factor*" at the end of the entity such as; "zinc finger *transcription factor*", "human proximal sequence element-binding *transcrip-*

tion factor" and "B-cell specific *transcription factor*". Similarly, many DNA names have the expression "*binding site*" at the end of the DNA name such as; "hexameric receptor *binding site*" and "erythroid Kruppel-like factor (EKLF) *binding site*". So, expanding tags of multi-word entities to differentiate between both sides of BioNEs may help not only in determining the BioNE, but also to assign it the correct class.

An example of tagging the NE protein "*human proximal sequence element-binding transcription factor*" using FROBES is shown in Figure 6.2.

human	**B-protein**
proximal	**F-protein**
sequence	**F-protein**
element-binding	**R-protein**
transcription	**R-protein**
factor	**E-protein**

Figure 6.2: An example of representing a protein name with FROBES

The general structure of FROBES scheme for representing multi-word entities where the tags **B** and **E** are used for the first and last token of an entity respectively is shown in Figure 6.3. Moreover, tags **F** and **R** are used for the tokens inside the entity.

B	... **F** ...	... **R** ...	**E**

Figure 6.3: General structure of FROBES SR scheme

The improvement of FROBES is achieved via differentiating both front and rear halves of intermediate part of BioNE. For specific BioNE classes, common words and expressions are usually used for describing these entities as described above. FROBES focuses on representing these common words by assigning the different halves of intermediate words of multi-word BioNEs with different tags to differentiate these parts.

6.3.2 Baseline model

The main structure of our model is given in Figure 6.4. The proposed model accepts a sequence of words $S = (w_1, w_2, ..., w_n)$ and the associated tags $T = (t_1, t_2, ..., t_n)$ as input and gives a contextual representation for each word as output. Each word is represented by two types of vectors namely character embeddings and word embeddings as described in Chapter 5. Character embeddings are used to capture the orthographic features of the words such as capitalization, hyphenation or special characters. Feeding these vectors to a BiLSTM network will output a contextual representation for each word as shown in Figure 6.4. The final step is decoding, i.e., converting the contextual representations into output tags as presented in Section 5.4.1.

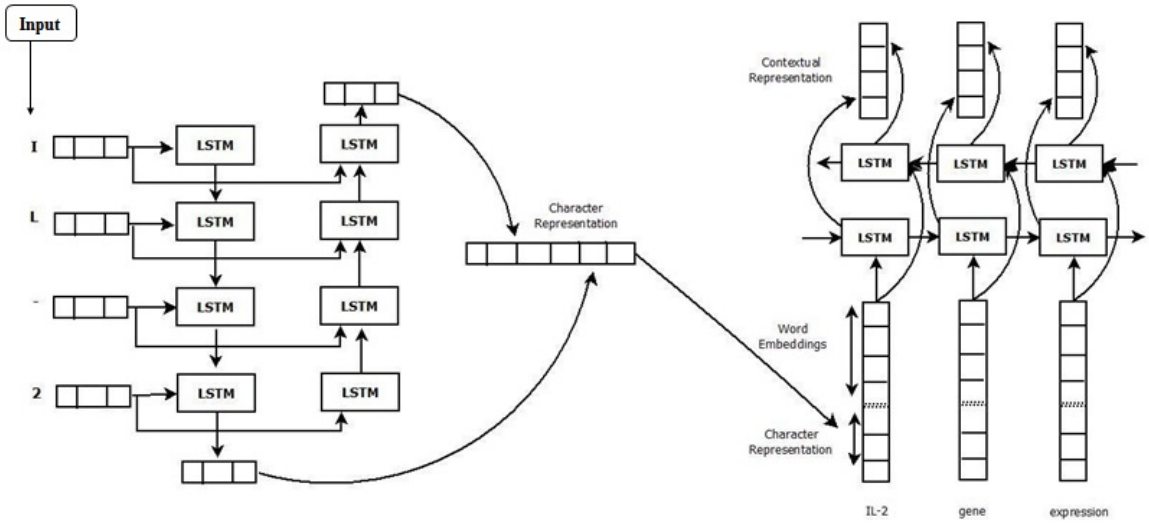


Figure 6.4: Character and contextual representation learning process

6.4 Experiments

The experiments were conducted using ANN model containing BiLSTM layer for character and word context representation and a CRF layer for decoding the contextual representation into tags. The proposed system has been evaluated using JNLPBA 2004 shared task dataset [46] and i2b2/VA 2010 challenge dataset [49] (mentioned in Sections 2.5.1 and 2.5.2 respectively) and performance of the system is reported in terms of precision, recall and f1-measure [53].

6.5 Results and Discussion

FROBES model is designed to represent long entities with more appropriate tags. Overall results of base line system in terms of Recall (R), Precision (P) and f1-measure with three SR schemes IOB2, IOBES and FROBES are shown in Table 6.2. Majority voting technique has been used to combine the outputs of these models using ensemble approach and results are shown in Table 6.3. General structure of ensemble approach applied in our work is shown in Figure 6.5. Results are shown in Table 6.2 and illustrate that the proposed model improves recall and f1-measure for JNLPBA dataset and precision for i2b2 dataset. FROBES improved f1-measure and recall of JNLPBA test set because the ratio of BioNEs with length greater than three is greater than that of i2b2/VA shared task dataset as shown in table 6.1. FROBES focuses on multi-words BioNEs with lengths greater than three. Moreover, results show that f1-measure of ensemble approach is near to the state-of-the-art works for both datasets.

Dataset	**Evaluation Measure**	**Baseline SR model**		
		IOB2	**IOBES**	**FROBES**
JNLPBA	R	75.18%	75.87%	76.23%
	P	67.82%	67.68%	67.69%
	f1-measure	71.31%	71.54%	71.71%
i2b2	R	81.56%	82.07%	81.74%
	P	83.84%	84.57%	84.62%
	f1-measure	82.68%	83.30%	83.15%

Table 6.2: Results of different SR models with baseline system

Dataset	**Baseline SR model**			**Ensemble**
	IOB2	**IOBES**	**FROBES**	
JNLPBA	71.31%	71.54%	71.71%	71.99%
i2b2	82.68%	83.30%	83.15%	83.62%

Table 6.3: f1-measure of different baseline SR models and Ensemble approach

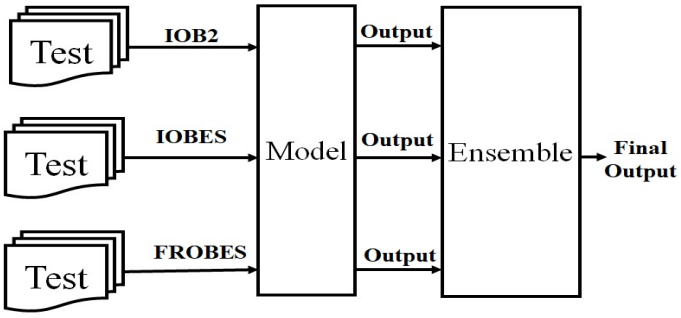


Figure 6.5: General structure of ensemble approach applied in our model

Dataset	Number of tokens per entity	Baseline SR models			Ensemble
		IOB2	IOBES	FROBES	
JNLPBA	N = 1	73.79%	73.57%	73.59%	73.91%
	N = 2	74.11%	74.10%	74.38%	74.34%
	N $\geq$ 3	64.53%	65.63%	65.83%	66.46%
i2b2	N = 1	87.66%	88.04%	87.92%	88.23%
	N = 2	81.64%	82.21%	81.63%	82.31%
	N $\geq$ 3	75.58%	76.82%	77.00%	77.48%

Table 6.4: F1-measure for JNLPBA and i2b2/VA 2010 shared task dataset

Table 6.4 shows the results of baseline systems with different SR schemes for JNLPBA and i2b2 datasets with different lengths of entities. In case of JNLPBA dataset, existing SR schemes, namely IOB2 and IOBES, achieved moderate performance for single word ($N = 1$) and double word ($N = 2$) BioNEs. Our proposed scheme focuses on multi-word NEs ($N \geq 3$) and gives better results when compared to the existing models (IOB2 and IOBES). This clearly indicates that our proposed scheme gives better performance for multi-word BioNEs instead of short lengths BioNEs.

Moreover, ensembling the existing models namely, IOB2 and IOBES with the proposed scheme gives better performance for all possible lengths of BioNEs and this is clearly indicated in the ensemble column of Table 6.4.

Similarly, in case of i2b2 dataset, the proposed scheme and ensemble outperformed IOB2 and IOBES schemes. IOB2 and IOBES schemes were focused on representing the boundaries of the NEs and the results were moderate. To enrich multi-word NEs representation, we added more tags to represent intermediate words in addition to boundaries of a NE which leads to better NER performance.

Table 6.5 and Table 6.6 give the comparisons between our model and the state-of-the-art models for JNLPBA and i2b2 datasets respectively. For i2b2 dataset, our model outperforms the state-of-the-art models.

Model	**f1-measure**
Unsupervised [157]	15.20%
Ensemble [44]	68.20%
CRF (Complete feature set) [19]	69.50%
CRF (Orthographic features only) [19]	69.80%
SVM and HMM [158]	72.55%
NERBio (CRF) [159]	72.98 %
Ensemble (CRF + SVM) [75]	75.17%
Our model	**71.99**%

Table 6.5: Comparisons of our model with related work on JNLPBA dataset

Model	**f1-measure**
Unsupervised [157]	26.50%
Pre-processing + CRF + Post-processing [160]	75.86%
Hybrid (CRF + Rule-based post processing) [156]	81.80%
Ensemble (ensemble of available tools) [100]	82.20%
CRF [161]	82.30%
CRF [162]	82.80%
Our model	**83.62**%

Table 6.6: Comparisons of our model with related work on i2b2 dataset

6.6 Chapter Summary

In this chapter, a new SR model - FROBES is presented to improve multi-word BioNEs representation. To compare FROBES with IO and IOBES, a BiLSTM based model has been used as a baseline system on JNLPBA and i2b2 datasets. Experimental results show that FROBES improves the performance of BioNER for multi-word BioNEs.

Chapter 7

Conclusion and Future Work

BioNER is a crucial task, as it is the first phase in any biomedical NLP pipeline and further tasks such as relation extraction, event extraction and knowledge representation depends on the performance BioNER. In this study, different aspects that affect BioNER such as SR schemes, learning models and feature engineering are explored. During the study ML algorithms such as SVM, CRF and ANN have been used to build BioNER systems. From the results of the study, it is noticeable that DL based models outperforms other ML based models. DL-based BioNER systems depends on word embeddings used as input, while classical ML-based BioNER systems depends on feature engineering. In addition, SR scheme which is used for representing the dataset is an important factor for the performance of BioNER systems. Our extended SR scheme improves the performance of BioNER for multi-word entities as multi-word is one of the major challenges for BioNER.

In this work, Chapter 1 gives an introduction to NLP and the need for BioNLP. Chapter 2 introduces BioNER, motivation, challenges and approaches for BioNER. In Chapter 3, the performance of BioNER using various SR scheme is studied. CRF and SVM based models have been used to evaluate the performance of BioNER with different SR schemes. Ensemble approach has been designed for BioNER in Chapter 4 as well as an ensemble-based system submitted to Indian Native Language Identification (INLI) task held in conjunction with FIRE 2017. In Chapter 5, a deep learning model for Disease-NER is designed. Proposed model is composed of two BiLSTM networks for training word embeddings and character embeddings, and CRF with global score for decoding step. FROBES, an extension of IOBES SR scheme has been introduced in Chapter 6. FROBES has been designed to improve multi-word BioNER.

Future work in BioNER includes exploring DL models on large datasets and integrating dictionary and rule based approaches with DL models. Enhancing word embeddings and their variants (various character-level word embeddings and end-to-end trained embeddings) to check the performance of DL models.

Bibliography

[1] Gary Miner, John Elder IV, and Thomas Hill. *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.

[2] Marko Grobelnik, Dunja Mladenic, and Natasa Milic-Frayling. Text mining as integration of several related research areas: report on kdd's workshop on text mining 2000. *ACM SIGKDD Explorations Newsletter*, 2(2):99–102, 2000.

[3] Ian H Witten, Katherine J Don, Michael Dewsnip, and Valentin Tablan. Text mining in a digital library. *International Journal on Digital Libraries*, 4(1):56–59, 2004.

[4] Aron Henriksson, Jing Zhao, Hercules Dalianis, and Henrik Boström. Ensembles of randomized trees using diverse distributed representations of clinical events. *BMC medical informatics and decision making*, 16(2):69, 2016.

[5] Israel Alonso and David Contreras. Evaluation of semantic similarity metrics applied to the automatic retrieval of medical documents: An umls approach. *Expert Systems with Applications*, 44:386 – 399, 2016.

[6] Aaron M. Cohen and William R. Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71, 2005.

[7] Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B Cohen. Frontiers of biomedical text mining: current progress. *Briefings in bioinformatics*, 8(5):358–375, 2007.

[8] Changki Lee, Yi-Gyu Hwang, and Myung-Gil Jang. Fine-grained named entity recognition and relation extraction for question answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 799–800, New York, NY, USA, 2007. ACM.

[9] Rohini K Srihari and Erik Peterson. Named entity recognition for improving retrieval and translation of chinese documents. In *International Conference on Asian Digital Libraries*, pages 404–405. Springer, 2008.

[10] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[11] Annalina Caputo, Pierpaolo Basile, and Giovanni Semeraro. Boosting a semantic search engine by named entities. In *International Symposium on Methodologies for Intelligent Systems*, pages 241–250. Springer, 2009.

[12] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools, Improving MT Through Other Language Technology Tools: Resources and Tools for Building MT*, EAMT '03, pages 1–8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[13] Goran Nenadic, Simon Rice, Irena Spasic, Sophia Ananiadou, and Benjamin Stapley. Selecting text features for gene name classification: from documents to terms. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 121–128, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[14] Kate Byrne. Nested named entity recognition in historical archive text. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007), September 17-19, 2007, Irvine, California, USA*, pages 589–596, 2007.

[15] Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. Feature selection techniques for maximum entropy based biomedical named entity recognition. *Journal of Biomedical Informatics*, 42(5):905 – 911, 2009.

[16] Sara Keretna, Chee Peng Lim, Doug Creighton, and Khaled Bashir Shaban. Enhancing medical named entity recognition with an extended segment representation technique. *Computer Methods and Programs in Biomedicine*, 119(2):88 – 100, 2015.

[17] Kai Xu, Zhanfan Zhou, Tianyong Hao, and Wenyin Liu. *A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition*, pages 355–365. Springer International Publishing, Cham, 2018.

[18] R. Grishman and B. Sundheim. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, June 1996.

[19] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110, Geneva, Switzerland, August 28th and 29th 2004. COLING.

[20] Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6):396 – 410, 2004. Named Entity Recognition in Biomedicine.

[21] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.

[22] Sophia Ananiadou and John Mcnaught. *Text Mining for Biology And Biomedicine*. Artech House, Inc., Norwood, MA, USA, 2005.

[23] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 82–86, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[24] Philip V Ogren, K Bretonnel Cohen, George K Acquaah-Mensah, Jens Eberlein, and Lawrence Hunter. The compositional structure of gene ontology terms. In *Biocomputing 2004*, pages 214–225. World Scientific, 2003.

[25] Tor-Kristian Jenssen, Astrid Lægreid, Jan Komorowski, and Eivind Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature genetics*, 28(1):21–28, may 2001.

[26] Yoshimasa Tsuruoka and Junichi Tsujii. Improving the performance of dictionary-based approaches in protein name recognition. *Journal of Biomedical Informatics*, 37(6):461 – 470, 2004.

[27] Robert S. Boyer and J. Strother Moore. A fast string searching algorithm. *Commun. ACM*, 20(10):762–772, October 1977.

[28] Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, March 1987.

[29] Peter Weiner. Linear pattern matching algorithms. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory (Swat 1973)*, SWAT '73, pages 1–11, Washington, DC, USA, 1973. IEEE Computer Society.

[30] Thierry Lecroq. Fast exact string matching algorithms. *Information Processing Letters*, 102(6):229 – 235, 2007.

[31] Christian Lovis and Robert H Baud. Fast exact string pattern-matching algorithms adapted to the characteristics of the medical language. *Journal of the American Medical Informatics Association*, 7(4):378–391, 2000.

[32] Cheol Ryu and Kunsoo Park. Improved pattern-scan-order algorithms for string matching. *Journal of Discrete Algorithms*, 49:27 – 36, 2018.

[33] Baohua Gu. *Recognizing named entities in biomedical texts*. PhD thesis, School of Computing Science-Simon Fraser University, 2008.

[34] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing*, 1998.

[35] Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, and Hae-Chang Rim. Biomedical named entity recognition using two-phase model based on {SVMs}. *Journal of Biomedical Informatics*, 37(6):436 – 447, 2004. Named Entity Recognition in Biomedicine.

[36] Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411 – 422, 2004.

[37] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 473–480, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[38] Lishuang Li, Rongpeng Zhou, and Degen Huang. Two-phase biomedical named entity recognition using {CRFs}. *Computational Biology and Chemistry*, 33(4):334 – 338, 2009.

[39] Lishuang Li, Liuke Jin, Zhenchao Jiang, Dingxin Song, and Degen Huang. Biomedical named entity recognition based on extended recurrent neural networks. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 649–652, Nov 2015.

[40] Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18(1):368, Aug 2017.

[41] Y. Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.

[42] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, Third 2006.

[43] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[44] Jon Patrick and Yefeng Wang. Biomedical named entity recognition system. In *Proceedings of the Tenth Australasian Document Computing Symposium (ADCS 2005)*, 2005.

[45] Balu Bhasuran, Gurusamy Murugesan, Sabenabanu Abdulkadhar, and Jeyakumar Natarajan. Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases. *Journal of Biomedical Informatics*, 64:1 – 9, 2016.

[46] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *Proceedings of the International Joint Workshop on NLP in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 70–75, Geneva, Switzerland, August 2004. COLING.

[47] Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors. *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[48] J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19, 2003.

[49] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

[50] Rezarta Islamaj Doan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47(Supplement C):1 – 10, 2014.

[51] Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9(2):S2, Sep 2008.

[52] Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068, 2016.

[53] George Hripcsak and Adam S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[54] Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics*, 7(1):92, Feb 2006.

[55] Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared*

Task, BioNLP '09, pages 1–9, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[56] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *Proceedings of EACL'99*, pages 173–179. Bergen, Norway, 1999.

[57] Adwait Ratnaparkhi. *Maximum entropy models for natural language ambiguity resolution*. PhD thesis, University of Pennsylvania, PA, USA, 1998.

[58] Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, and Junichi Tsujii. Named entity recognition with multiple segment representations. *Information Processing & Management*, 49(4):954 – 965, 2013.

[59] A. Dehghan, J. A. Keane, and G. Nenadic. Challenges in clinical named entity recognition for decision support. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 947–951, Oct 2013.

[60] Frédéric Béchet, Alexis Nasr, and Franck Genet. Tagging unknown proper names using decision trees. In *Proceedings of the 38th Annual Meeting on ACL*, ACL '00, pages 77–84. ACL, 2000.

[61] Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. 1999.

[62] Taku Kudo and Yuji Matsumoto. Chunking with support vector machines. In *Proceedings of the second meeting of the North American Chapter of the ACL on Language technologies*, pages 1–8, 2001.

[63] Jiashen Sun, Tianmin Wang, Li Li, and Xing Wu. Person name disambiguation based on topic model. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*, page 391, 2010.

[64] Michal Konkol and Miloslav Konopík. Segment representations in named entity recognition. In *Text, Speech, and Dialogue: 18th International Conference, TSD 2015*, pages 61–70. Pilsen, Czech Republic, Springer, 2015.

[65] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[66] Erik F. Tjong Kim Sang and Fien De Meulder. 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*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[67] Michal Konkol and Miloslav Konopík. Crf-based czech named entity recognizer and consolidation of czech ner research. In Ivan Habernal and Václav Matoušek, editors, *Text, Speech, and Dialogue*, pages 153–160, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[68] Robert Leaman and Zhiyong Lu. Taggerone: Joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 2016.

[69] Chikashi Nobata, Nigel Collier, and Jun-ichi Tsujii. Automatic term identification and classification in biology texts. In *Proc. of the 5th NLPRS*, pages 369–374, 1999.

[70] Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8, Phildadelphia, Pennsylvania, USA, July 2002.

[71] S. Keretna, C. P. Lim, and D. Creighton. A hybrid model for named entity recognition using unstructured medical text. In *9th International Conference on System of Systems Engineering (SOSE)*, pages 85–90, June 2014.

[72] Richard Tzong-Han Tsai. A hybrid approach to biomedical named entity recognition and semantic role labeling. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: doctoral consortium*, pages 243–246, 2006.

[73] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[74] Tyler W. Rinker. *qdapDictionaries: Dictionaries to Accompany the qdap Package*. University at Buffalo/SUNY, Buffalo, New York, 2013. 1.0.7.

[75] Asif Ekbal and Sriparna Saha. Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowledge-Based Systems*, 46(0):22 – 32, 2013.

[76] Akiko Aizawa. An information-theoretic perspective of tf—idf measures. *Inf. Process. Manage.*, 39(1):45–65, January 2003.

[77] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[78] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[79] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[80] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[81] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003.

[82] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[83] Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In *Proceedings of the 13th Italian Workshop on Neural Nets-Revised Papers*, WIRN VIETRI 2002, pages 3–22, London, UK, UK, 2002. Springer-Verlag.

[84] Kerui Min, Chenggang Ma, Tianmei Zhao, and Haiyan Li. Bosonnlp: An ensemble approach for word segmentation and pos tagging. In Juanzi Li, Heng Ji, Dongyan Zhao, and Yansong Feng, editors, *Natural Language Processing and Chinese Computing*, pages 520–526, Cham, 2015. Springer International Publishing.

[85] Ted Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 63–69, Stroudsburg, PA, USA, 2000.

[86] Dan Klein, Kristina Toutanova, H. Tolga Ilhan, Sepandar D. Kamvar, and Christopher D. Manning. Combining heterogeneous classifiers for word-sense disambiguation. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions - Volume 8*, WSD '02, pages 74–80, Stroudsburg, PA, USA, 2002.

[87] Lluis Marquez, Horacio Rodriguez, Josep Carmona, and Josep Montolio. Improving pos tagging using machine-learning techniques. In *Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, 1999.

[88] Felice DellOrletta. Ensemble system for part-of-speech tagging. *Proceedings of EVALITA*, 9:1–8, 2009.

[89] Prasenjit Majumder, Mandar Mitra, Jainisha Sankhavara, and Parth Mehta, editors. *FIRE'17: Proceedings of the 9th Annual Meeting of the Forum for Information Retrieval Evaluation*, New York, NY, USA, 2017. ACM.

[90] TG Ditterrich. Machine learning research: four current direction. *Artificial Intelligence Magzine*, 4:97–136, 1997.

[91] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pages 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[92] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.*, 40(2):139–157, August 2000.

[93] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.*, 36(1-2):105–139, July 1999.

[94] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001.

[95] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, March 1998.

[96] Eugene M Kleinberg. A mathematically rigorous foundation for supervised learning. In *International Workshop on Multiple Classifier Systems*, pages 67–76. Springer, 2000.

[97] Youngjun Kim, Ellen Riloff, and John F Hurdle. A study of concept extraction across different types of clinical notes. In *AMIA Annual Symposium Proceedings*, volume 2015, page 737. American Medical Informatics Association, 2015.

[98] GuoDong Zhou, Dan Shen, Jie Zhang, Jian Su, and SoonHeng Tan. Recognition of protein/gene names from text using an ensemble of classifiers. *BMC bioinformatics*, 6(1):S7, 2005.

[99] Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(1):S3, May 2005.

[100] Ning Kang, Zubair Afzal, Bharat Singh, Erik M. van Mulligen, and Jan A. Kors. Using an ensemble system to improve concept extraction from clinical records. *Journal of Biomedical Informatics*, 45(3):423–428, June 2012.

[101] Burr Settles. Abner: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[102] Udo Hahn, Ekaterina Buyko, Rico Landefeld, Matthias Mühlhausen, Michael Poprat, Katrin Tomanek, and Joachim Wermter. An overview of jcore, the julie lab uima component repository. In *Proceedings of the LREC*, volume 8, pages 1–7, 2008.

[103] John Gibbons. *Forensic linguistics: An introduction to language in the justice system*. Wiley-Blackwell, 2003.

[104] Lourdes Ortega. *Understanding Second Language Acquisition*. Hodder Education, Oxford, 2009.

[105] Alla Rozovskaya and Dan Roth. Algorithm selection and model adaptation for esl correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 924–933, Portland, Oregon, USA, June 2011.

[106] Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. Author profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 263–272, 2007.

[107] Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75, September 2005.

[108] Carole E Chaski. Whos at the keyboard? authorship attribution in digital evidence investigations. *International journal of digital evidence*, 4(1):1–13, 2005.

[109] Scott Jarvis, Yves Bestgen, and Steve Pepper. Maximizing classification accuracy in native language identification. pages 111–118, 2013.

[110] Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013(2), 2013.

[111] Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In *Proceedings of COLING 2012*, pages 2585–2602. The COLING 2012 Organizing Committee, 2012.

[112] Shervin Malmasi, Mark Dras, and Irina Temnikova. Norwegian native language identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 404–412, 2015.

[113] Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. Topic modeling for native language identification. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 115–124, 2011.

[114] Anand Kumar M, Barathi Ganesh B, Shivkaran S, Soman K P, and Paolo Rosso. Overview of the inli pan at fire-2017 track on indian native language identification.

In *Notebook Papers of FIRE 2017, FIRE-2017*. Bangalore, India, December 8-10, CEUR Workshop Proceedings, 2017.

[115] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM, 2001.

[116] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[117] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[118] Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. Cnn-based ranking for biomedical entity normalization. *BMC Bioinformatics*, 18(11):385, Oct 2017.

[119] Helmut Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 172–176, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[120] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM.

[121] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18(1):198, Mar 2017.

[122] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California, June 2016. Association for Computational Linguistics.

[123] Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. Biomedical event extraction using abstract meaning representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,, August 2017. Association for Computational Linguistics.

[124] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks*, 5(2):157–166, March 1994.

[125] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1310–III–1318. JMLR.org, 2013.

[126] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[127] Alex Graves and Jrgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602 – 610, 2005. IJCNN 2005.

[128] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010.

[129] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 795–798, New York, NY, USA, 2015. ACM.

[130] Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[131] Mahmood Yousefi-Azar and Len Hamey. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93 – 105, 2017.

[132] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[133] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[134] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[135] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[136] Cícero Nogueira Dos Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages 1818–1826. JMLR.org, 2014.

[137] Robert Leaman, Rezarta Islamaj Doan, and Zhiyong Lu. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917, 2013.

[138] Sunil Sahu and Ashish Anand. Recurrent neural network models for disease name recognition using domain invariant features. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2216–2225, Berlin, Germany, August 2016. Association for Computational Linguistics.

[139] Qikang Wei, Tao Chen, Ruifeng Xu, Yulan He, and Lin Gui. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, 2016.

[140] Robert Leaman, Graciela Gonzalez, et al. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, volume 13, pages 652–663. Kohala Coast, Hawaii, USA, 2008.

[141] Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. Cd-rest: a system for extracting chemical-induced disease relation in literature. *Database*, 2016:baw036, 2016.

[142] Zhehuan Zhao, Zhihao Yang, Ling Luo, Lei Wang, Yin Zhang, Hongfei Lin, and Jian Wang. Disease named entity recognition from biomedical literature using a novel convolutional neural network. *BMC medical genomics*, 10(5):73, 2017.

[143] Haodi Li, Buzhou Tang, Qingcai Chen, Kai Chen, Xiaolong Wang, Baohua Wang, and Zhe Wang. Hitsz_cdr: an end-to-end chemical and disease relation extraction system for biocreative v. *Database*, 2016:baw077, 2016.

[144] Hsin-Chun Lee, Yi-Yu Hsu, and Hung-Yu Kao. An enhanced crf-based system for disease name entity recognition and normalization on biocreative v dner task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pages 226–233, 2015.

[145] Allan Peter Davis, Thomas C. Wiegers, Michael C. Rosenstein, and Carolyn J. Mattingly. Medic: a practical disease vocabulary used at the comparative toxicogenomics database. *Database*, 2012:bar065, 2012.

[146] SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. Distributional semantics resources for biomedical text processing. In *Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan*, pages 39–43, 2013.

[147] Hongwei Liu and Yun Xu. A deep learning way for disease name representation and normalization. In Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong, editors, *Natural Language Processing and Chinese Computing*, pages 151–157, Cham, 2018. Springer International Publishing.

[148] Nianwen Xue et al. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48, 2003.

[149] Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. Effective tag set selection in chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pages 87–94. Tsinghua University Press, 2006.

[150] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155. Association for Computational Linguistics, 2009.

[151] Hamada Nayel and H L Shashirekha. Improving ner for clinical texts by ensemble approach using segment representations. In *Proceedings of the 14th Interna-*

tional Conference on Natural Language Processing (ICON-2017), pages 197–204, Kolkata, India, December 2017. NLP Association of India.

[152] H. L. Shashirekha and H. A. Nayel. A comparative study of segment representation for biomedical named entity recognition. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1046–1052, Sept 2016.

[153] Zhenfei Ju, Jian Wang, and Fei Zhu. Named entity recognition from biomedical text using svm. In *5th International Conference on Bioinformatics and Biomedical Engineering, (ICBBE) 2011*, pages 1–4, May 2011.

[154] Ying He and Mehmet Kayaalp. Biological entity recognition with conditional random fields. In *In AMIA Annual Symposium Proceedings*, pages 293–297, 2008.

[155] Yi-Feng Lin, Tzong-Han Tsai, Wen-Chi Chou, Kuen-Pin Wu, Ting-Yi Sung, and Wen-Lian Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2004), August 22th, 2004, Seattle, Washington, USA*, pages 56–61. Springer-Verlag, 2004.

[156] Gurulingappa H, Hofmann-Apitius M, and Fluck J. Concept identification and assertion classification in patient health records. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

[157] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, 46(6):1088 – 1098, 2013.

[158] Zhou GuoDong and Su Jian. Exploring deep knowledge resources in biomedical name recognition. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko, editors, *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 99–102, Geneva, Switzerland, August 28th and 29th 2004. COLING.

[159] Richard Tzong-Han Tsai, Cheng-Lung Sung, Hong-Jie Dai, Hsieh-Chuan Hung, Ting-Yi Sung, and Wen-Lian Hsu. Nerbio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. *BMC Bioinformatics*, 7(S-5), 2006.

[160] Xiao Fu and Sophia Ananiadou. Improving the extraction of clinical concepts from clinical records. *Proceedings of BioTxtM14*, 2014.

[161] Siddhartha Jonnalagadda, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. Enhancing clinical concept extraction with distributional semantics. *Journal of Biomedical Informatics*, 45(1):129 – 140, 2012.

[162] Yonghui Wu, Jun Xu, Min Jiang, Yaoyun Zhang, and Hua Xu. A study of neural word embeddings for named entity recognition in clinical text. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1326. American Medical Informatics Association, 2015.

Appendix A

Complete list of manually written stop words

about	below	fifteen	indeed	now	something	under
above	beside	fiftey	interest	nowhere	sometime	until
across	besides	fill	into	of	sometimes	up
after	between	find	is	off	somewhere	upon
afterwards	beyond	fire	it	often	still	us
again	both	first	its	on	such	very
against	bottom	five	itself	once	system	via
all	but	for	keep	one	take	was
almost	by	former	last	only	ten	we
alone	call	formerly	latter	onto	than	well
along	can	forty	latterly	or	that	were
already	cannot	found	least	other	the	what
also	cant	four	less	others	their	whatever
although	co	from	ltd	otherwise	them	when
always	con	front	made	our	themselves	whence
am	could	full	many	ours	then	whenever
among	couldnt	further	may	ourselves	thence	where
amongst	cry	get	me	out	there	whereafter
amoungst	de	give	meanwhile	over	thereafter	whereas
amount	describe	go	might	own	thereby	whereby
an	detail	had	mill	part	therefore	wherein
and	do	has	mine	per	therein	whereupon
another	done	hasnt	more	perhaps	thereupon	wherever
any	down	have	moreover	please	these	whether
anyhow	due	he	most	put	they	which
anyone	during	hence	mostly	rather	thick	while
anything	each	her	move	re	thin	whither
anyway	eg	here	much	same	third	who
anywhere	eight	hereafter	must	see	this	whoever
are	either	hereby	my	seem	those	whole
around	eleven	herein	myself	seemed	though	whom
as	else	hereupon	name	seeming	three	whose
at	elsewhere	hers	namely	several	through	why
back	empty	herself	neither	she	throughout	will
be	enough	him	never	should	thru	with
became	etc	himself	nevertheless	show	thus	within
because	even	his	next	side	to	without
become	ever	how	nine	since	together	would
becomes	every	however	no	sincere	too	yet
becoming	eg	hundred	nobody	six	top	you
been	everyone	i	none	sixty	towards	your
before	everything	ie	noone	so	twelve	yours
beforehand	everywhere	if	nor	some	twenty	yourself
behind	except	in	not	somehow	two	yourselves
being	few	inc	nothing	someone	un	

Publications

• International Conferences

- H. L. Shashirekha and Hamada A. Nayel: "A Comparative Study of Segment Representation for Biomedical Named Entity Recognition", *In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1046-1052, Sep 2016.

- Hamada A. Nayel and H. L. Shashirekha: "Mangalore-University@INLI-FIRE-2017: Indian Native Language Identification using Support Vector Machines and Ensemble Approach", *In Working notes of Indian Native Language Identification (INLI) task held in conjunction with Forum of Information Retrieval Evaluation (FIRE-2017), IISc, Bangalore, India, 8th - 10th December.*, (**Our submissions securely achieved second and third ranks**)

- Hamada A. Nayel and H. L. Shashirekha: "Improving NER for Clinical Texts by Ensemble Approach using Segment Representations", *In the Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 197-204, NLP Association of India, December-2017, Kolkata, India.

• International Journals

- Hamada A. Nayel, H. L. Shashirekha, Hiroyuki Shindo, Yuji Matsumoto: "Improving Multi-Word Entity Recognition for Biomedical Texts", *International Journal of Pure and Applied Mathematics*, Volume 118(16), pp:301-319, 2018.

- Hamada A. Nayel and H. L. Shashirekha: "A Deep Learning-Based Model for Disease Named Entity Recognition", In Informatics, Multidisciplinary Digital Publishing Institute, (*Communicated*).