Generic Urdu NLP Framework for Text Analysis: Hybridization of Heuristic and Machine Learning Techniques

PhD(CS) Thesis

Submitted by:

Wahab Khan

Reg: No. 72-FBAS/PHDCS/S-12

Supervisor: Dr. Ali Daud
Co-Supervisor: Dr. Jamal Abdul Nasir

Department of Computer Science and Software Engineering, International
Islamic University, Islamabad
2019

1- Computer programme
2- Data base management
3- Information retrieval
4- Natural language processing
5- Machine learning techniques
6- Theses

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF BASIC & APPLIED SCIENCES
DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

Date: 07-02-19

Final Approval

It is certified that we have read this thesis, entitled "**Generic Urdu NLP Framework for Text Analysis: Hybridization of Heuristic and Machine Learning Techniques**" submitted by **Mr. Wahab Khan, Registration No.72-FBAS/PHDCS/S12**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the award of the degree of Doctor of Philosophy in Computer Science.

Committee

External Examiner (i):
Dr. Waseem Shahzad,
Professor / Head of Department
Department of Computer Science, FAST, H-11, Islamabad

External Examiner (ii):
Prof. Dr. Sohail Asghar,
Professor,
COMSATS institute of Information Technology (CIIT), Islamabad.

Internal Examiner
Dr. Qamar Abbas,
Assistant Professor,
Department of Computer Science & Software Engineering
FBAS, IIUI

Supervisor
Dr. Ali Daud,
Associate Professor,
Department of Computer Science & Software Engineering
FBAS, IIUI

Co-Supervisor
Dr. Jamal Abdul Nasir,
Assistant Professor,
Department of Computer Science & Software Engineering
FBAS, IIUI

Chairman:
Dr. Ayyas Husnain,
Chairman,
Department of Computer Science & Software Engineering
FBAS, IIUI

Dean:
Prof. Dr. Muhammad Arshad Zia,
Dean,Faculty of Basic and Applied Sciences,
International Islamic University, Islamabad

This dissertation is submitted to
Faculty of Basic and Applied Science,
Department of Computer Science and Software Engineering,
International Islamic University Islamabad, Pakistan.
In partial fulfillment of the requirement of the degree of
Doctor of Philosophy (Computer Science)

Declaration

I unfeignedly declare that this dissertation, tilted “Generic Urdu NLP framework for Urdu text analysis: hybridization of heuristics and machine learning techniques**”** is a presentment of my germinal research work, has been written by me, neither as a whole nor as part has been plagiarized out from any source. Wherever contributions of others are encumbered, every effort is made to indicate this intelligibly, with anticipated reference work to the literature. It is further declared that I have completed this thesis entirely based on my personal effort, made under the sincere guidance of my supervisors. Some of the incisions therein dissertation curb subject from papers and journals published by me as the first author. The incisions are either reordered or rewritten as per formation of this dissertation.

I also declare that this dissertation reflects my actual research findings and has not been submitted for any previous degree.

Wahab Khan
72-FBAS/PHDCS/S12

Dedication

I am dedicating this dissertation to my parents, for their perpetual love and stark support and whose valuable examples inspired me to work hard for the things that I aspire to achieve. I am esteemed to have you as my parents. I am also dedicating this dissertation to my beloved wife, who look after my kid’s day and night and who also take care of my beloved parents. I am truly thankful to having you my life partner.

WAHAB KHAN
72-FBAS/PHDCS/S12

Acknowledgement

Most importantly, I devote my subservient gratitude for Allah (SWT) who has conferred me the potency and calibers due to which I have could accomplished my research.

I would like to admit the support and contributions of my Supervisor Dr Ali Daud who has all the time been the primary intuition and stirring of my hard work. I am appreciative for his semiprecious ideas, suggestions and experience that propelled me to concluded the research thesis. Special thanks to my Co supervisor Dr Jamal Abdul Nasir IIU, Islamabad, for his valuable suggestion and innovative ideas to complete my PhD research.

wahab khan

72-FBAS/PHDCS/S12

Abstract

The internet was initially designed to present information to users in English. However, with the passage of time and the development of standard web technologies such as browsers, programming languages, libraries, frameworks, databases, front and back-ends, protocols, APIs, and data formats, the internet became a multilingual source of information. In the last few years, the natural language processing (NLP) research community has observed a rapid growth in online multilingual contents. Thus, the NLP community maims to explore monolingual and cross-lingual information retrieval (IR) tasks. Digital online content in Urdu is also currently increasing at a rapid pace. Urdu, the national language of Pakistan and the most widely spoken and understandable language of Indian sub-continent, is considered a low-resources language (Mukund, Srihari, & Peterson, 2010).

Part of speech (POS) tagging and named entity recognition (NER) are considered the most basic NLP tasks. Investigation of these two tasks in Urdu is very hard. POS tagging, the assignment of syntactic categories for words in running text is significant to natural language processing as a preliminary task in applications such as speech processing, information extraction, and others.

Named entity recognition (NER) corresponds to the identification and classification of all proper nouns in texts, and predefined categories, such as persons, locations, organizations, expressions of times, quantities and monetary values, etc. it is considered as a sub-task and/or sub-problem in information extraction (IE) and machine translation. NER is one of the hardest task in Urdu language processing. Previously majority Urdu NER systems are based on machine learning (ML) models. However, the ML model needs sufficiently large annotated corpora for better performance(Das, Ganguly, & Garain, 2017). Urdu is termed as a scared resource language in which sufficiently large annotated corpus for ML models' evaluation is not available. Therefore, the adoption of semi-supervised approach which is largely dependent on usage of the huge amount of unlabeled data is a feasible solution.

In this thesis, we propose a generic Urdu NLP framework for Urdu text analysis based on machine learning (ML) and deep learning approaches. Initially, we addressed POS challenges by developing a novel tagging approach using the linear-chain conditional random fields (CRF). We employed a strong, stable, balanced language-independent and language dependent feature set for Urdu POS task and used the method of *context words window*. Our approach was evaluated against a support vector machine (SVM) technique for Urdu POS - considered as the state of the art - on two benchmark datasets. The results show our CRF approach to improving upon the F-measure of prior attempts by 8.3 to 8.5%.

Secondly, we adopted deep recurrent neural network (DRNN) learning algorithms with various model structures and word embedding as a feature for the task of Urdu named entity recognition and classification. These DRNN models include long short-term memory (LSTM) forward recurrent neural network (RNN), LSTM bi-directional RNN, backpropagation through time (BPTT) forward RNN and BPTT bi-directional RNN. We consider language-dependent features such as part of speech (POS) tags as well as language independent features such as N-grams. Our results show that the proposed DRNN-based approach outperforms existing work that employ CRF based approaches. Our work is the first to use DRNN architecture and word embedding as a feature for Urdu NER task and improves upon prior attempts by 9.5% in the case of maximum margin.

List of Publications

Journal Publications from Thesis

1. Urdu Part of Speech Tagging Using Conditional Random Fields.

   Wahab Khan, Ali Daud, Jamal Abdul Nasir, et al., Language Resources and Evaluation (2018), DOI: https://doi.org/10.1007/s10579-018-9439-6
2. Part of Speech Tagging in Urdu: Comparison of Machine and Deep Learning Approaches

   Wahab Khan, Ali Daud, Khairullah Khan, Jamal Abdul Nasir, et al., IEEE Access, 2019, Volume: 7, Issue:1, 38918-38936, DOI, 10.1109/ACCESS.2019.2897327
3. Deep Recurrent Neural Networks with Word Embeddings for Urdu Named Entity Recognition

   Wahab Khan, Ali Daud, et al., (**Paper accepted in ETRI Journal**)
4. A survey on the state-of-the-art machine learning models in the context of NLP.

   Wahab Khan, Ali Daud, Jamal A. Nasir, and Tehmina Amjad

   Article published in Kuwait journal of Science, vol. 43, pp. 66-84, 2016.
5. Urdu language processing: A survey

   Ali daud, **Wahab Khan** and Che, D

   Article published in *Artificial Intelligence Review (2017)* Vol.47 issue :3, 279-311.

Conference Publication from Thesis

1. Wahab Khan., Daud, A., Nasir, J. A., & Amjad, T. (2016) Urdu Named Entity Dataset for Urdu Named Entity Recognition Task. In proceeding of the 6th International Conference on Language & Technology pp. 51-55.

Table of Contents

Chapter 1 1
Introduction 1
1.1 The Urdu Language 1
1.1.1 Characteristic 3
1.1.2 Orthography (املا, *Imla*) 4
1.1.3 Morphology 4
1.2 Basic Concepts 5
1.2.1 Part of Speech Tagging 5
1.2.2 Named Entity Recognition 6
1.3 Problem Statement 6
1.4 Motivations 7
1.5 Research Objectives and Research Questions 7
1.5.1 Research Questions 7
1.5.2 Research Contribution 8
1.6 Thesis Organization 10
Chapter 2 12
Related Work 12
2.1 POS Related Work 12
2.1.1 Rule-Based Approaches 12
2.1.2 Machine Learning Approaches 13
2.1.3 Hybrid Approaches 14
2.2 NER Related Work 15
2.2.1 Rule-Based Approaches 16
2.2.2 Machine Learning Approaches 17
2.2.3 Hybrid Approaches 18
2.3 Chapter Summary 19
Chapter 3 21
Part of Speech Tagging using CRF 21
3.1 Urdu parts of speech Challenges 22
3.2 Proposed Method 25
3.2.1 Features 25
3.2.2 Linear Chain CRF 25

3.3 Experiments 27
3.3.1 Datasets 27
3.3.2 Training and Testing 29
3.3.3 Performance Evaluation Matrices 33
3.3.4 Results and Discussion 33
3.4 Chapter Summary 40
Chapter 4 41
Named Entity Recognition using DRNN 41
5.1 Challenges to Urdu NER 42
4.2 Proposed Method 45
4.2.1 Features 46
4.2.2 Recurrent Neural Network 50
4.2.3 Deep Recurrent Neural Network 51
4.3 Experiments 52
4.3.1 Datasets 52
4.3.2 Training and Testing 53
4.3.3 Performance Evaluation Measures 56
4.3.4 Results and Discussion 56
4.4 Chapter Summary 64
Chapter 5 66
Urdu Named Entity Dataset 66
5.1 Importance of Tagged corpora 66
5.2 IJCNLP-2008 NE tagged dataset 67
5.3 Jehangir et al Dataset 68
5.4 The UNER Dataset 68
5.4.1 Development 69
5.4.2 Segment Representation Techniques 73
5.5 Comparative analysis of UNER dataset 75
5.4 Chapter Summary 76
Chapter 6 78
Conclusion and Future Work 78
6.1 Conclusion 78
6.2 Future Work 80
References 81
Appendix 88

List of Tables

Chapter 1 1
Chapter 2 12
Chapter 3 21
Table 3-1: Example showing Urdu POS Complexity 22
Table 3-2: Example of POS assignment 23
Table 3-3: Noun used as adjective example 24
Table 3-4: Adjective used as an adverb example 24
Table 3-5: Adjective used as noun example 24
Table 3-6: Description of Proposed Features set 25
Table 3-7: Feature Function Structure 27
Table 3-8: Train file Format 30
Table 3-9: Pseudocode of the Proposed CRF based POS system using CRFSharp libraries .. 31
Table 3-10: Average Precision, Recall and F_Measure of Proposed and baseline approach .. 33
Table 3-11: Average precision, recall & f-measure of individual 34
Table 3-12: Average precision, recall & f-measure of individual POS tags of BJ Dataset 35
Table 3-13: Portion of the baseline approach confusion matrix on BJ dataset 37
Table 3-14: Portion of proposed approach confusion matrix on BJ dataset 37
Table 3-15:Concatenated output of CRF on CLE dataset for CC Tag 39
Table 3-16: Concatenated output of the CRF model on CLE Dataset for NNP and NN tags . 39
Table 3-17: Summary of Most confused tags along with its 40
Chapter 4 41
Table 4-1: Various named entities and its description 42
Table 4-2: Spelling Variation of Urdu person names 45
Table 4-3: Template Feature Set 47
Table 4-4: Cooccurrence Matrix 49
Table 4-5:Word Vector 49
Table 4-6: Pseudocode of the Proposed LSTM RNN model for NER task using RNNSharp libraries 52
Table 4-7: Training File Format 55
Table 4-8: Average Precision, Recall & F-Measure of Proposed and Baseline approaches on IJCNLP-2008 Dataset. 57
Table 4-9: Average precision, Recall & F-Measure of proposed and baseline model 58
Table 4-10: Average precision, Recall & F-Measure of proposed and baseline model on UNER News dataset 59
Table 4-11: Entity Wise Statistics of average precision, recall and F-Measure of baseline and LSTM forward approach 60
Table 4-12: Entity Wise Statistics of average precision, recall, and F-Measure of the proposed approach on IJCNLP-2008 dataset 61
Table 4-13: Average Precision, Recall and f-Measure of Person, Location and Organization in UNER and Jahangir et al datasets 61
Table 4-14: Average Precision, Recall and f-Measure of Title and Number in UNER and Jahangir et al datasets 62

Table 4-15: Average Precision, Recall and f-Measure of Date and Time in UNER and Jahangir et al datasets..62
Table 4-16: Test Data NE Statistics..63
Table 4-17: Confusion Matrix of BPTT Bi-Direction RNN on International News Domain of UNER dataset...63
Table 4-18: Original Statistics of multiple tags acquiring word in original Test data.............64
Table 4-19: Statistics of multiple tags acquiring word after Testing.......................................64
Chapter 5..66
Table 5-1: Consolidated Statistics of UNER dataset 71
Table 5-2: Domain wise consolidated statistics of each entity class 71
Table 5-3: List of Generic Urdu Named Entity Types with the kind of Entities they refer. ..72
Table 5-4: Domain wise No. of Documents 72
Table 5-5: Example of various NE tags..72
Table 5-6: Example of Various SR approaches in the task of NER 74
Table 5-7: Comparative statistics of UNER dataset with IJCNLP and Jehangir et al datasets75
Table 5-8: Domain wise statistics of UNER dataset...75
Table 5-9: Domain wise Entity distribution of UNER dataset 76
Table 5-10: Entity wise statistics of each dataset 76
Chapter 6..78
Table 6-1:Tag wise Confusion matrix of SVM model on BJ dataset.......................................88
Table 6-2: Tag wise Confusion matrix of CRF model on BJ dataset.......................................90
Table 6-3: Tag wise confusion matrix of CRF model on CLE dataset....................................92

List of Figures

Figure 1-1: Screenshot of Urdu POS tagged Text 5
Figure 1-2: Screenshot of Urdu Text containing NER Tags 6
Figure 1-3: Overall Proposed System Overview 9
Figure 1-4:Schematic Representation of Thesis Organization 10
Figure 3-1:Schematic Representation of CLE POS Tagset 28
Figure 3-2Detail of of tagset used in BJ 28
Figure 3-3:Schematic Representation of Sajjad POS tagset 29
Figure 3-4: Overall POS proposed system overview 32
Figure 3-5: Graphical Representation of Average Precision, Recall and F_Measure 33
Figure 3-6: Graphical Representation of Average f-Measure of individual POS tags of CRF and SVM Model on CLE Dataset 36
Figure 3-7:Graphical Representation of Average f-Measure of individual 36
Figure 4-1: Schematic Representation of the Proposed Deep learning based Urdu NER system 46
Figure 4-2: Graphical Representation of DRNN Training and Testing Process 56
Figure 4-3: Graphical Representation of Jahangir et al dataset results 58
Figure 4-4:Graphical Representation of Average precision, Recall & F-measure of Proposed and Baseline approach on IJCNLP dataset 59
Figure 4-5: Graphical Representation of Average Precision, Recall and F-Measure of proposed and baseline model on UNER News dataset 60
Figure 5-1:Screenshot of IJCNLP-2008 dataset before preprocessing 68
Figure 5-2: Screenshot of IJCNLP dataset 68
Figure 5-3: Screenshot of National News Domain 70
Figure 5-4: Screenshot of International News Domain 71
Figure 5-5: Screenshot of Sports News Domain 71
Figure 5-6: Graphical Representation of UNER Dataset 73

List of Abbreviations

ACL	Association for computational linguistics
ANN	Artificial neural network
BPTT	Back propagation through time
CES	Corpus encoding scheme
CLE	Centre for language engineering
CNN	Convolutional neural network
CRF	Conditional random field
DL	Deep Learning
DRNN	Deep recurrent neural network
DTD	Document Type Definition
EMILLE	Enabling Minority Language Engineering
IE	Information extraction
IR	Information retrieval
LSTM	Long term short term memory
ML	Machine learning
NER	Named entity recognition
NLP	Natural language processing
NN	Neural network
POS	Part of speech tagging
RNN	Recurrent neural network
SR	Segment representations
SVM	Support vector machine
TBL	Transformation based learning
ULP	Urdu Language processing
UNER	Urdu named entity recognition dataset

About this Chapter

This chapter encircles basic concepts regarding Urdu grammar, feature set used, dataset annotation and the motivations of this study. In addition, objectives and research questions are presented along with the contributions of this work.

Introduction

The language wreaks a very significant role in the interpreting of the culture and mentality of societies. In the presence of enough linguistic data, automatic text analysis provides us an opportunity to understanding many insights of a community (Mukund et al., 2010).

In the last few years, multilingual contents on cyberspace expanded speedily. This phenomenon pulled ahead researcher from the NLP research community to explore monolingual and cross-lingual IR task (Daud, Khan, & Che, 2016). In the beginning, the web was dominated by English digital text, but it subsequently turned over to multilingual contents. The IR task in which both the subject information only retrieved as well the query used, are in the same language is termed as monolingual IR, while the IR task in which the subject information accessed and the query used, belongs to different languages is termed as cross-lingual IR (Capstick et al., 2000).

Currently research activities in South Asian languages including Urdu are in full action. Nowadays Urdu is mainly investigated from multiple perspectives. Urdu has a very rich morphology compared to other South Asian languages, and this feature makes it a favorite choice for researchers in the Asian language processing research community (Mukund et al., 2010).

To analyze the mindset of a society, robust and accurate NLP system is paramount necessary (Hovy & Spruit, 2016). NLP systems developed for English as well as for other Western languages have recommendable accuracies. On the other hand, Urdu lags far behind regarding the availability of robust and accurate NLP systems.

1.1 The Urdu Language

Urdu is a major stakeholder in South Asian languages (Lai & Surood, 2013). Since the independence of Pakistan and India Urdu is arousing in popularity. Contrary to Arabic and Persian, Urdu belongs to Indo-Aryan languages which is closely related to Hindi. Not only in South Asia but there exist millions of urdu speakers around the globe.(Riaz, 2010).

The total number of mother tongues spoken in Pakistan exceed sixty, among them only Urdu is listed as national language of the country in constitution of Pakistan (Rahman, 1997). In Pakistan, the total number of Urdu speakers whose mother tongue is Urdu are about 8 %. Nationwide, about 75% Pakistanis quickly understand Urdu and are using it as a communication medium with each other (Rahman, 1997). In Pakistan it is used in a wide range of fields such as media of instruction in government schools, the primary language in journalism, poetry and much more. In Pakistan, the estimated number of people speaking Urdu are more than 11 million and around the globe, it is more than 300 million (Riaz, 2010). Most Urdu speakers are found in the province of Sindh while Punjab is the second leading province. In Punjab, not only there exist a significant number of Urdu speakers, but it is also the main center of Urdu literature publication.

Urdu is also a widely spoken language in India. In India, Urdu is declared as an official language of six states, and it is also included in the recognized languages list of India by the constitution of India. In India, the estimated number of Urdu native speakers are around 52 million, and the total number of its speakers in both Pakistan and India exceeds 65 million (Hussain, 2008; Riaz, 2008). Leading countries in which Urdu speaker exist in high ratio are Pakistan, India, USA, Gulf countries, and Canada.

Most Urdu words are either derived or loaned from Persian and Arabic. Besides this Urdu also have similarities with most South Asian languages. For example, Urdu does not support capitalization; same is the case with Hindi, Arabic, and Persian. Similarly, like other Asian languages there is no concept of small and capital letters in Urdu.

Urdu and Hindi are akin in structural similarity, the difference between Urdu and Hindi exist in vocabulary and writing scripts. Urdu follows Nastaliq script while Hindi is based on Devanagari script. Contrast to Hindi, Urdu is more sophisticated as its morphology and syntax structure is the hybridization of some languages such as Persian, English, Arabic, and Turkish (Adeeba & Hussain, 2011).

In the past few years, the rapid growth of multilingual content on the cyberspace was observed. This phenomenon motivated researchers from NLP domain to explore the multilingual information retrieval task(Daud et al., 2016).

These days South Asian languages are engaged with much research involvement from researchers, Especially, Urdu is leading comparatively to others (Mukund et al., 2010).

Urdu lacks sophisticated tools for POS and NER, which are paramount necessary to develop a standard NLP system (Riaz, 2010). English and other English-like languages have plentiful standard NLP tool with high results comparatively to Urdu (Adeeba & Hussain, 2011; Al-Shammari, 2008; Bushra Jawaid & Ahmed, 2009)

1.1.1 Characteristic

Urdu is Arabic script-based language, the most important aspect of Arabic script is its context sensitivity which means that the shape of each letter depends on the position of its occurrence. Urdu is both morphological and inflectional rich language, richness in morphology in that sense that one word may consist of several morphemes while inflectional in the sense that Urdu derive its maximum vocabulary from other morphological rich languages such as Persian, English, and Turkish and many others.

Automated NLP for Urdu is the stat-of-the-art demand of current age as Urdu is termed as rich morphological language, bears unique nature and also there exist millions of its speakers throughout the world. Due to its rich morphology and inflectional nature, it always remained in the mainstream of literature writing, particularly for poetry writings (Jahangir, Anwar, Bajwa, & Wang, 2012; Riaz, 2010).

In automated Urdu NLP, there are a number of challenges but limited resources, the existence of huge number of derived words, the occurrence of single words in desperate spellings, the existence of nested entities, conjunction ambiguity, and the free word order characteristic of Urdu are the major one(Riaz, 2010; Singh, Goyal, & Lehal, 2012).

Urdu and Hindi are two live vernacular of Hindustani and are conceived very closely related to each other. The major aspects in which the two languages are interlinked with each other are morphology, phonology and syntactic structure. Regarding syntax, both are SOV (subject, object, and verb) based language. Since Urdu and Hindi are akin to each other in many aspects, therefore we can assume that NLP tools and systems developed for one language might easily work interchangeably, but actually this assumption inapplicable(Riaz, 2010). Dissimilarities in the script, the difference in vocabulary, lack of diacritics in Hindi are major points which restrict the adoption of tools being developed for one language for other (Riaz, 2010). Though both languages are dissimilar, by the use of some standard tools such as translator, some resources can be shared.

1.1.2 Orthography (املا, *Imla*)

Orthography is concerned with the legal writing system of a language. The objective of this field of study is to analyze the writing and to explore how letters of alphabet blend to comprise voices and shape words.

Urdu orthography suffers from both standardization and unification. A single word can be written in various spelling. Consequently one can notice basic phrases wrote differently in different publications and it can be obscuring and discouraging for students.

کے لیے(Kayliay)	کیلئے(Kayliay)
اس کا(iska)	اسکا(iska)
ہسپتال(Hospital)	اسپتال(Aaspatal)

Urdu is an Arabic script-based language. In Arabic script, a word formulation based on the combination of several letters or a single character forms the whole word. The most important property of these languages is context sensitivity, such that the shape of the current character is dependent on its neighbor character. This language follows right to left text writing architecture.

These days the most widely adopted style for writings in Arabic script-based languages including Urdu is Nastaliq. In Nastaliq style a latter can appear in various positions. These positions are isolated, middle, center or ending character. While writing in Nastaliq style and character joining occurs automatically while some character moves up and backward.

1.1.3 **Morphology**

In natural language processing, Morphology is concerned with a word analysis. This field of study answers that how words are developed, it also investigates kinship among words in the same language. In Urdu existence of a single word in multiple forms is very common and this makes Urdu a morphologically rich language.

As far as morphology of Urdu is concerned, like other Indo-European languages it also demonstrates a concatenative inflective morphological model. Though most of Urdu morphology is concatenative for example, the causative formation by vowel lengthening in, for example, “mar vs. “maar” or “nikal” vs. “nikaal” does not represent concatenative morphology. Reduplication also has unusual morphological properties.

- **Nouns**

Nouns in Urdu bears two genders: the masculine and feminine. Nouns may accept grammatical gender postfix (marking), or possibly unmarked. Nouns are also modulated to demonstrate quantity and case. By Marking Urdu nouns can be put into two groups. First, all those nouns which experience acquit gender postfix are referred to as marked nouns while all those nouns which lack the gender suffix are referred to as unmarked nouns.

- **Verbs**

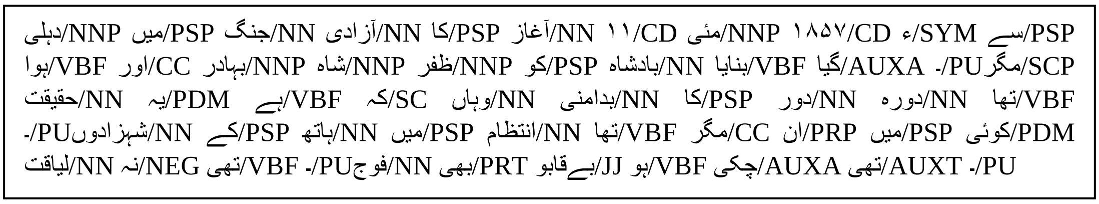
Verb represents happening or execution of any activity. Verbs can be transitive and intransitive; transitive verbs are based on the need of direct object while intransitive verbs do not require an object. In most languages, the verb stem relates to the radical morpheme that shows the significance of the verb. The verb stems typically acquire affixes. This form usually the most basic form which comes up in dictionaries. In languages such as Urdu, this task can be quite complicated.

Urdu verbs have four basic forms: The root, imperfective participle, perfective participle, and infinitive (Platts et al., 1920; Schmidt, 2005). These are detailed with auxiliaries and suffixes into a complex system of rules of verb tense and expression. The basic forms of a verb determine its expressions, whereas auxiliaries determine its tense.

1.2 Basic Concepts

1.2.1 Part of Speech Tagging

POS is considered one of the most basic NLP tasks. The objective of this task is to assign a syntactic category the part of speech to each word of the running text. Either can accomplish the assignment by applying rules, machine learning approaches or by hybridization of rules and machine learning approaches. POS plays a significant role in natural language processing as a preliminary task in applications areas such as speech processing, information extraction, and others. The core application areas which take advantage from the POS tagging task includes: speech recognition, text to speech, word sense disambiguation, information retrieval, semantic processing, parsing, information extraction and automatic machine translation (Anwar, Wang, Li, & Wang, 2007; Daud et al., 2016; Roth & Zelenko, 1998). The following example given in Figure 1-1 describes Urdu POS task.



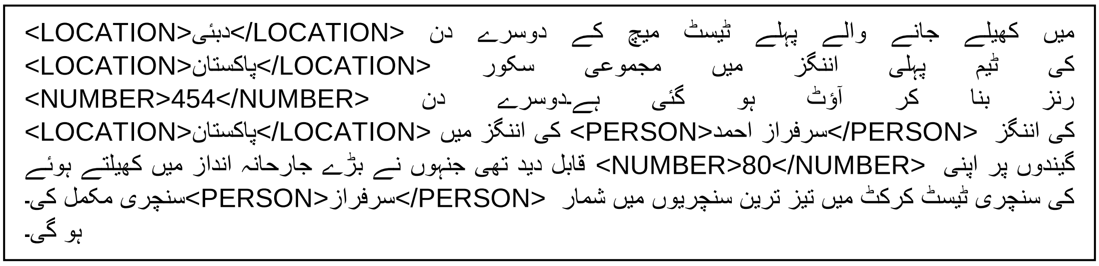
دہلی/NNP میں/PSP جنگ/NN آزادی/NN کا/PSP آغاز/NN ۱۱/CD مئی/NNP ۱۸۵۷/CD ء/SYM سے/PSP
ہوا/VBF اور/CC بہادر/NNP شاہ/NNP ظفر/NNP کو/PSP بادشاہ/NN بنایا/VBF گیا/AUXA ۔/PUمگر/SCP
حقیقت/NN یہ/PDM ہے/VBF کہ/SC وہاں/NN بدامنی/NN کا/PSP دور/NN دورہ/NN تھا/VBF
۔/PUشہزادوں/NN کے/PSP ہاتھ/NN میں/PSP انتظام/NN تھا/VBF مگر/CC ان/PRP میں/PSP کوئی/PDM
لیاقت/NN نہ/NEG تھی/VBF ۔/PUفوج/NN بھی/PRT بےقابو/JJ ہو/VBF چکی/AUXA تھی/AUXT ۔/PU

Figure 1-1: Screenshot of Urdu POS tagged Text

1.2.2 Named Entity Recognition

NER, known variously as `entity identification’, `entity chunking’ or `entity extraction’, is among the most basic of NLP tasks. The NER task is considered to be a sequential labeling task, corresponding to the identification and classification of proper nouns into predefined categories such as persons, locations, organizations, expressions of times, quantities, monetary values, etc. (Sundheim, 1996). It is considered to be an essential preliminary step in common NLP tasks such as question answering, conversation, voice search, etc. (Lu, Li, & Xu, 2015). As such, NER plays a vital role in the management and extraction of intelligent information from the text (Seok, Song, Park, Kim, & Kim, 2016).

The example given in Figure 1-2 describes Urdu text containing named entity class labels.



میں کھیلے جانے والے پہلے ٹیسٹ میچ کے دوسرے دن <LOCATION>دبئی</LOCATION>
کی ٹیم پہلی اننگز میں مجموعی سکور <LOCATION>پاکستان</LOCATION>
رنز بنا کر آؤٹ ہو گئی ہے۔دوسرے دن <NUMBER>454</NUMBER>
کی اننگز <PERSON>سرفراز احمد</PERSON> کی اننگز میں <LOCATION>پاکستان</LOCATION>
گیندوں پر اپنی <NUMBER>80</NUMBER> قابل دید تھی جنہوں نے بڑے جارحانہ انداز میں کھیلتے ہوئے
کی سنچری ٹیسٹ کرکٹ میں تیز ترین سنچریوں میں شمار <PERSON>سرفراز</PERSON>سنچری مکمل کی۔
ہو گی۔

Figure 1-2: Screenshot of Urdu Text containing NER Tags

1.3 Problem Statement

The goal of this study is to develop and propose an effective NLP framework for Urdu text analysis, initially surmounting the two most important NLP tasks namely part of speech tagging and named entity recognition. Both POS and NER are considered an integral component of several other natural language processing (NLP) task to carry effective text analysis. But in spite of its significant importance to other high level tasks, the ULP community not only lacks linguistic resources but also is dealing with a language without capitalization, unlike NLP in a European language. Generally, POS and NER are conceived a challenging task and a bit of challenges are required to be handled in each language including Urdu. However, the characteristics and peculiarities of Urdu cause accosting with POS and NER a difficult. POS and NER in Urdu is, therefore, a difficult task, demanding a greater sophistication in linguistic analysis and the development of techniques for effective task performance. In response to this problem, our study proposes (a) to investigate POS task with CRF, a most popular machine learning model for sequential labeling with both language dependent and independent feature set and (b) to undertake DRRN model along with the word embedding as a feature for Urdu NER.

1.4 Motivations

The world wide web has overturned the information processing system and communication spectrum like nothing before. Initially, the cyberspace was dominated by English text but with the passage of time and ease of access to online resources, the web turned to multilingual contents. Consequently, Urdu text has also experienced exploded growth in cyberspace.

To understand the mindset of a society and to mine relevant information, sophisticated and robust NER and POS tool development is state-of-the-art need of modern era, as the tasks of machine translation and information extraction are majorly dependent on accurate POS and NER systems. The contemporary period has witnessed the intense development of machine learning techniques, often as state-of-the-art approaches to solving problems such as POS tagging and NER. The main reason for extensive usage is based on four features: a) capability of automatic learning b) the degree of accuracy c) the speed of processing and d) generic nature. From literature survey, we observed, though Urdu - spoken in a vast area of the Indian sub-continent – is an example of a low-resource language that is yet to attract any significant body of work from machine learning perspectives. About Urdu, English, and English like western languages benefit from an extensive body of research covering various NLP tasks, and thus, benefit from abundant resources for computational analysis such as gold standard data. Notable such resources include annotated NER datasets, WordNet, dictionaries, gazetteers, and related tools most being readily available due to being open-source.

1.5 Research Objectives and Research Questions

In this research work, we reported the development of a generic Urdu NLP framework based on machine learning and deep learning (DL) approaches for carrying computational text analysis in Urdu. Since Urdu is termed as resource scare language, due to lack of linguistic resources to carry supervised learning based research in it. Therefore, to promote automated research in Urdu, especially with machine learning perspectives, we also report the development of monolingual tagged named entity dataset.

1.5.1 Research Questions

This research study primary focus on the following research questions

1. How to improve upon the state-of-the-art POS approaches with the use of conditional random fields (CRF)?
2. How to improve upon the state of the art NER approaches with the use of RNN?
3. How to undertake word embedding as feature for Urdu NER?

1.5.2 Research Contribution

This dissertation earmarked me to bring in galore contributions to the Urdu NLP research community, particularly to the NER, POS field and in Urdu linguistic resources.

In this research work, we reported the development of Urdu NLP Framework covering POS and NER tasks, the two most basic NLP tasks. Our framework is based on the hybridization of machine learning and deep learning approaches. We solved the POS task with CRF while NER task with the help of deep recurrent neural network (DRNN) with LSTM and BPTT algorithms along with its two extensions, the Forward and Bi-directional. Below task wise contributions are provided. We also reported the development of new Urdu NER dataset termed as UNER.

- **POS Contributions**

1. The ever first adoption of CRF as learning algorithm for Urdu POS tagging task
2. A novel feature set with strong, stable and balanced language independent features.
3. To highlight various challenges which make Urdu POS task crucial.

- **NER Contributions**

1. The ever first employment of RNN architecture for Urdu NER on benchmark datasets
2. A systematic evaluation of DRNN models with CRF based baselines on three datasets
3. A novel feature set for Urdu NER that includes template features, word embedding features, context features and Runtime features

- **Dataset Contributions**

1. we reported the development of ever first sizeable NE tagged dataset for automated NER research in Urdu, especially with the ML perspective

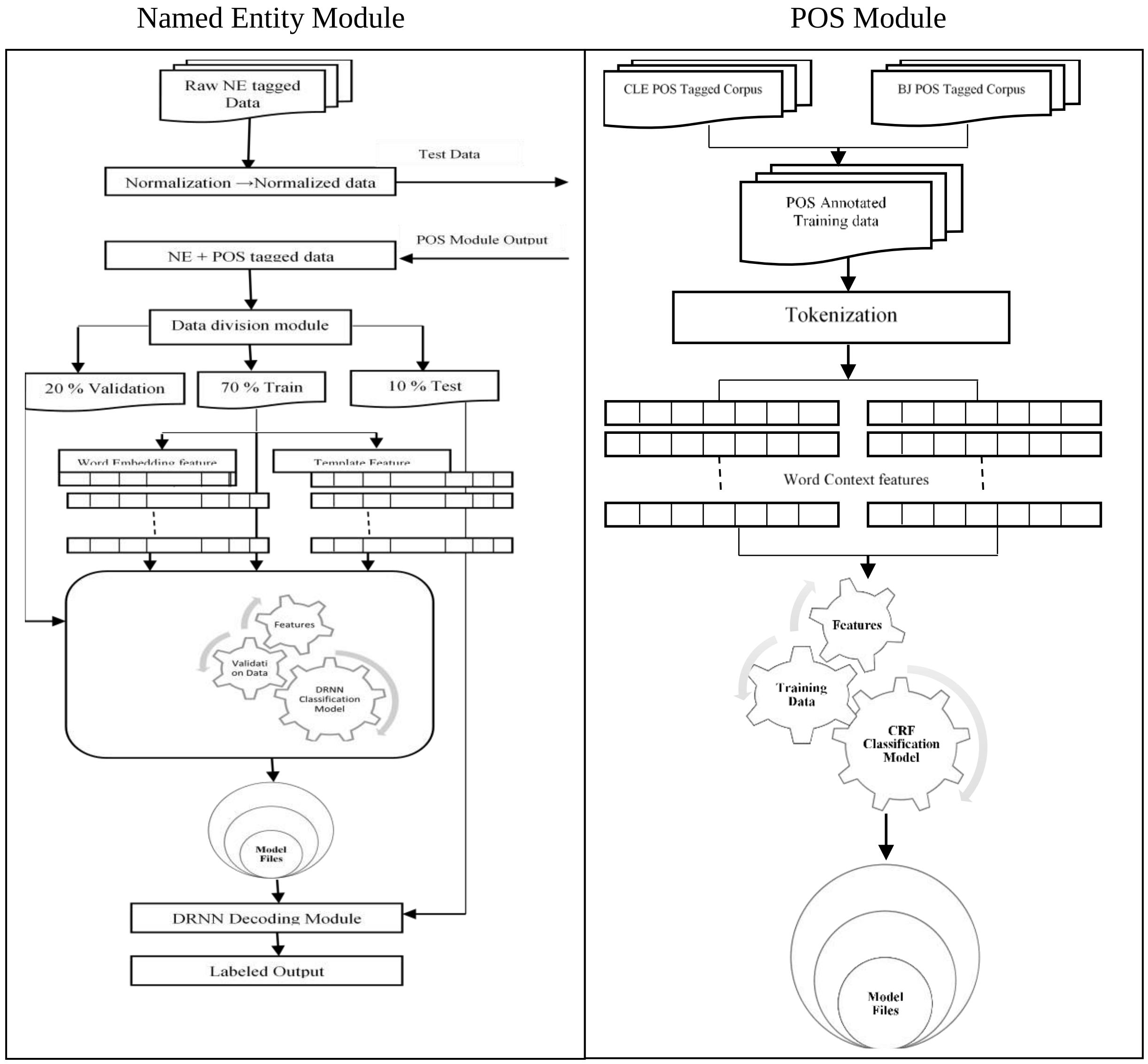


Figure 1-3: Overall Proposed System Overview

1.6 Thesis Organization

Topics covered in this thesis are organized in six chapters (see Figure 1-4)

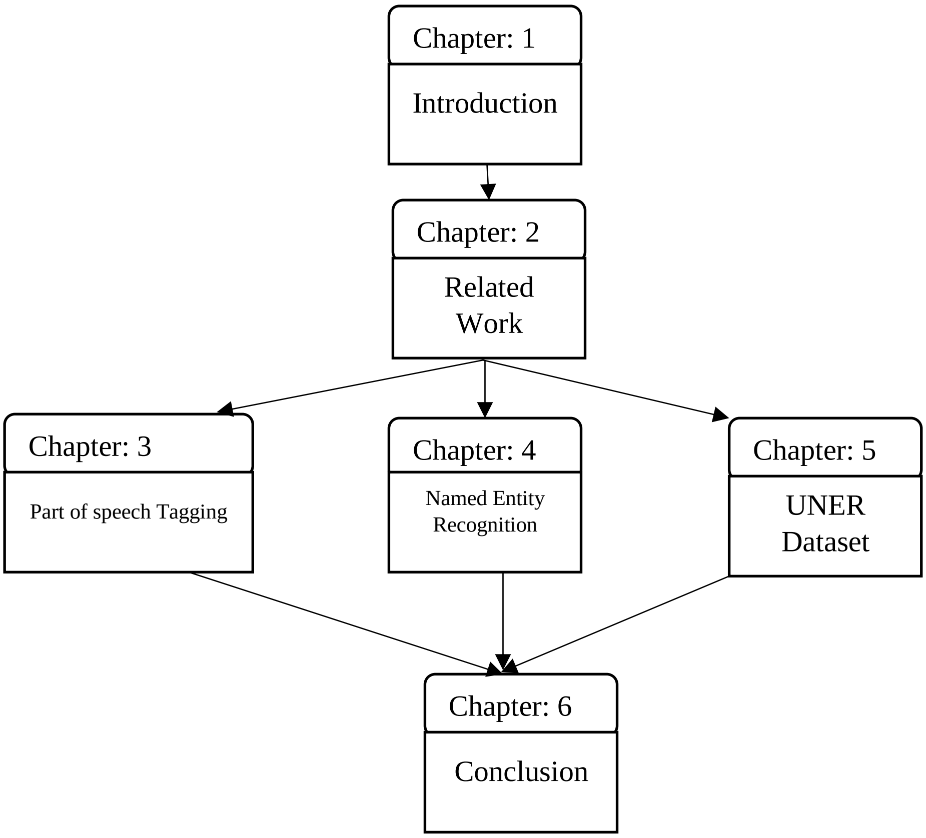


Figure 1-4:Schematic Representation of Thesis Organization

Chapter 2 provides consolidated literature survey of POS and NER tasks. The focus in this chapter was to discuss the need and demand of machine learning and deep learning based Urdu NLP, to highlight the importance of automatic Urdu NLP, to explore significance of Urdu POS and NER task in Urdu IR applications and to report state of the art work in progress in this domain. Chapter 3 covers first module of the proposed Urdu NLP framework, e.g., development of Urdu POS tagger based on CRF. Here we demonstrated the impact of our proposed feature set on the performance of CRF for Urdu POS task experimentally by comparing it with SVM of baseline. Chapter 4 includes second part of our framework, e.g., development of Urdu NER tagger based on deep learning. In this chapter, we presented the adoption of deep recurrent neural network model and word embedding as a feature for Urdu named entity task and performed a careful comparison of the standard RNN architectures with CRF and ANN. Chapter 5 describes our NER dataset development contribution. The main objective of this cahpter is to provide detailed information about the newly created NER dataset and detailed description of various dataset used for performance evaluation of our proposed

POS and NER models. And in the end chapter 6 will conclude ou research work and provide us with future directions.

About this Chapter

This chapter provides a comprehensive literature review of the two modules of the proposed framework. Initially literature review regarding Urdu POS is provided and after that literature review of Urdu NER discussed

Related Work

These days, South Asian countries like Western countries have equal opportunities of accessing the cyberspace resources e.g computer applications, social media, etc. In most South Asian countries cyberspace resources are on hand and user-friendly to be exercised it in almost every walk of life. Much of these web resources, interface as well as text processing technologies, are based on English and its grammar. Since Urdu has larger number of speakers in South Asia but unfortunately, sophisticated Urdu NLP systems are hardly available. This phenomenon attracted researchers from Urdu language processing research community to focus on Urdu language processing.

2.1 POS Related Work

From resources availability perspective, Urdu is termed as low resources language. Core linguistic resources such as corpora, tagsets , WordNet and much more are not available at finger tips when compared with Western languages, if any, Then it has license restriction (Daud et al., 2016). So far Urdu tagsets are concerned, the ULP research community is limited to only three available POS tagsets. Prior approaches to Urdu POS tagging are generally either rule-based, machine learning/statistical, or hybrid approaches.

2.1.1 Rule-Based Approaches

The first known POS tagset for Urdu, the ‘Hardie tagset’ (Hardie, 2003), consists of 350 tags in which 48 subcategories are defined for nouns and 115 for verbs. Given this dataset contains many tags, NLP tasks that employ it face a number of challenges like morphosyntactic ambiguity and much more.

The first such approach by Hardie (Hardie, 2003), which can be understood as a foundational resource for Urdu POS tagging, was rule-based, and resulted in the formulation of a tagset for Urdu employing the EAGLES heuristics for morpho-syntactic annotation of dataset and the grammatical rules of Urdu as defined by Schmidt(Schmidt, 1999). The EAGLES heuristics was primarily scripted for European Union (EU) languages. Due to its generic nature, the same can be adopted for Urdu language with little effort, due to Urdu’s morphological resemblances with Indo-European language class. Hardie’s tagger was based on about 270 linguistic rules and reached an accuracy of 90% (Hardie, 2003). The shortcoming of Hardie approach is that it is based on 350 tagset and training and testing with small dataset and large tagset adversely affect performance. Moreover, construction of rules for such a huge tagset is laborious task.

Rule-based approaches suffer from lack of potency and manageability (Chiong & Wei, 2006), and in addition, require new rules to be specified where new information to be added to the training domain. Furthermore, to incorporate rules a specific task, one must have sound knowledge of that particular language as well as expertise in rule synthetization. In addition, the rules being developed tend to be area specific and do not applied to other domains. Finally, rule-based approaches suffer from a significantly longer development time.

In contrast, popular contemporary approaches for POS in most languages are based on supervised learning approaches, which currently constitute an intensely active research area, especially with respect to big-data contexts. Supervised learning models work by automatically inducing rules from pre-labelled data, i.e. training data. The machine learns through input datasets or using complex logics to produce the desired output (Khan, Daud, Nasir, & Amjad, 2016). Machine learning based POS frameworks are preferred because they are versatile and updatable within a relatively short time frame and require less processing where sufficient amount of training data is available (Daud et al., 2016).

2.1.2 Machine Learning Approaches

In machine learning category the Urdu POS work of (Anwar et al., 2007) is considered as pioneering work. Anwar et al. (2007) proposed the first statistical approach for Urdu POS task. Their tagger is based on an n-gram Markov model and its performance similar to existing taggers. They evaluate their tagger using the EMILE corpus for training and testing, and adopting two sorts of tagsets, one with more than 250 tags and the other being their modified tagset containing 90 tags, They used unigram, bigram and Backoff models with both tagsets. The best accuracy was 95%, and this was with the smaller tagset and the Backoff model. The main limitation of Anwar et al. (2007) is that they claimed the use of two tagsets but did not revealed the description of the tagsets used in their study. Similarly, the authors also used very limited context information and suffix information is also not provided.

Among the prima limitations of HMM is that all state only bring forth a local observation. So one observation devolves on entirely a single state instantly while indirectly is dependent on other states. Another problem with which HMM suffers is the existance of inconsistency between learning and predictive objective functions. HMM learninig process is based on joint probability distribution P(Y,X) over states and observation, however for prediction taks conditional probability P(Y|X) is required

Sajjad and Schmid (2009) compared the performance of four taggers for Urdu POS tagging task: Tree Tagger, RF tagger, TnT tagger and SVM model-4 tagger(Giménez & Marquez, 2004). Their evaluations employed a corpus of around 110000 items collected from the web, with a tagset of 42 syntactic categories. Almost all four taggers produced good accuracy. However, the SVM tagger exhibited superior performance at 95.66% accuracy.

Bushra Jawaid and Ondřej (2012) proposed an SVM with a voting-scheme approach for the Urdu POS task. The authors experimentally demonstrated that by combining the existing methods and linguistic resources available for Urdu, the Urdu tagger performance progresses to a higher degree. Authors compared their proposed SVM approach with a morphological analyzer and existing Urdu parser. For comparison, they converted the output of baseline approach to a common format by using a unified tagset. Their evaluations used POS tagged data as training data obtained from the CRULP. They chose SVM model-4 out of the five available SVM models and achieved an accuracy of 87.98%.

Bushra Jawaid, Kamran, and Bojar (2014) extending the work in (15), crawled sizeable Urdu text from cyberspace and performed automatic POS tagging using SVM. On test data, the authors proposed a standalone POS tagger which achieved the accuracy of 88.74%.

Since, in the research work of Sajjad and Schmid (2009) , Bushra Jawaid and Ondřej (2012) and Bushra Jawaid et al. (2014) the best results are reported with SVM. However, training time of VM is extremely slow (Antony & Soman, 2010).Since SVM is kernel based model while kernel models can be quite sensitive to over-fitting the model selection criterion (Cawley & Talbot, 2010).Additionaly, SVMs do not perform advantageously for multiclass classification problem.

2.1.3 Hybrid Approaches

F. Naz, Anwar, Bajwa, and Munir (2012) adopted Brill’s transformation-based learning (TBL) approach for Urdu POS task. TBL is a kind of hybrid approach which makes use of both, the rules and the statistical models. Brill’s TBL approach has the potency of automatic rules generation from training data. The authors evaluated their proposed approach on training data of size 123755 words, annotated with tagset of size 36 tags and reported an accuracy of around 84%.

The main disadvantages of TBL approach are: its learning process is greedy by nature; therefore, the discovered rule sequence could not be best. Similarly, it is a non probalistic modle and, therefore, can not return more than one result (Brill, 1995). Additionally, due to its non probalistic nature it is also unable to measure confidence of results (Florian, Henderson, & Ngai, 2000).

Recently, Abbas (2014) presented a semi-semantic technique for POS annotation. The dataset used in their experiment was made by crawling Wikipedia for text and is manually annotated with Urdu POS tags. The tagset used for annotation consists of 22 tags and the total size of the dataset used is 1400 sentences. They used Krippendorff's 'α' measure as evaluation for Urdu KON-TB treebank developed for Urdu and reported an accuracy of 96.4%.

2.2 NER Related Work

NER techniques were initially studied in the Message Understanding Conferences (MUCs)(Sundheim, 1996) initiated by DARPA. The main objective of this project was to aid in the development of information extraction techniques, and grew rapidly from thereon as they became employed in a wide range of systems (Daud et al., 2016; Nadeau & Sekine, 2007; Roberts, Gaizauskas, Hepple, & Guo, 2008). The majority of these systems dealt with European languages (Tjong Kim Sang & De Meulder, 2003), especially English, wherein they achieved a mature status with respect to their effectiveness. Machine learning based NER frameworks have been proposed for non-European languages such as Arabic(Alotaibi & Lee, 2014), Persian and South Asian languages (Shaalan & Raza, 2009; Singh et al., 2012). For Urdu, however, NER systems are at an initial stage (Mukund et al., 2010). The most mature systems, such as those for English, depend heavily on extrinsic linguistic resources such as annotated corpora, human-made dictionaries, and gazetteers to improve accuracy (Daud et al., 2016; Kazama & Torisawa, 2007). The Urdu NLP research community, however, currently lack these resources, and furthermore, face difficulty – unlike the European languages NLP – of dealing with a language that does not support Capitalization. As a result, the Urdu NER task requires a greater degree of sophistication with respect to scientific analysis and the techniques employed for effective task performance.

The long tradition of NER research for English from the early 1990s(Sundheim, 1996), has exhibited various techniques, from rule-based techniques (Rau, 1991) , to purely supervised techniques (Bikel, Miller, Schwartz, & Weischedel, 1997; Borthwick, Sterling, Agichtein, &

Grishman, 1998; McCallum & Li, 2003) and hybrid approaches (Collins & Singer, 1999; Nadeau & Sekine, 2007). NER schemes built for particular domains do not usually transfer effectively to others (Daud et al., 2016; Poibeau & Kosseim, 2001). Thus, techniques developed for European language traditions are not immediately applicable to the relatively new ULP context. Techniques for Urdu NER, however, still divide in the same general classes: (a) rule-based (b) machine learning (ML) and (c) hybrid.

2.2.1 Rule-Based Approaches

Rule-based methods, e.g., (Riaz, 2010; Singh et al., 2012b), use a collection of handmade rules – i.e. grammar, affixes look-up and lexicon look-up - designed for each class of named entity, which is then applied to a given text to extract named entities. A rule-based algorithm first searches the named entity, compares it with predefined rules, and once the rule is matched, assigns the entity the corresponding classification, i.e. the output. The auspicious aspect of systems formulated by applying the linguistic rule supported approach guarantees improved accuracy (Daud et al., 2016; Riaz, 2010; Singh et al., 2012). Rule-based methods, however, generally lack potency and manageability (Chiong & Wei, 2006). This is due to several reasons, (a) they need to be continuously updated with new rules as the corresponding domain changes, (b) adding rules for some specific task requires knowledge of the corresponding language as well expertise in rule synthetization, (c) rules developed for one language generally do not port to other language domains, and finally (d) rule-based techniques suffer from a much longer development time than other techniques.

Riaz (2010) presents the first instance of a rule-based algorithm for Urdu NER. The algorithm considers six named entities classes in offline plain Urdu text: proper (human) names, territory or state names, organizations, numbers, date, and designations. It is evaluated on the Becker-Riaz dataset, and achieved an f-measure value, precision and recall of 91.1 per cent, 91.5 per cent and 90.7 per cent respectively. Though Riaz study presents very good overall results but unfortunately individual entity results are not provided. Similarly, it also lacks details regarding that how the proposed system handles nested entities.

Singh et al. (2012) presented a rule-based approach that, instead, recognizes thirteen entities in offline plain Urdu text as compared to the twelve tags in IJCNLP-2008. The rules were evaluated through test data constructed mainly from new sources, divided into two sets of 12032 and 150243 tokens respectively. An accuracy of 74.09 per cent was achieved for all thirteen tags. Though Singh et al. (2012) considered thirteen named entity classes however the detailed description of the thirteen named entity classes is missing. Similarly, rules are not described, nested named entities are avoided, for experiments gold standard datasets are not used. Finally rules creation for thirteen entities are a very tedious job.

2.2.2 Machine Learning Approaches

These days, leading contemporary approaches for NER - in most languages – are, instead, based on ML approaches. The recent trend of analyzing large datasets with supervised learning approaches (Khan et al., 2016; Oudah & Shaalan, 2016) is particularly indicative of the overall inclination towards ML approaches. Supervised models work by inducing rules from pre-labeled data, known as training data, and correspond to parametric, non-parametric, or kernel-based learning algorithms; or to algorithms that employ logic (Khan et al., 2016; Silver et al., 2016).

ML-based NER frameworks are, generally, more versatile than rule-based methods. If training data is readily available, an ML approach can adapt to changing domains with very limited effort (Shaalan, 2014). Furthermore, semi-supervised ML methods using a bootstrapping learning scheme (Thenmalar, Balaji, & Geetha, 2015) are particularly effective in the absence of large annotated corpora. Semi-supervised methods require an initial seed model built, using a small initial token set with predefined categories. This is used to classify tokens in the text, whereupon classifications attributed with high confidence are fed back into the training data. This process, then, iterates over the remaining tokens.

The lack of prior work in Urdu NER, especially prior ML approaches, is partly due to the low interest from the NLP community and partly to the scarcity of resources for computational analysis. The first significant works and techniques on Urdu NER were exhibited at the 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)1 which aimed at investigating NER for five South Asian languages including Urdu, using ML techniques. Since that pioneering conference, several works have to-date explored ML approaches to Urdu NER.

Malik and Sarwar (2015) adopt a CRF based approach and evaluate it on the benchmark IJCNLP-2008 NER dataset with three named entity types e.g. Person, Organization, and Location. They limit to language-independent features such as context words, reported precision, recall, and f -measure values are 63.72 percent, 62.30 percent, and 63.00 percent.

Ekbal, Haque, Das, Poka, and Bandyopadhyay (2008) is considered the ever first work to investigate NER on five widely spoken Indian languages - Urdu, Hindi, Oriya, Bengali, and Telugu – with CRF models. Evaluations with the IJCNLP-2008 dataset placed the F-measure

[1] http://ltrc.iiit.ac.in/nlp-lpl-08/

value for Urdu at 35.52 percent, and at 59.39 percent, 28.71 percent and 4.74 percent for Bengali, Oriya, and Telugu respectively.

Mukund and Srihari (2009) adopt a CRF based approach and demonstrate that NE labeling can be enhanced by expanding the set of historical data for part of speech (POS) learning followed by subjecting data to bootstrapping procedures. Their model achieved F-Score of 68.9 percent on CRULP and CRL datasets. These outcomes are taken to be the state of the art for statistical approaches to Urdu NER task.

Mukund et al. (2010) also apply a CRF approach but use both language independent and language dependent features such as POS. They train and test their algorithm on two datasets; one for POS learning and one for NE learning. Evaluations were done for three main named entity classes person, location, and organization. Results, upon ten-fold cross-validation are reported as 68.89 percent for F-measure value with the maximum being 69.21 per cent. Furthermore, experimental evidence is shown for improvements due to use of heuristics, with improved values of F-measure for the worst test set being from 68.89 percent to 74.67 percent and for the best from 69.21 percent to 71.3 percent.

Since the research works of Malik and Sarwar (2015), Ekbal, Haque, Das, et al. (2008) Mukund and Srihari (2009) and Mukund et al. (2010) are all based on CRF model. However, all machine learning models are heavily dependent on the existence of a large number of annotated training data for better performance(Das et al., 2017). Similarly, all the previous approaches just considered three named entity classes.

Jahangir et al. (2012) developed unigram and bigram models that use a gazetteer list and employ smoothing algorithms in the bigram approach. Five named entity classes were considered: person, location, organization, date and time. The CRL dataset was used for training and testing, with precision, recall and f-measure values for the unigram approach using gazetteer lists reported at 65.21 percent, 88.63 percent and 75.14 percent, and at 66.20 percent, 88.18 percent and 75.83 percent for the bigram approach using gazetteer lists and Backoff smoothing. The main limitation of N-gram Markov models is its data sparseness and users are required to apply additional algorithms to handle it.

2.2.3 Hybrid Approaches

Hybrid methods usually combine both rule-based and ML methods (Oudah & Shaalan, 2012). Kumar Saha, Sarathi Ghosh, Sarkar, and Mitra (2008) have presented a hybrid NER system for named entities identification in five languages namely Hindi, Bengali, Telugu, Oriya, and

Urdu. Their proposed hybrid approach makes use of linguistic rule along with the Maximum Entropy model. They also used gazetteer lists for performance improvement. The F-measure values reported for the aforementioned five languages are 65.13, 65.96, 44.65, 18.74 and 35.47 percent respectively. Since the authors used maximum entropy model along with rules, however, maximum entropy models suffer from label bias problem and the authors did not mention that how they resolved the label bias problem of maximum entropy.

The hybrid system proposed by Gali, Surana, Vaidya, Shishtla, and Sharma (2008) makes use of handcrafted rules along with conditional random fields. In the absence of enough training data, their system yielded 43.46 percent F-Measure.

Kumar and Kiran (2008) proposed a hybrid NER system. Their proposed hybrid system was based on the joint use of CRF and HMM (Hidden Markov Model) model along with handmade heuristics. The authors evaluated the performance of their proposed hybrid approach on the IJCNLP-2008 workshop dataset. The hybrid approach based on rules and HMM reported 44.73 percent F-measure for Urdu while 38.25 percent after using hybrid CRF.

2.3 Chapter Summary

In this chapter, we provided a literature survey of Urdu POS and NER tasks. From Urdu POS literature survey, it came in our knowledge that very little work was reported from ULP research community to address Urdu POS challenges and SVM model is considered as state of the art for Urdu POS. We also observed that till date CRF based work for Urdu POS do not exist, though it has reported state of the art results in diverse sequential labeling tasks. Therefore, in this thesis, we reported the ever first adoption of the linear-chain conditional random field as a learning algorithm for Urdu POS tagging task with strong, stable and balanced language-independent and language dependent feature set.

Similarly, from a literature review of NER, it came in our knowledge that to date, there has no research work has been reported from the ULP research community in which the researchers have examined the effect of deep learning models and word embeddings of training data for Urdu NER. Therefore, this gap, the limitation of rule-based approaches, advantages of deep learning architecture in diverse areas has motivated us to undertake DRNN model along with the word embedding as a feature for Urdu NER in this study.

Similarly, from literature review we also observed that Urdu suffers from both, lacks linguistic resources as well as standard tools and applications. Since POS and NER tasks play a very vital role in the development of the standard IR application, also considerable advancement has been reported for English as well as many other Arabic script languages such as Farsi, Arabic. But unfortunately, research work regarding Urdu POS and NER in the context Urdu IR is still in infancy stage.

About this Chapter

This chapter explores the first contribution namely the Urdu POS module of our proposed framework in this study. The main contents covered are: (A) introduce POS Task (B) Challenges explored in detail (C) Proposed model presented (D) the presented feature set explained (E) datasets used are discussed and finally results are presented.

Part of Speech Tagging using CRF

Part of speech tagging, the assignment of syntactic categories for words in running text, is significant to natural language processing as a preliminary task in applications such as speech processing, information extraction, and others. POS tagging tasks employ taggers composed of various linguistic rules that assign a syntactic tag to each word in a given text (Brill, 1995). Evaluation of such taggers, as with NLP systems in general means to investigate the efficiency and effectiveness of the association patterns the system derives, such as the association between tags and text, given initial training data. A mature NLP system is that which can produce more robust results with limited resources.

In Urdu language processing, POS tagging is conceived as crucial and complex task. There may be several reasons, but morphosyntactic ambiguity or in other words, the dual behavior of various Urdu POS tags in various situations is the prime one. ML-based POS research for English and other Western languages has a long tradition and a significant amount of work has been done to solve POS problems in these languages. On the other hand, automated Urdu POS task lags far behind in terms of machine learning. In the last few years, machine learning model such as CRF has shown remarkable performance in dealing with various sorts of linguistic ambiguities (Khan et al., 2016). CRF is decisive for resolving POS ambiguity as well as capturing every kind of linguistic knowledge (Daniel & James, 2009; Khan et al., 2016). Avant-garde Urdu POS approaches that are proposed in the literature, depending on statistical machine learning models or exclusively on hand writeen linguistic rules. The performance of machine learning/statistical models for POS tagging mainly depend on the domain of the training set, the tagset used for annotation and the size of the dataset(Daud et al., 2016; Khan et al., 2016; Mukund, 2012). POS taggers require two main resources to work, (a) a gold standard pre-tagged dataset for training and (b) an algorithm for tag generation and assignment (Biemann, 2006; Daud et al., 2016). This makes POS tagging a difficult task for languages such as Urdu that have limited pre-tagged datasets (Daud et al., 2016; Graça, Ganchev, Coheur, Pereira, & Taskar, 2011).

The core application areas which take advantage from POS tagging task includes speech recognition, text to speech, word sense disambiguation, information retrieval, semantic processing, parsing, information extraction and automatic machine translation (Anwar et al., 2007; Daud et al., 2016; Roth & Zelenko, 1998).

3.1 Urdu parts of speech Challenges

Urdu grammar, like Arabic grammar, classifies words into three parts of speech: (1) اسم *(ism, noun*) (2) فعل (*fil, verb*) (3) حرف (*harf, particles*)(Platts, 1874).

The POS has been one of the well-established and largely investigated task in Western languages as compared with Eastern counterparts particularly South Asian languages. Generally, POS has conceived a challenging task and a bit of challenge requires to be handled in each language including South Asian languages. The task of POS is more challenging in Urdu as it experiences more complex inflexional morphology than English (Atwell, 2008), resulting in a much larger tagset and owing to the larger quantity of tags it is difficult to arrive at an eminent accuracy with a limited sized corpus (Muaz, Ali, & Hussain, 2009).

The most predominant errors in machine-controlled POS systems occur in the case of noun prediction for noun based phrases. For example, such phrases where nouns are used as adjectives or adjectives are used as nouns. Then in such cases, one usually requires employing contextual information to make progress. The context of a word in a phrase is thereby used to decide an appropriate tag, however, as in Urdu like languages, automatic word context engineering from a machine learning perspective is considered a difficult task(Daud et al., 2016; F. Naz et al., 2012). For example, consider the following Urdu sentence given in Table 3-1:

Table 3-1: Example showing Urdu POS Complexity

ہے	کھانا	کھانا	نے	میں
hey	khana	khana	ney	main
I have to eat				

In the above example, the term کھانا(Khana) appeared twice with a unique meaning and unique tags, and it neither belongs to Urdu compound words and nor to reduplication words. So far as the appropriate tags are concerned it acquires noun tag at first occurrence in the sentence while a verb tag at the second occurrence. Consider the below sentence given in Table 3-2.

Table 3-2: Example of POS assignment

ہے	کھانا	کھانا	نے	میں
hey	khana	khana	ney	main
PU	VBF	NN	PSP	PRP
I have to eat				

We tested the above sentence with CLE online statistical POS tagger[2]. The CLE tagger confused verb tag with noun tag and generated the following output when the two terms کھانا(Khana), کھانا(Khana) are separated by a blank space.

میں/PRP نے/PSP کھانا/NN کھانا/NN ہے/VBF

Therefore, in such cases selection of an appropriate POS tag is based on context information which is not very easy in case of machine learning.

The most common challenges in Urdu POS tagging task are listed below:

- **Suffixation**

In Urdu, there exist some words which can be used as isolated words as well as the suffix of other words. For example, the word ناک(Naak, Nose) can be used as a separate word where it acquires the noun tag while using as a suffix in word خطرناک (khatar-naak, dangerous) it acquires the adjective tag(Sajjad & Schmid, 2009). This behavior of suffixation might likewise commove the learning process of the machine-controlled tagger and step-up the ambiguity for the tagger.

- **Nouns used as adjectives**

Nouns describing measure, quantity, and price may behave like adjectives, and also precede the nouns they qualify. It is customary for Urdu adjectives to behave as if they were nouns, in noun based phrases. In fact, if the nouns were to be removed, the preceding adjectives would behave like nouns (Daud et al., 2016). For example, consider sentence 1 of Table 3-3 where adjective (ضرورت مند (Zaroratmand, needy)) and noun (لوگوں (logun, people)) occur sequentially. In sentence two, the main noun (لوگوں (logun, people)) is removed, now the adjective (ضرورت مند (Zaroratmand, needy)) behaves like a noun instead of an adjective(Sajjad & Schmid, 2009).

[2] http://182.180.102.251:8080/tag/

Table 3-3: Noun used as adjective example

Sentence No.1	دو خوراک کو لوگوں ضرورتمند do khurak ko logun Zaroratmand Give food to needy people
Sentence No.2	دو خوراک کو ضرورتمند do khurak ko Zaroratmand Give food to the needy

- **Adjectives used as adverbs**

Some Urdu adjectives can be used adverbially to modify other adjectives. In sentence 1 below (Table 3-4), the word بڑا(bara, very) is used as an adverb modifying 'intelligent' while in sentence 2 it is used as an adjective modifying 'man'.

Table 3-4: Adjective used as an adverb example

Sentence No.1	ہے آدمی ذہین بڑا وہ hey admi zahin bada wo He is a very intelligent man
Sentence No.2	ہے آدمی بڑا وہ hey admi bada wo He is a great man

- **Adjectives used as nouns**

Urdu admits numerous adjectives that can be used as nouns. In sentence one below (Table 3-5) the word امیر (ameer, rich) is used as adjective while it is used as a noun in sentence two. Furthermore, loan words such as نوجوان (nowjwan, young man) and غیر ملکی(ghair mulki, foreigners) are classified as belonging both to the category of nouns and adjectives.

Table 3-5: Adjective used as noun example

Sentence 1	ہے	امیر	احمد								
	hey	ameer	ahmad								
	Ahmad is rich										
Sentence 2	ہیں	یکساں	کیلئے	دونوں	غریب	اور	امیر	نظام	عدالتی	کا	پاکستان
	Hain	Yaksan	Kelliey	Dono	Gharib	Aur	Ameer	Nizam	Adalti	Ka	Pakistan
		The judicial system of Pakistan treats both the rich and the poor equally									

3.2 Proposed Method

3.2.1 Features

Model performance in machine learning tasks is strongly dependent on the feature set. As such, effective feature set development is essential to all machine learning models. For CRFs, feature templates are used to produce feature sets from training and test files. Our evaluation used seven unigram templates for feature generation. The template file is organized row-wise, such that each row represents a single template. Each template comprises a prefix, identification number, and rule-string. The prefix is used to indicate template type, "U" is used for a unigram template while "B" for bigram features. An identification number is used to differentiate the templates. Rule-string is used to guide CRF to generate features. The encoding process thereby works to generate (then store) a feature set from the records in the training corpus. Details of the features template file used during the training and testing phase are given in Table 3-6.

Table 3-6: Description of Proposed Features set

Feature Template	Description
U01: %x [-1,0]	Previous lexical word
U02: %x [0,0]	Current lexical word
U03: %x [1,0]	Next lexical word
U04: %x [0,0] %x [-1,0]	Current lexical word + Previous lexical word
U05: %x [0,0] %x [1,0]	Current lexical word + Next lexical word
U06: %x [0,0] %x [-1,0] %x [-2,0]	Current lexical word + N-1 and N-2 previous words
U07: %x [0,0] %x [1,0] %x [2,0]	Current lexical word + N+1 and N+2 next words
U08: %x [0,1]	Part of speech tag of a previous lexical word
U09: %x [0,2]	Suffix of current lexical word
U10: %x [0,3]	Length of current lexical word

3.2.2 Linear Chain CRF

CRFs (Lafferty, McCallum, & Pereira, 2001) are one of the most common sequential models, widely used for segmentation and labelling tasks in NLP and have reported state-of-the-art results. In the training phase, it can handle efficiently a large number of multiple, overlapping, and independent features(Žitnik, Šubelj, & Bajec, 2014).

CRFs are probabilistic models for tagging and segmenting sequential data. They support a conditional approach and structurally follow characteristics of an undirected graphical model, where the nodes represent the label sequence 'y' corresponding to the observation sequence 'x'. CRF model aims at finding the label 'y' which maximizes the conditional probability p(y|x) for a sequence x (Benajiba & Rosso, 2008). CRF model is a feature-based model where features can have both real and binary values. In the case of our experiments, all features are binary valued. When applying CRFs to the POS problem, an observation sequence (x) is a token of a sentence or document of text and the state sequence (y) is its corresponding label sequence (Ekbal, Haque, & Bandyopadhyay, 2008)

Formally, the conditional probability for each input sequence $X = (x_1, x_2, x_3 \dots \dots . x_n)$ and tag sequence $Y = (y_1, y_2, xy_3 \dots \dots . y_n)$ of a CRF model can be expressed as follows(Lafferty et al., 2001):

$$P(y|x) = \frac{1}{Zx} exp\left(\sum_{e\epsilon E}\sum_{i} \lambda_i \, t_i(e, y|_e, x) + \sum_{v\epsilon V}\sum_{k} \mu_k \, s_k(v, y|_v, x)\right) \qquad (3\text{-}1)$$

Z(x) is a normalization factor which can be expressed as:

$$Z(x) = \sum_{y}\left(exp\left(\sum_{e\epsilon E}\sum_{i} \lambda_i \, t_i(e, y|_e, x) + \sum_{v\epsilon V}\sum_{k} \mu_k \, s_k(v, y|_v, x)\right)\right) \qquad (3\text{-}2)$$

The parameters (λ) in CRF can be estimated using iterative scaling algorithm, gradient-based methods or L-BFGS methods (Song, Zhang, & Huang, 2017)., However, in this study, the libraries used for CRF are based on the L-BFGS method for parameter estimation.

- **CRF Feature Function**

Feature functions are core supplements of the CRF training phase and are generated according to the mentioned features. Final features are synthesized by using feature function by going through the entire training and testing data. CRFs can use both real-valued as well as binary-valued features. However, in our experimentation, all features are binary valued.

$$f_k(y_{i-1}, y_i, x, i) = \begin{cases} 1, & if\; y_{i-1} = \; ADJECTIVE\; and\; y_i = \; NOUN \\ 0, & \qquad\qquad Otherwise \end{cases}$$

The number of final features generated by feature function during encoding process ranges from thousands to several hundred thousands or even in millions, mainly depending on the (a) size of training data, (b) number of output classes and (c) the number of distinct string expended from a given template. In this study, the number of context word window feature used is 7 while the number of output classes are 35 in case of CLE POS tagset. Thus, in case of the unigram template, the number of feature functions generated by a feature template for a single token will be ((1x7) x35=245). Similarly, for a record having total of 12 tokens, the number of features generated will be ((12x7) x35=2,940)). In our experiments, when a CRF evaluates any feature e.g. “The word to the left of the current word”, for any token e.g. the token انگریزی

(Angrazee, English) then the set of feature functions generated will be like those shown inTable 3-7:

Table 3-7: Feature Function Structure

func1 = if (output class = PNN and feature e.g. The word to the left of the current word = “پڑھنے”) return 1 else return 0
func2 = if (output class = NN and feature e.g. The word to the left of the current word = “پڑھنے”) return 1 else return 0
....
funcN = if (output = QM and feature e.g. The word to the left of the current word = “پڑھنے”) return 1 else return 0

3.3 Experiments

All ten-fold cross-validation experiments are conducted on 2.6 GHz Intel Core i7 PC with 16 GB of RAM.

3.3.1 Datasets

We evaluated the effectiveness of our proposed approach on two benchmark datasets, the CLE dataset[3] and Bushra Jawaid (BJ) dataset(Bushra Jawaid et al., 2014). The latter is the first sizeable Urdu dataset freely available while the CLE dataset has a license restriction.

- **CLE Dataset**

Our CRF model was evaluated on the *Urdu Digest* POS Tagged dataset released by the Center for Language Engineering (CLE) for research and computational processing in Urdu. The CLE dataset contains 100 K Urdu words from various areas, such as politics, health, education, world affairs, business-related data, humor, sports, and literature. Creators of this dataset put data in two main classes namely (1) The Informational (80%) and (2) Imaginative (20%). The major domains from which the informational portion is built are: politics, health, education, international affairs, entertainment, and science. Similarly, the imaginative portion contains text from sources like book reviews, novels, translation of foreign literature and short stories. The only restriction of the CLE POS labeled dataset is the requirement of a license.

[3] http://www.cle.org.pk/

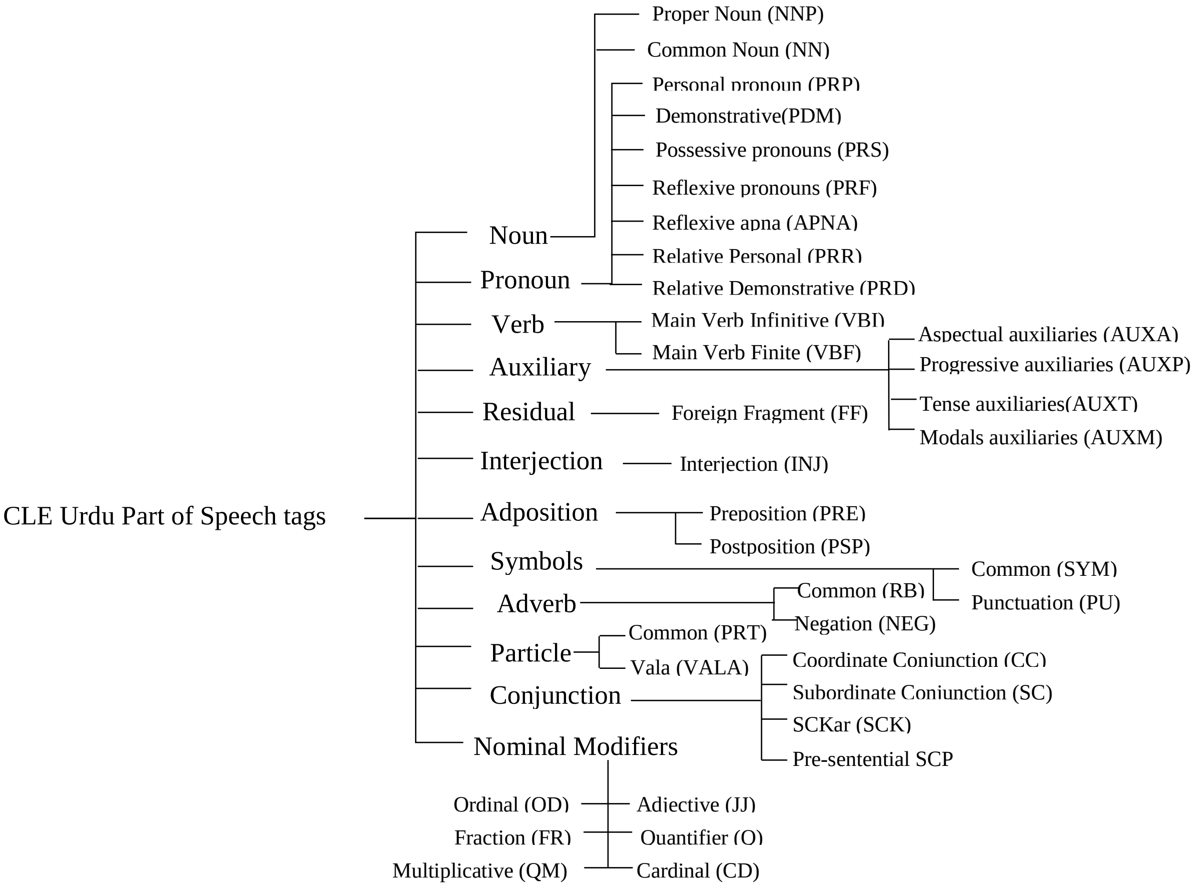


Figure 3-1:Schematic Representation of CLE POS Tagset

Tagsets and word disambiguation rules are fundamental parts of any POS tagger. Primarily tags in CLE Urdu POS tagset are put in twelve core categories with subdivisions, resulting in 35 unique POS tags(Tafseer et al., 2015). The CLE tagset has been used to tag 100k words of the CLE Urdu Digest Corpus(Tafseer et al., 2015). The various tags used in CLE Urdu Digest dataset annotation are graphically represented Figure 3-1.

- **Bushra Jawaid Dataset**

Bushra Jawaid et al. (2014) released the first sizeable (95.4 million words), freely available, Urdu POS tagged dataset. The dataset, containing 95.4 million words distributed in about 5.4 million sentences, was constructed through crawling various Urdu website e.g. BBC Urdu and Urdu Planet and automatically annotated with POS tags.

Owing to its size and our limited hardware resources, we worked, instead, with a 200k subset of data. We pre-processed the Bushra Jawaid Dataset, and in pre-processing, we deleted "," from the whole corpus as SVM support CSV file format. e.g. 54,64,574 was transformed to 5464574.

The tagset used in the annotation of Bushra Jawaid dataset is graphically in Figure 3-3.

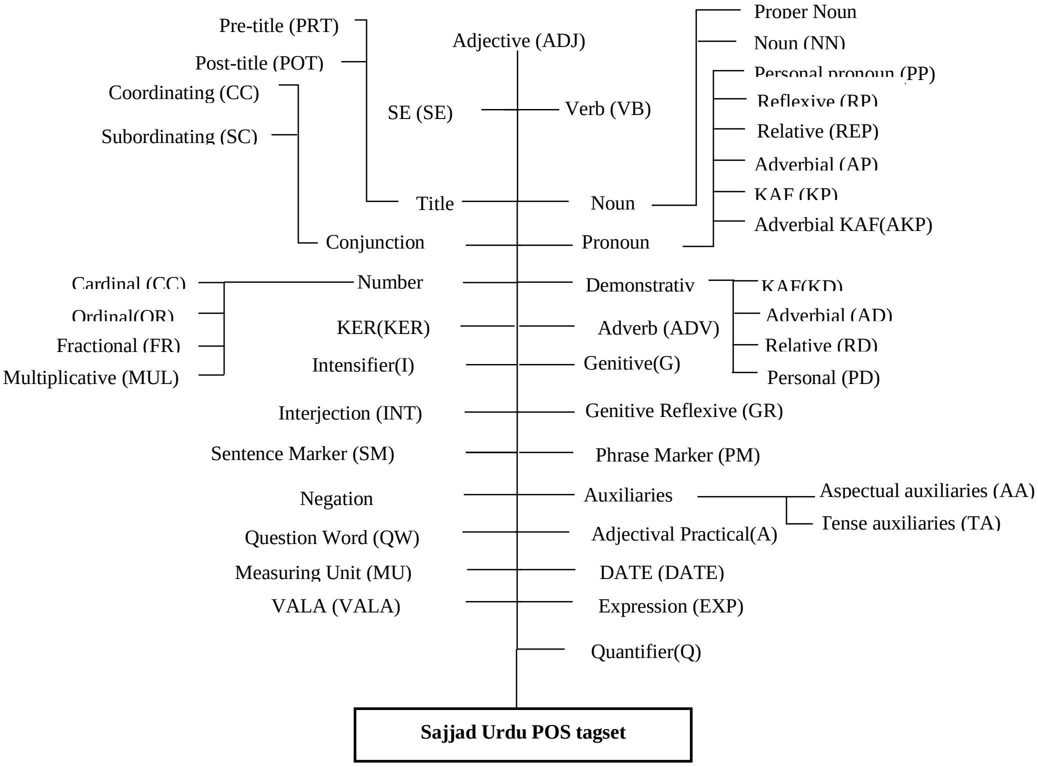


Figure 3-3:Schematic Representation of Sajjad POS tagset

3.3.2 Training and Testing

Our approach applied the CRF approach for part of speech tag labeling and prediction. We used the c-sharp based package *CRFSharp* and used the SVM_Text_Classifier[4] package for the baseline approach. These two packages are open source packages. CRFSharp is a new CRF package and we believe that it is more flexible than the current state-of-the-art. While prior works have used CRFSharp for western languages for similar tasks (segmentation, POS tagging, named entity recognition and so on), its adoption for the same task in South Asian languages including Urdu is non-existent.

The main reason behind using CRFSharp is its model parameters encoding capability by L-BFGS. Moreover, it benefits from many significant improvements over other CRF packages like CRF++[5], such as totally parallel encoding, optimized memory usage, and N-best result generation. In the context of training on a large data set with numerous tags, CRFSharp can make full use of multi-core CPUs with efficient usage of memory compared to CRF++. Hence,

[4] https://github.com/alexandrekow/svmtutorial
[5] http://crfpp.googlecode.com/svn/trunk/doc/in-dex.html

in the same environment and configuration, CRFSharp is capable of encoding models that are relatively more complex, and at a cost lesser than CRF++.

Encoding generates encoded feature values iteratively from training corpus using different feature templates before storing it. Data in a training file is organized record-wise, such that each sentence represents a single record. In a training file, rows or records are separated via an empty line. Each record is divided into its constituent units or tokens along with its feature vector. We represented our training file in a matrix form (see Table 3-8), where each row describes one token and its features, and each column represents a feature in one dimension.

The example given in Table 3-8 depicts a typical training file which consists of one record and each token has two columns showing how different records are labeled. The first column is the term of a token and the second column is to describe a token part of speech tag and its type. The first column is input data for the encoding model, and the second column is the model's ideal output as an answer. In our case, the proposed language independent features or contextual features are pertaining in the training phase includes the N-2 previous and N+2 next words (±2 word). After training CRF on training data through the proposed feature set, a trained classification model is produced.

Table 3-8: Train file Format

Token	**POS tag of the previous token**	**Suffix**	**Word Length**	**Tag (Class type)**
اس	PSP	س	2	PRP
نے	PRP	ے	2	PSP
کلام پاک	PSP	پاک	6	NNP
پڑھا	NNP	ڑھا	4	VBF
-	VBF	-	1	PU

In the decoding phase, the encoded model file generated during the encoding phase is decoded using template features to predict POS tags for each word of the testing file. Test files have a similar format as the training file. The only difference that distinguishes a test file from a training file is the exclusion of class label column from test file.

Pseudocode of the proposed POS system is given Table 3-9:

Table 3-9: Pseudocode of the Proposed CRF based POS system using CRFSharp libraries

Encoding Phase

```
CRFInput_POS (string [Train file, Feature template, Train mode] Args)
    for (a = 0; a < args.Length; a++)
        Load Arguments
    End for
    If mode is train
        Load Train File
        while (Train_file.End == false)
            Read line of train file
            for (b = 0; a < Template File; b++)
                Loading feature template from {0}
                Generate feature set
                Filter out features whose frequency is less than {0}
                Save feature in model file
            End for
        End while
```

The time complexity of the training process O(mNTQ2nS)
(1) where:m is the number of training iterations, N is the number of training data sequences T is the average length of training sequences Q is the number of class labels, n is the number of CRF features, S is the searching time of the optimization algorithm (for example, L-BFGS algorithm, which is considered good for this).

Decoding Phase

```
CRFoutput_POS (string [Test file, Model_file, Test mode,Result] Args)
    for (a = 0; a < args.Length; a++)
        Load Arguments
    End for
    If mode is Test
        Load model from file
        while (test file.End == false)
            Read line of test file
            while (model file.End == false)
                Read feature set for each record
                Create decoder tagger instance
                Initialize result
                Call CRFSharp wrapper to predict given string's tags
                Save segmented tagged result into a file
                Output raw result with probability
                show the best result and its probability
                Show the probability of all tags
                Save the result in the result file
            End while
        End while
```

The steps of the proposed system are as follows:

1. Initially both CLE and BJ datasets are preprocessed for unnecessary text. In preprocessing the unnecessary text such as XML tags, special characters, emojis are removed. Also organized data in sentences format. Words are splited based on the space character.

2. After preprocessing data is split into test and train portion of 10-fold cross validation experiments format.
3. Training data is modeled in the required format of the libraries used.
4. Features are extracted in the encoding phase as per the context features used
5. Features are saved in model files
6. In decoding phase test data is tested against the feature model file generated in the encoding phase
7. The labled output is generated.
8. Evaluation of the test data using precision, recall, and f-Measure
9. Average results are produced.

Figure 3-4 shows an overall overview of our proposed POS system,

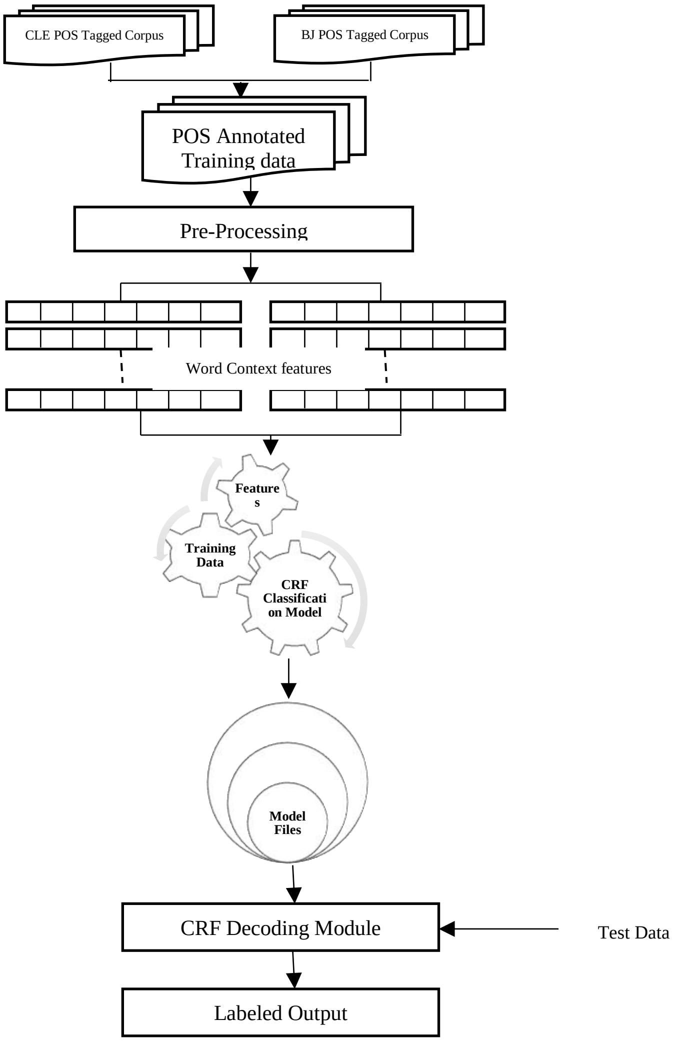


Figure 3-4: Overall POS proposed system overview

3.3.3 Performance Evaluation Matrices

In this thesis, for performance evaluation of our proposed systems, we used precision, recall, and f-measure matrices.

3.3.4 Results and Discussion

We use Precision, Recall, and F-measure metrics to measure the performance of our system. Our experiments proposed the SVM model (Bushra Jawaid et al., 2014) as a baseline model. Bushra Jawaid et al. (2014) crawled a huge amount of Urdu text from cyberspace and performed automatic POS tagging using SVM. On test data, the authors proposed standalone POS tagger achieved an accuracy of 88.74%.

After testing, the proposed CRF model could correctly predict and classify all the POS tags with little variation in results for each tag. In this study, we also investigated the impact of features on the performance of CRF. We compared the results generated by our feature set with the results generated by the SVM model mentioned in the baseline. Following a 10-fold cross-validation test, our CRF based approach was found to supersede the baseline approach. Table 3-10 shows the average Precision, Recall and F-measure of proposed and baseline approach. Similarly, Table 3-11 shows individual POS tag results on CLE dataset while Table 3-12 shows BJ dataset individual POS tag results.

Table 3-10: Average Precision, Recall and F_Measure of Proposed and baseline approach

Dataset	Model	Precision	Recall	F_Measure
CLE Dataset	Baseline (SVM)	82.99	78.12	78.44
	Proposed (CRF)	90.72	85.42	86.99
BJ Dataset	Baseline (SVM)	88.67	83.75	85.22
	Proposed (CRF)	95.13	92.73	93.56

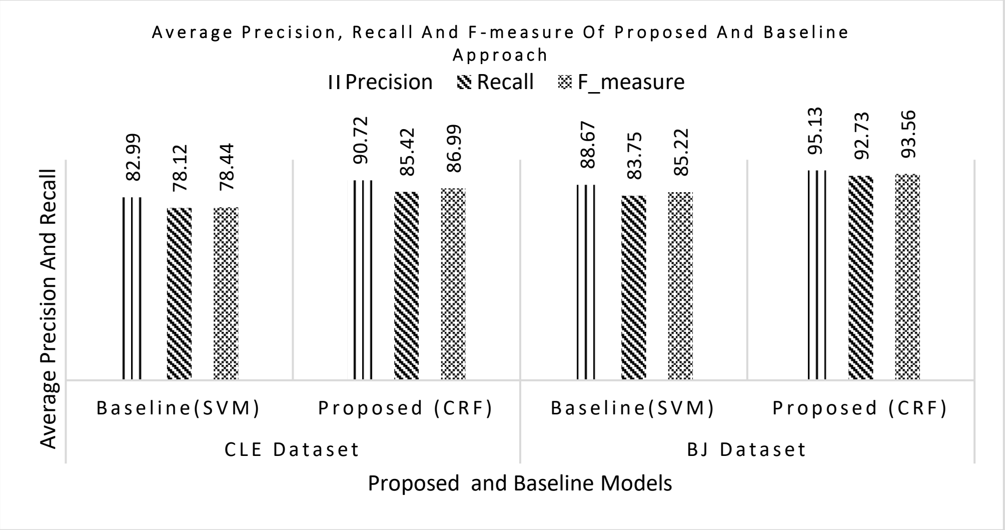


Figure 3-5: Graphical Representation of Average Precision, Recall and F_Measure

On CLE dataset CRF recorded 86.99 f_measure value while SVM recorded 78.44 f_measure value. On CLE dataset the maximum margin value by which CRF supersede baseline approach is 8.55.

On BJ dataset CRF recorded 93.56 f_measure value while SVM recorded 85.22 f_measure value and the maximum margin value by which CRF supersede baseline approach is 8.34.

Table 3-11: Average precision, recall & f-measure of individual POS tags of CLE Dataset

S. No	**POS tag**	**Proposed (Linear Chain CRF)**			**Baseline (SVM)**		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
1	NN	94.2	98.78	96.43	83.6	98.22	90.23
2	NNP	89.77	80.17	84.62	93.65	53.79	67.21
3	PSP	99.23	98.45	98.84	96.31	99.76	98
4	VBI	98.06	88.79	93.16	96.44	88.77	92.41
5	VBF	97.24	93.45	95.3	84.43	62.39	71.69
6	CC	96.66	99.48	98.05	94.9	99.17	96.98
7	JJ	92.52	91.51	92	95.96	81.34	87.98
8	PRP	93.15	95.37	94.23	79.6	81.14	80.26
9	AUXA	92.31	97.78	94.95	80.39	79.89	80.01
10	PU	99.57	99.85	99.7	99.94	93.93	96.79
11	SC	95.45	98.16	96.78	83.43	96.47	89.34
12	RB	93.4	86.16	89.6	93.73	74.56	82.73
13	PDM	93.76	96.42	95.06	71.91	58.53	64.44
14	CD	98.43	92.66	95.45	98.3	89.29	93.51
15	PRT	98.38	99.31	98.83	98.54	87.56	92.69
16	APNA	99.32	99.86	99.59	99.33	99.85	99.59
17	PRR	97.46	98.1	97.76	83.37	99.29	90.44
18	AUXT	95	98.97	96.93	61.98	93.56	74.47
19	PRD	92.38	88.38	90.16	0	0	0
20	Q	99.07	97.95	98.5	97.47	98.49	97.97
21	AUXM	97.82	96.05	96.89	93.13	90.99	91.92
22	SYM	98.57	92.94	95.37	93.97	95.78	93.84
23	NEG	99.76	99.7	99.73	100	100	100
24	AUXP	98.13	98.39	98.24	79.9	93.15	85.93
25	SCK	95.06	96.72	95.81	43.5	80.21	56.25
26	OD	97.02	87.65	91.94	98.11	91.14	94.44
27	FF	77.7	73.84	75.34	65	15.47	19.63
28	PRS	97.7	98.07	97.84	99.55	99.37	99.46
29	PRF	97.5	90	92.88	98	90.54	93.71
30	VALA	99.71	100	99.85	99.46	100	99.73
31	SCP	87.63	28.76	42.97	53.69	14.93	23.01
32	FR	90	66.37	74.54	100	94.5	97.02
33	PRE	30	20	23.33	70	56.67	61.33
34	INJ	73.21	33.13	42.31	83.67	43.89	50.41
35	QM	20	8.33	11.67	33.33	31.67	32

Table 3-12: Average precision, recall & f-measure of individual POS tags of BJ Dataset

S. No	**POS tag**	**Proposed (Linear Chain CRF)**			**Baseline (SVM)**		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
1	A	99.66	98.06	98.81	100	95.28	97.56
2	AA	93.67	93.49	93.57	77.49	76.51	76.97
3	AD	100	100	100	100	100	100
4	ADJ	91.64	88.2	89.88	89.2	74.18	80.99
5	ADV	96.05	91.8	93.87	91.69	88.93	90.28
6	AKP	100	98.14	99.04	99.65	99.85	99.75
7	AP	100	99.89	99.95	100	100	100
8	CA	97.01	89.44	93.05	97.74	83.67	90.14
9	CC	99.82	99.92	99.87	99.71	100	99.86
10	EXP	69.37	84.2	75.76	45.08	52.28	48.21
11	FR	60	40.83	45.67	100	94.17	96.57
12	G	98.64	99.41	99.01	98.42	99.32	98.86
13	GR	99.27	100	99.63	100	100	100
14	I	98.29	97.56	97.92	99.45	73.52	84.52
15	INT	100	86.63	92.06	92.68	87.66	89.72
16	KD	98.98	98.38	98.66	95.34	86.86	90.22
17	KER	93.03	93.1	92.95	0	0	0
18	KP	92.57	80.28	85.62	76.16	55.64	60.69
19	NEG	99.76	99.65	99.7	99.61	99.42	99.51
20	NN	92.66	97.53	95.03	79.68	93.93	86.21
21	OR	97.92	86.5	91.45	99.71	90.55	94.78
22	P	99.52	99.96	99.73	96.87	99.98	98.4
23	PD	94.06	97.4	95.67	57.31	62.08	59.55
24	PM	97.86	97.93	97.89	99.18	70.78	82.54
25	PN	87.15	81.55	84.24	70.3	57.04	62.96
26	PP	98.76	97.07	97.9	85.64	79.14	82.25
27	Q	98.16	95.99	97.04	96.6	93.42	94.98
28	QW	87.83	95.02	91.12	100	34.03	50.67
29	REP	99.46	100	99.73	100	100	100
30	RP	100	97.47	98.63	100	96.94	98.4
31	SC	98.43	97.59	98.01	85.51	97.55	91.13
32	SE	99.95	100	99.97	99.34	100	99.67
33	SM	99.83	99.98	99.9	100	100	100
34	TA	95.84	94.94	95.38	64.34	97.04	77.37
35	U	90	59.92	71.26	100	92.36	95.85
36	VB	95.82	93.17	94.47	84.12	66.85	74.5
37	WALA	98.83	100	99.4	100	100	100

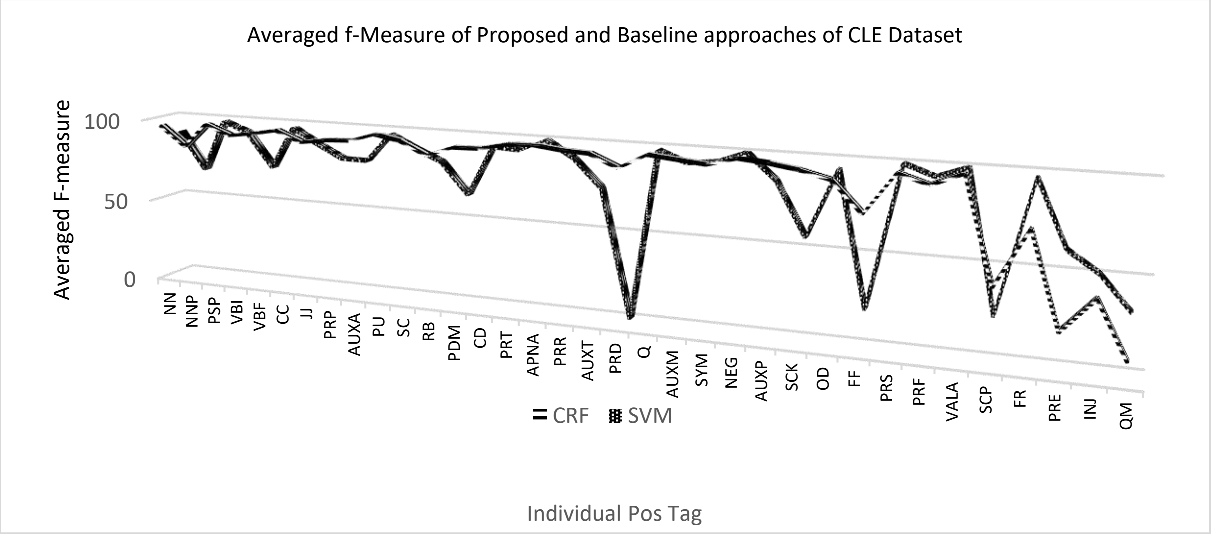


Figure 3-6: Graphical Representation of Average f-Measure of individual POS tags of CRF and SVM Model on CLE Dataset

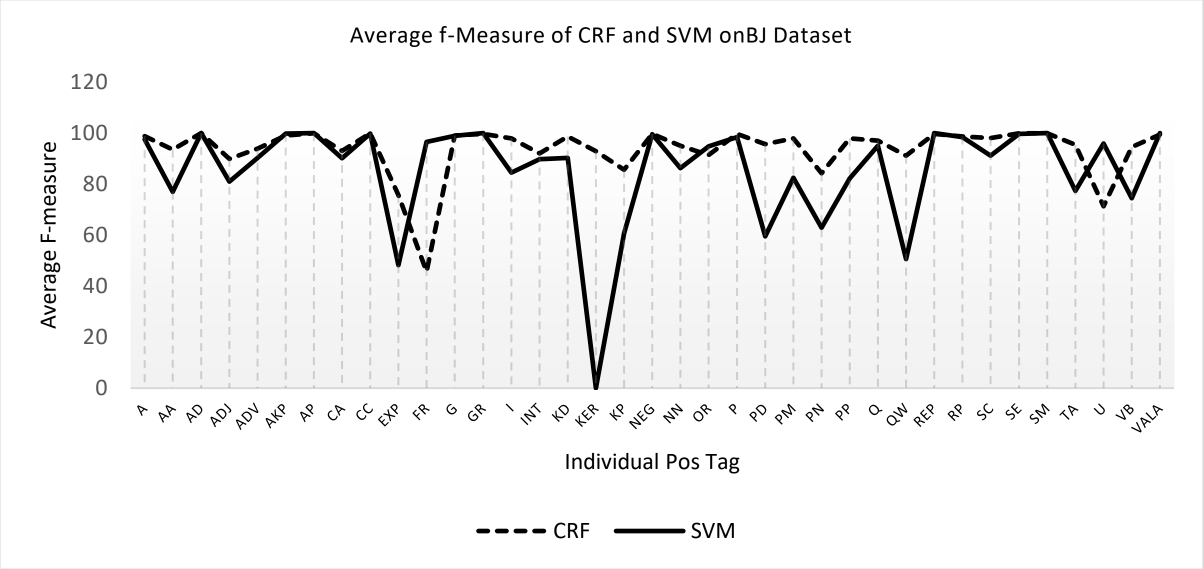


Figure 3-7:Graphical Representation of Average f-Measure of individual POS tags of CRF and SVM Model on BJ Dataset

Like Arabic, most Urdu words consist of two or more morphemes e.g the Urdu word خوش اخلاق (Khush Akhlaq, Good Character) is a combination of two morphemes خوش (Khush, Happy) and اخلاق (Akhlaq, Character). Similarly, it is common to find that the majority of Urdu words acquiring a different meaning with different context words. For example, the word جان(Jan) can be used as a common noun (NN), as a proper noun (PN) and as an adjective (Adj). These polymorphemic and dual behavior of most Urdu words makes Urdu POS task very challenging.

The SVM model in its training phase benefits from the maximum margin technique and have the capability to handle all observations one by one. This model also requires encoding last word and the next word as features of any particular observation (Hoefel & Elkan, 2008). SVMs classify any data into classes with the usage of various kernel functions. They carry out this through mapping out the data onto a best linear classifying hyperplane. Lastly, SVMs do not perform advantageously for many feature values. For multitude feature values, it would be harder to discover discriminatory lines to categorize the labels. During automatic learning of SVM, most of the errors are due to the confusion of PN tag with NN and VB and EXP tags, and NN tags with PNN, Adj, and VB tags. Details of most confused tags using SVM on BJ dataset is given in Table 6-1 (See Appendix). The majority of these confused tag errors are due to lack of feature utilization in the learning approach of the SVM model. Table 3-13 and Table 3-14 represent a portion of the detailed confusion matrixes of the proposed and baseline approaches given in the appendix. From the confusion matrixes, it is clear that the proposed approach more correctly identified all POS tags than the baseline approach. For example, the number of times the proposed approach correctly identified PN tag is 7802 times while the baseline approach identified it 7089 times. Similarly, the proposed approach confused 1784 times PN tag with NN tag while the bassline approach confused it 4617 times with NN tag. The proposed approach confused it 95 times with EXP tag while the baseline approach confused it 132 times with EXP tag. With ADV tag it confused 20 by the proposed approach and 26 times by the baseline approach on BJ dataset.

Table 3-13: Portion of the baseline approach confusion matrix on BJ dataset

	PN	NN	ADJ	VB	EXP	AA	ADV	CA	PM	TA	SC	I	Q	G	INT
PN	7089	4617	262	174	132	52	26	25	24	9	7	5	4	2	1

Table 3-14: Portion of proposed approach confusion matrix on BJ dataset

	PN	NN	SM	ADJ	VB	EXP	CA	PM	TA	SC	ADV	RP	I	G	U
PN	7802	1784	404	294	192	95	104	35	24	21	20	15	13	11	16

We moved towards enhancing the efficiency of existing Urdu POS tagging model by presenting the CRF model as an alternative. CRFs bring together the best feature of both generative and classification models. Similar to classification models, they adopt a lot of statistically related features from input data to build training model and like generative models, they deal away decisions at divergent sequence points(Raymond & Riccardi, 2007). CRF take advantage of its ability of utilization of correlation between two tags. CRFs can also, by nature, deals with the state-to-state dependencies and feature-to-state dependencies. On the test data, the CRF model confused mostly PN tag with NN tag and ADJ tag with NN tag. Details of most confused tags using CRF on BJ dataset is given in Table 6-2 (See Appendix)

In Urdu **(اور, Aur**) and (**لیکن, Lekin)** belongs to the conjunction word category, and in the text, it acquires Coordinating Conjunction(CC) tag, a subcategory of Urdu conjunctions (Tafseer et al., 2015).
Below example explains the acquiring of CC tag of **(اور, Aur**) and (**لیکن, Lekin).**

جب/NN سرسیدؒ/NNP نے/PSP انگریزی/NNP پڑھنے/VBI پر/PSP زور/NN دیا/VBF
اور/CC عام/JJ مسلمانوں/NN نے/PSP ان/PRP کی/PSP صدائیں/NN ان‌سنی/JJ کر/VBF
دیں/AUXA ،/PU تو/SC اس/PRP کے/PSP نتیجے/NN میں/PSP وہ/PRP مادی/JJ
ترقی/NN کے/PSP لحاظ/NN سے/PSP پیچھے/NN ضرور/RB رہ/VBF گئے/AUXA
لیکن/CC یہ/PDM ایک/CD طرح/NN سے/PSP اچھا/NN ہی/PRT ہوا/VBF -/PU
بڑے/JJ بڑے/JJ رجواڑوں/NN ،/PU نوابوں/NN ،/PU جاگیرداروں/NN اور/CC
سرمایہ‌داروں/NN نے/PSP اپنے/APNA بچے/NN انگریزی/JJ اسکولوں/NN **اور/CC**
کالجوں/NN میں/PSP بٹھا/VBF دیے/AUXA **اور/CC** غریب/JJ مسلمانوں/NN کے/PSP
لیے/PSP صرف/JJ یہی/PDM راہ/NN باقی/JJ رہ/VBF گئی/AUXA کہ/SC وہ/PRP
اپنے/APNA بچے/NN دینی/JJ مدرسوں/NN میں/PSP داخل/NN کرا/VBF دیں/AUXA
اور/CC انہوں/PRP نے/PSP داخل/NN کرا/VBF دیے/AUXA -/PU اس/PDM
طرح/NN قال‌اللہ/NNP اور/CC قال‌الرسولؐ/NNP کی/PSP مقدس/JJ گونج/NN میں/PSP
ان/PRP کی/PSP متاع/NN ایمان/NN محفوظ/JJ ہو/VBF گئی/AUXA -/PU

The example given in Table 3-15 describes that how our proposed CRF model confused CC tag with SCP tags. In test data, the total occurrences of the word (aur) with CC tag are 1143, after evaluation, it is 1 time confused with NN, 1 time with NNP, eighteen times with JJ while 55 times with SCP tag respectively. Details of the most confused tags using CRF model on CLE POS tagged corpus can be viewed in Table 3-17.

Table 3-15:Concatenated output of CRF on CLE dataset for CC Tag

Before CRF Evaluation (Original Data)		***After CRF Evaluation (after testing)***	
Word	**Tag**	**Word**	**Tag**
کچھ	Q	کچھ	Q
کاٹھیاں	NN	کاٹھیاں	NN
مرمتطلب	JJ	مرمتطلب	VBF
ہیں	VBF	ہیں	AUXT
اور	**CC**	**اور**	**SCP**
دو	CD	دو	CD
تین	CD	تین	CD
گدھوں	NN	گدھوں	NN
کے	PSP	کے	PSP
نعل	NN	نعل	NN
غائب	JJ	غائب	JJ
ہیں	VBF	ہیں	AUXT

Similarly, in test data, the words (آکسفورڈ, Oxford) and (جنت, Jannat) (Heaven) qualifies NNP tag, but after evaluation it with CRF model these words are concatenated with NN tag. An original tag for the word (اسٹریٹ, street) is NN while after testing, it is concatenated with VBF tag. Table 3-16 depicts the confused behavior of the CRF model for NNP and NN tags.

Table 3-16: Concatenated output of the CRF model on CLE Dataset for NNP and NN tags

Before CRF Evaluation (Original Data)		***After CRF Evaluation (after testing)***	
Word	**Tag**	**Word**	**Tag**
ہم	PRP	ہم	PRP
'	PU	'	PU
ٹاورآفلندن	NNP	ٹاورآفلندن	NNP
'	PU	'	PU
کی	PSP	کی	PSP
سیر	NN	سیر	NN
کر	VBF	کر	VBF
کے	SCK	کے	SCK
باہر	NN	باہر	NN
نکلے	VBF	نکلے	VBF
اور	CC	اور	CC
بس	NN	بس	NN
میں	PSP	میں	PSP
بیٹھ	VBF	بیٹھ	VBF
کر	SCK	کر	SCK
پھر	RB	پھر	RB
آکسفورڈ	**NNP**	**آکسفورڈ**	**NN**
اسٹریٹ	**NN**	**اسٹریٹ**	**VBF**
جا	VBF	جا	VBF
پہنچے	AUXA	پہنچے	AUXA
جو	PRR	جو	PRR
خریداروں	NN	خریداروں	NN
کی	PSP	کی	PSP
جنت	**NNP**	**جنت**	**NN**
ہے	VBF	ہے	VBF
-	PU	-	PU

Table 3-17: Summary of Most confused tags along with its corresponding words

Word	Tag	Frequency
اور	NN	1
اور	NNP	1
اور	CC	1068
اور	JJ	18
اور	SCP	55
اس	PRP	469
اس	PDM	318
دیے	NN	2
دیے	VBF	8
دیے	AUXA	9
ہو	NN	1
ہو	VBF	406
ہو	AUXA	1
ہو	AUXT	20
ہو	AUXM	6
لیے	PSP	251
لیے	VBF	2
لیے	AUXA	2
یعنی	NN	1
یعنی	NNP	1
یعنی	CC	11
یعنی	JJ	2

3.4 Chapter Summary

The contemporary period has witnessed the intense development of machine learning techniques, often as state-of-the-art approaches to solving problems such as POS tagging. The main reason for wide usage is based on four features: a) the capability of automatic learning b) the degree of accuracy c) the speed of processing and d) generic nature. For training and testing, such ML approaches only require a pre-POS tagged dataset. In machine learning oriented approaches, the highest degree of intelligence exists in a good feature set construction(Daud et al., 2016; Khan et al., 2016). A good set of features is more important than the learning model. Hence, the construction of a good feature set is highly required. Our experiments show our proposed CRF based model to outperform the baseline approach. The benefit of our work is that researchers from the ULP research domain can find both Urdu POS challenges and its corresponding solution in single research space.

About the Chapter

This chapter explores the second contribution namely the Urdu NER module of our proposed framework. The main content covered are: (A) NER in context of DRNN is discussed (B) Challenges are explored in sufficient detail (C) feature set is presented (D) Proposed model is discussed (E) and finally results are discussed.

Named Entity Recognition using DRNN

Named Entity Recognition is amongst the most basic of NLP tasks and is referred to in the literature by various names, e.g as “entity identification”, “entity chunking” or “entity extraction”. It corresponds to the identification and classification of all proper nouns in texts, into predefined categories, such as persons, locations, organizations, expressions of times, quantities, monetary values, etc. (Sundheim, 1996). NER is generally considered a sequential labeling task and plays a vital role in the management and extraction of intelligent information from the text, which may be helpful in building a relatively simple system (Seok et al., 2016). Generally, NER is considered as an essential preliminary to the higher steps in complex NLP tasks, such as question answering, conversation, voice search etc (Lu et al., 2015).

The general approach of NER tasks is to generate and assign one class label to each part of a sentence. This task is particularly challenging for languages such as Urdu, suffering from both lack resources as well as lack capitalization feature for computational linguistic analysis. The morphological richness characteristic of Urdu also makes the NER task a challenging one. Since in morphologically rich languages, a single word can admit multiple forms, and can thereby acquire multiple NE tags in different contexts

In Urdu, it is common to find that the majority of Urdu words acquiring a different meaning with different context words. For example, the word جان(Jan) can be used as a common noun, as a proper noun, and as an adjective. These polymorphemic and dual behavior of most Urdu words makes Urdu NER task very challenging.

From the literature review, it came in our knowledge that up to now, there has no research work reported from the ULP research community in which the researchers have examined the effect of deep learning models and word embeddings of training data for Urdu NER. Therefore, this gap, the limitation of rule-based approaches, advantages of deep learning architecture in diverse areas, discussed above, has motivated us to undertake DRNN model along with the word embedding as a feature for Urdu NER in this study.

Table 4-1: Various named entities and its description

Type	Tag	Sample Category
Person	<PERSON>	Individuals, small groups
Location	<LOCATION>	Territory, land, kingdom, mountains, site, locality etc
Organization	<ORGANIZATION>	firms, a group of players, Political parties, bureau etc
Designation	<DESIGNATION>	Various designations e.g. Professor, Dean, Mufti, Captain etc.
Number	<NUMBER>	Counts e.g. Hundred, Ten Thousand One, 10 million etc.
Date	<DATE>	Date stamps
Time	<TIME>	Clock time stamps

Generally, most previous NER research in Urdu is limited to only three entity classes namely "PERSON", LOCATION" and "ORGANIZATION" here in this study we extended it to seven entity classes e.g. "PERSON", LOCATION", "ORGANIZATION", "DESIGNATION", "NUMBER", "DATE", "TIME". The seven entity classes along with its description can be found in Table 4-1.

5.1 Challenges to Urdu NER

Morphologically-rich languages such as Urdu and Arabic pose a challenge to NER. This is due to them possessing short vowels and complex orthography, and since they lack capitalization and sufficient machine-readable linguistic resources. This section briefly discusses each such challenge.

5.1.1 Lack of Capitalization

Most of the proper nouns in English and other Latin script-based languages such as German, French, Irish and Italian etc begin with capital letters. The presence of capital letters allow proper names to be easily recognized(Mukund et al., 2010; S. Naz et al., 2014; Riaz, 2010). A wide range of NLP tasks depends on capitalization. A key such task, which arises a subtask for other NLP processes, is sentence boundary detection (SBD). This is an exploratory method for setting up text files for machine-driven NLP tasks, for instance, named entity recognition, machine translation, POS labeling, text summarization, information retrieval, parsing, chunking and many more (Daud et al., 2016; Wong, Chao, & Zeng, 2014). A requirement of most readily available ML algorithms and tools, for example, CRFSharp,[6] RNNSharp[7] , and

[6] https://github.com/zhongkaifu/CRFSharp

[7] https://github.com/zhongkaifu/RNNSharp/tree/master/RNNSharp

CRF++, for tasks such as NER, POS and so on, is that the training data is organized sentence-wise.

Unfortunately, as the orthography of Arabic script-based languages does not support it, in common with other South Asian languages, Urdu lacks capitalization(Daud et al., 2016; Riaz, 2010). As such, SBD and tasks that depend upon it(Farghaly & Shaalan, 2009), such as NER, are far more challenging in Urdu than in Western languages.

4.1.3 Lack of Supporting Resources

Urdu is disadvantaged by a lack of NER supporting resources (e.g grammatically annotated corpora, specialized dictionaries, gazetteers) and enabling technologies(Riaz, 2010), both of which are useful for doing machine learning based computational linguistic analysis. There are numerous linguistic resources of Urdu that linguistics exercises every day(Daud et al., 2016). In general, these resources are used by rule-based systems.

Research on machine learning based NER for Urdu is still in the initial stages, mainly due to the paucity of standard NER supporting resources(Mukund et al., 2010). Any supervised ML approach for NER requires a large, pre-labeled NER dataset (Khan et al., 2016).

4.1.4 Optional Short Vowels

As are Arabic and its derivations, also in Urdu each letter is associated with a single sound (Daud et al., 2016; Farghaly & Shaalan, 2009; Hussain, 2004). Urdu does not maintain separate characters for short vowels (Zer 'i' "◌ِ", Zabar 'a' "◌َ", and Pesh 'u' " ُ"), instead of representing them by adding diacritics, although doing so is neither a mandatory nor a common aspect of writing in Urdu. Thus, as diacritics/*Aerab* are not systematically checked – their use being left to the author's discretion(Raza & Hussain, 2010) – each text that is represented in Unicode commonly requires interpretation to remove semantic ambiguity.

For example the word سرور can be interpreted as person NE when it is diacritized as سَرور(Sarwar) and as a normal entity (NOR) when diacritized as سَرُور(Saroor). Thus, the practice of diacritics/Aerab commute the phonetic representation and generate dissimilar meaning to the same lexical form.

4.1.5 Agglutinative nature

Typically, an agglutinate language starts out with a stem and produces new words by concatenating small, significant supplementary parts - called affixes(Jabbar, Iqbal, & Khan, 2016). The affixes are morpheme or set of morphemes that are oftentimes connected at the fore as well as at the back of the word to produce a new word. When the affixes are glued to the front it is termed as a prefix, while suffix when glued at the back of the word. The Urdu holds agglutinative nature, to which few supplementary features can be tallied to the word to produce harder meaning(Riaz, 2010). E.g from the word گجرات (Gujrat) → گجراتی(Gujrati). گجرات (Gujrat) points location when aggregated with a case marker like (I) becomes گجراتی(Gujrati) which means a people residing in گجرات(Gujrat) which is not a named entity. It is hard for the NER system to discover it as a NE and assign a location class.

4.1.6 Ambiguity in Named Entities

Urdu and other South Asian languages confront the problem of ambiguity as in English(Riaz, 2010). The more common situation is the common Noun vs proper noun. Common noun occasionally comes as proper names e.g as a personal name. For example, look at the given example in Urdu such as "وکیل" (wakeel) which entails an advocate can also be used as person name, thereby producing to a conflict situation, thus creating ambiguities between a common noun and proper noun(S. Naz et al., 2014; Riaz, 2010). Similarly, the term انضمام (Inzimam) which means merging, denoting a common noun, is oftentimes used as person name in Pakistan as well as in the Pakistani communities around the globe.

4.1.7 Lack of Uniformity in Spellings

Spelling variations are oftentimes caused by Urdu writers and reports with relation to certain proper names(Riaz, 2010). The root cause for these variations in Urdu is the phenomenon that different alphabetic character comprising the same phoneme(Ijaz & Hussain, 2007). Such characters are also called homophone characters(Ijaz & Hussain, 2007). Besides, the majority of people learn to mix up several homophones for one another. Therefore, as a result, Inappropriate spelling of words bearing homophones gets quite common. In Table 4-2 we provided common examples of spelling variation for the same person names in which the spelling variation for the personal names "Masood" and "Noman" already pointed out in the research work of (S. Naz et al., 2014; Riaz, 2010).

Table 4-2: Spelling Variation of Urdu person names

Person names	Spelling-I	Spelling-II
Anzah	عنزہ	آنزہ
Azkah	ازکیٰ	ازکہ
Masood	مسعود	مسود
Noman	نعمان	نومان
Amber	عنبر	امبر
Nazrana	نظرانہ	نذرانہ
Toorpakai	طورپکئی	تورپکیئ
Mashal Khan	مشال خان	مشعل خان
Samreen	ثمرین	سمرین
Ismael	اسما عیل	اسمعیل

4.1.8 Nested Entities

Nested entities also produce ambiguity because they hold in two or more proper names(S. Naz et al., 2014; Riaz, 2010). For example قائداعظم یونیورسٹی (Quaid-e-Azam University) represents a nested entity. It produces a problem for NER system as the term قائداعظم (Quaid-e-Azam) relates to a person, whereas the term یونیورسٹی (University) falls into the organization class which produces a problem to distinguish the appropriate label.

4.1.9 Conjunction Ambiguity

The conjunction ambiguity was initially pointed out in IJCNLP-2008 NER workshop (Riaz, 2010). Conjunctions are words which are used to concatenate two are more words or phrases and even sentences. Since these days a lot of English words are commonly used in spoken and written Urdu among which one word is اینڈ[and] equivalent to the Urdu word اور [aur, and], is the most commonly practiced conjunction word in Urdu. It is usually used to join two separate entities to make up a single entity. Common examples of conjunction ambiguity are شاہد اینڈ عثمان انٹرپرئیزز(Shahid and Usman Enterprises), انگلینڈ اینڈ ویلز(England and Wales), انگیلنڈ اینڈ ویلز کر کٹ بورڈ(England and Wales Cricket Board). It is hard for the NER system to discover the conjunct entities as a single NE.

4.2 Proposed Method

Our proposed deep learning based Urdu NER system consists of two stages. The first stage makes use of a restricted amount of labeled data, whereas the second stage makes use of a huge amount of unlabeled data. This kind of learning is called semi-supervised learning due to the fact that it makes use of both labeled and unlabeled data (see Figure 4-1).

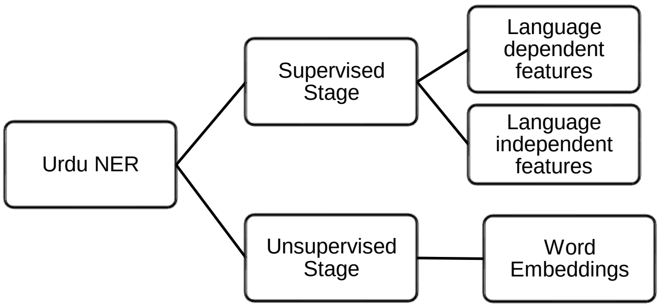


Figure 4-1: Schematic Representation of the Proposed Deep learning based Urdu NER system

4.2.1 Features

In almost all approaches, most high-pitched performance devolves on good feature set construction. In fact, a good set of features positively affects performance more than the choice of a learning model could. Many progressive NER systems use various linguistic features ranging from semantic information about words to information about the morphological and syntactic structure of words(Seok et al., 2016). In most contemporary approaches to NLP tasks, it is typical to first learn word representation from labeled or unlabeled data and then to use these representations as features in supervised algorithms. From literature review, we observed that the joint use of word embedding along with other feature has shown record-setting performance for NER in low resources languages of the globe. Das et al. (2017) proposed an NER system with word embeddings for Bengali a low resource language. The authors evaluated the performance of their proposed system on INCNLP-2008 dataset of Bengali by comparing its performance with CRF. The proposed system achieved an overall 11.2% improvement in F-score over the baseline approach. Similarly, Demir and Ozgur (2014) also showed the significance of word embedding as a feature for NER in the Turkish language, also belongs to a low resource language. They evaluated the proposed system for Turkish and Czech languages and showed improvement over the state-of-the-art F-score obtained for Turkish by 2.26% and for Czech by 1.53%. Therefore, the significance of word embedding for NER in low resources language motivated us to propose a new feature set for Urdu which is also a low resource language. Our proposed new feature set is consisting of word embedding along with language dependent and language independent features.

To the best of our knowledge, DRNN along with word embedding, language dependent, and language independent features have not been explored previously for Urdu NER task.

The following details the features we used for training.

4.2.1.1 Template Feature

Template features, also known as sparse features. In this type of features, if current token hits some features, the value of only these features become non-zero, however, other features values are zero. Feature templates are used to produce feature sets from training and test files where feature template used in our experiments consists of nine unigram language independent features as well as language-dependent features. Our template files are organized row-wise, such that each row represents a single template, and stored in a separate text file. All templates have three components, prefix, id and rule-string, where the prefix relates to template typecast. The two supported prefix types are “U” for unigram and “B” for bi-gram. The Id is used to discriminate templates from each other while the rule-string portion of the template is accustomed to guiding DRNN to produce the feature. DRNN stores feature in a separate model file generated from training records according to given templates. Details of features template file used during training and testing phase are given in Table 4-3.

Table 4-3: Template Feature Set

Template	Description
U01: %x [-1,0]	Previous lexical word
U02: %x [0,0]	Current lexical word
U03: %x [1,0]	Next lexical word
U04: %x [0,1]	POS tag of current lexical word
U05: %x [0,0] %x [0,1]	Bi-gram of current word + current POS tag
U06: %x [1,0] %x [1,1]	Bi-gram of N+1word + POS tag of N+1 word
U07: %x [-1,0] %x [-1,1]	Bi-gram of N-1 word+ POS tag of N-1 word
U08: %x [-2,0] %x [-2,1]	Bi-gram of N-2Previous word + POS tag of N-2 previous word
U09: %x [2,0] %x [2,1]	Bi-gram of N+2 next word + POS tag of N+2 next word

4.2.1.2 Context Template Feature

Context template features are based on features generated by templates for a particular token, and is represented by mentioning the context set up, in our case the context setup for this feature is: “-2, -1, 0, 1”. In this context set up “0” refer to the current token, “1” refer to N+1 token while “-1” and “-2” refers to N-1 and N-2 previous tokens. This context setup will combine the features of the current token with its N+1 token along with its N-1 and N-2 tokens. Consider the following Urdu sentence.

صفائی نصف ایمان ہے۔

Suppose our current token is ایمان (Eman) of the given sentence then generated feature set will be:

Template feature of (“نصف, Nisaf”), Template feature of (“صفائی, Safai”), Template features of (“ایمان, Eman”), Template features of (“ہے, hey”).

4.2.1.3 Word Embedding Feature

Recently the use of word embedding as a feature for NER task in the context of low resources languages arousing intense research interest in NLP research community (Das et al., 2017). The motivation behind the use of word vector embedding’s is that words acquiring congruent named entity label should occur close to each other.

Word embedding features, known as dense features, play a vital role in cases where the unlabeled corpus is much larger than the labeled corpus. These features are used to describe the features of the given token. In word embedding feature each token can point to a vector and the value of this vector will be used as embedding feature (Seok et al., 2016). In word embedding vectors of corresponding words are produced in an unsupervised manner. Word embedding corresponds to the grouping together of similar words, to capture the diverse aspects of meaning, and in addition, the phrase based information within the vector representing features. This improves NLP task performance as it resolves the problem of data scarcity of low resources languages (Seok et al., 2016). Each word is represented in ‘n’ dimensional vector. Consider the following example with n=14.

اگر ہم اچھے مسلمان نہ صحیح اچھے اِنسان ہی بن جاتے تو ایسا کبھی نہ ہوتا۔

For the above-given sentence, we would like to produce a word vector of dimension n equal to fourteen for each unique word. The vocabulary of the above sentence is given below.

اگر,ہم, اچھے, مسلمان, نہ, صحیح, اِنسان, ہی, بن, جاتے, تو, ایسا, کبھی, ہوتا, ,۔

For each word of the vocabulary, we are required to create a word vector. The following cooccurrence matrix depicts the occurrence of a neighbor’s word.

Table 4-4: Cooccurrence Matrix

	اگر	ہم	اچھے	مسلمان	نہ	صحیح	اِنسان	ہی	بن	جاتے	تو	ایسا	کبھی	ہوتا
اگر	0	1	0	0	0	0	0	0	0	0	0	0	0	0
ہم	1	0	1	0	0	0	0	0	0	0	0	0	0	0
اچھے	0	1	0	1	0	0	1	0	0	0	0	0	0	0
مسلمان	0	0	1	0	1	0	0	0	0	0	0	0	0	0
نہ	0	0	0	1	0	1	0	0	0	0	0	0	1	1
صحیح	0	0	1	0	1	0	0	0	0	0	0	0	0	0
اِنسان	0	0	1	0	0	0	0	1	0	0	0	0	0	0
ہی	0	0	0	0	0	0	1	0	1	0	0	0	0	0
بن	0	0	0	0	0	0	0	0	1	1	0	0	0	0
جاتے	0	0	0	0	0	0	0	0	1	0	1	0	0	0
تو	0	0	0	0	0	0	0	0	0	1	0	1	0	0
ایسا	0	0	0	0	0	0	0	0	0	0	1	0	1	0
کبھی	0	0	0	0	1	0	0	0	0	0	0	1	0	0
ہوتا	0	0	0	0	1	0	0	0	0	0	0	0	0	0

In the above matrix, each row corresponds to vector for each word, after extracting rows from the above matrix indicates simple initialization of word vectors.

We used Txt2Vec[8]project - a C# implementation – to generate our word embedding features. Txt2Vec project interpret specified words, phrases into its corresponding vectors such that each dimension of the vector corresponds to a feature. Characteristics such as incremental training capability and model vector quantization are its additional features. The corresponding vector generation is based on the semantics of words, and for this purpose, the cosine similarity measure is used.

Table 4-5:Word Vector

اگر	= [01000000000000]
ہم	=[10100000000000]
اچھے	=[01010010000000]
مسلمان	=[00101000000000]
نہ	=[00010100000011]
صحیح	=[00101000000000]
اِنسان	=[00100001000000]
ہی	=[00000010100000]
بن	=[00000000110000]
جاتے	=[00000000101000]
تو	=[00000000010100]
ایسا	=[00000000001010]
کبھی	=[00001000000100]
ہوتا	=[00001000000000]

[8] https://github.com/zhongkaifu/Txt2Vec

4.2.1.4 Run Time Feature

Run-time features extend the feature pool by employing previous output tokens as features of the current token. This feature is only available for forward-RNN, bi-directional RNN does not support it.

4.2.2 Recurrent Neural Network

Recurrent neural networks (RNNs)(Rumelhart, Hinton, & Williams, 1985) belong to the artificial neural networks family (Deng, 2014). RNN, a type of forward neural network, but then, can manage input/output of arbitrary lengths. The internal structure (hidden layers) of an RNN contains recursive edges (feed-back connections) between units as compared to its counterparts, the traditional neural nets. The hidden layers of RNNs which contain recursive edges are termed as memory units of the network as its calculation is based on the output of previously hidden state, along with the input of the current input state. The hidden layers of RNNs are the core element, which captures information about the input sequence. Therefore, with this information persistence capability of hidden layers neurons, as the time steps increase, the units get influenced by larger and larger neighborhood or in simple words the network acquires long-term dependencies, enabling RNNs to demonstrate propelling historical performance. Traditional neural networks and shallow structure classifiers such as CRFs, hidden Markov models (HMMs), maximum entropy (MaxEnt) and support vector machines (SVMs) do not support long-term dependencies, preventing them from developing information persistence (Deng, 2014; Schmidhuber, 2015; Zhang et al., 2016)

The equations that regulate the calculation taking place in an RNN are given below:

$$h_t = \sigma(U.x(t) + W.h_{t-1}), \qquad t = 1 \dots .T \quad (4\text{-}1)$$

$$y(t) = g(Vh(t)) \qquad (4\text{-}2)$$

- ***x(t)*** represents any particular input at any time stamp (*t*) to the input layer and corresponds to one vector
- h(t) is considered the memory unit of the network and corresponds to the hidden layer at any time stamp (*t*). Its calculation is based on the output of the previous hidden state, along with the input of the current input state;
- g(Vh(t)) calculates y(t) by using the output of the hidden layers and weights between the hidden layer and output layer y. y(t) corresponds to output at time stamp (*t*);

- $\sigma(Tanh)$ represents a function, responsible for initializing the first hidden state.

4.2.3 Deep Recurrent Neural Network

The most naive, conceivable and widely used architecture of deep RNN is proposed in the works of (Schmidhuber, 1992). The authors proposed architecture is settled upon stacking of multiple recurrent layers to frame a deep RNN, with the objectives to resolve computational expensiveness problem of previous versions of neural networks. The layers stacking architecture possibly consider the hidden state at each level to operate at varied timescale(Pascanu, Gulcehre, Cho, & Bengio, 2013). This affords an enhanced learning capability.

- **RNNs Training**

The training of RNNs can be accomplished by either long short-term memory (Hochreiter & Schmidhuber, 1997) or back-propagation through time (BPTT) (Williams & Hinton, 1986) (Goller & Kuchler, 1996) algorithms.

The computational complexity of learning LSTM models per weight and time step with the stochastic gradient descent (SGD) optimization technique is O(1). Therefore, the learning computational complexity per time step is O(N) (Sak, Senior, & Beaufays, 2014). In case of d be the size of input sequence than in a single layer with input dimension n and output dimension m, the forward and reverse propagations will always be O(nmd) assuming a naive matrix product algorithm.

Pseudocode of the proposed DRNN based NER using RNNSharp libraries is given in Table 4-6.

Table 4-6: Pseudocode of the Proposed LSTM RNN model for NER task using RNNSharp libraries

Training Phase

```
RNNSharpInput_NER (string [mode, -trainfile -validfile -modelfile -tagfile -layersize 200,200 -modeltype   -ftrconfigfile  -alpha 0.1 -maxiter 20 -savestep 200K -dir 0 -dropout 0 -vq 0] Args)
for (a = 0; a < args.Length; a++)
Load arguments
End for
If mode is train
Load Train File
Initialize all given parameters
while (Train_file.End == false)
Read line of train file
for (b = 0; a < ctrconfigfile; b++)
Loading feature template from {0}
Extract features from it and convert it into sequence
Set label for the sequence
Add the sequence into data set
Show state at every 1000 record
Filter out features whose frequency is less than {0}
Save feature in model file
End for
End wile
```

Testing Phase

```
RNNSharpoutput_NER (string [-mode -testfile  -tagfile  -cfgfile -outfile ] Args)
for (a = 0; a < args.Length; a++)
Load arguments
End for
If mode is Test
Load model from file
while (test file.End == false)
Read line of test file
while (model file.End == false)
Create feature extractors and load word embedding data from file
Create decoder tagger instance
Initialize result
Append the decoded result into the end of feature set of each token
Save result in result file
End while
End while
```

4.3 Experiments

All 10-fold cross-validation experiments are conducted on 2.6 GHz Intel Core i7 PC with 12 GB of RAM and two hidden layers of RNN of layer size 200,200 is used.

4.3.1 Datasets

The effectiveness of the execution of an NLP task depend on the particular approach the corresponding algorithm takes – e.g. whether statistical or rule-based – and on the availability of gold standard data. Relative to southeastern languages, northwesterly languages benefit from an extensive body of research covering diverse NLP tasks, and as a result, benefit from plentiful resources for computational analysis such as gold standard data. Notable such resources include annotated NER datasets, WordNet, dictionaries, gazetteers, and related tools, most being easily accessible due to being open-source.

Most southeastern languages instead suffer from a scarcity of research and resources, Urdu - spoken in a vast area of the Indian sub-continent – is an example of a low-resource language that is yet to attract any significant body of work (Daud et al., 2016). Before the release of UNER dataset a ULP researcher, working on NER was limited to two datasets, the IJCNLP-2008 NE tagged dataset and Jahangir et al., dataset (Jahangir et al., 2012).

Our DRNN models were evaluated on these. In addition, we evaluated our DRNN models on UNER *news* dataset. UNER dataset contains text from three subdomains of the BBC Urdu website: sports, national news, and international news. This dataset is annotated manually by linguistic experts with by seven NE tags (PERSON, LOCATION, ORGANIZATION, DESIGNATION, NUMBER, DATE and TIME and is constituted by 48,673 tokens.

4.3.2 Training and Testing

Our work proposes a DRNN approach, testing it against CRF baselines using C# based open source libraries: the RNNSharp[9] and CRFSharp[10]. CRFSharp and RNNSharps are new CRF and DRNN packages and we believe that these are more flexible than the current state of the art. The main reason behind using CRFSharp is its model parameters encoding capability by L-BFGS (Zhu, Byrd, Lu, & Nocedal, 1997). Furthermore, it has a lot of respectable advances than CRF++[11], such that (a) complete parallel encoding (b) computer memory utilization in an optimized way and soon. During the training phase of large size training data with numerous tags, CRFSharp has the capability to manage efficient memory of multi-core CPUs with full usage compared to CRF++. Therefore, in the homophonic environment, CRFSharp is capable of encrypting harder models with lower cost than CRF++. Similarly, RNNsharp is an advanced implementation of the DRNN model and is widely adopted for various automatic linguistic tasks such as sequence labeling tasks including NER and much more.

The core feature of RNNSharp is its support of various RNN model types and structures. For example, RNNSharp supports BPTT (BackPropagation Through Time) (Goller & Kuchler, 1996) and LSTM (Long Short-Term Memory) (Hochreiter & Schmidhuber, 1997). Recurrent Neural structure in terms of memory and in terms of output layer it supports RNN-CRF and in

[9] https://github.com/zhongkaifu/RNNSharp

[10] https://github.com/zhongkaifu/CRFSharp

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

terms of direction, it supports forward and Bi-directional RNNs. For BPTT and LSTM, BPTT-RNN is usually called as simple RNN, since the structure of its hidden layer node is very simple. In all experiments for all model architecture, we keep the hidden layer size 200 and alpha value to 0.1.

We conducted our experiments on the IJCNLP-2008 dataset, Jehangir et al. dataset, and the UNER news dataset. As CRFsharp and RNNSharp require the input data in SBME[12] tagging format, we converted the data of the three datasets into an SBME format (structure is given inTable 4-7).

While S means that this entity is single, B, M, and E represent the beginning, middle and end portion of an entity, while NOR indicates that the token is a normal term and not an entity.

In the decoding phase, the encoded model file generated during the encoding phase is decoded using template features to predict named entity tags for each word of the testing file. Test files have a similar format as the training file. The only difference that distinguishes a test file from a training file is the exclusion of class label column from test file. Figure 4-2 shows the whole training testing process graphically, the steps are as follows:

1 Since the IJCNLP-2008 dataset contains extra information. e.g. a separate tag for start and end of a sentence, mixed sentences structure, nested entities, token number etc. so we performed preprocessing of IJCNLP dataset. In preprocessing we eliminated unnecessary data, organized data in sentences format, nested entities are converted to flat entities
2. The refined data is then tagged with Parts of speech tags (POS tags). For POS tagging we make legal use of CLE[13] POS tagged Urdu digest corpus. CLE dataset is a 100k manually POS tagged dataset.
3. For tagging, we incorporated the longest maximum matching technique (Sassano, 2014). For missing words, a default tag out of vocabulary (OOV) was assigned. Similarly, for all named entities, an NNP tag was assigned. After POS tagging the resultant data now contains both POS tags as well as NE tags.
4. After POS tagging the resultant data is now converted to SBME tagged format.
5. Divided the data into 70 percent train, 20 percent validation and 10 percent test data.

[12] https://github.com/zhongkaifu/CRFSharp

[13] http://www.cle.org.pk/clestore/index.htm

6. The template features are generated from 70 percent training data using the feature utility of RNNSharp.
7. Word Embedding features are generated from 70 percent training data using Txt2Vec[14] packages.
8. The configuration file is adjusted in which all features are mentioned and is supplied as a parameter to encoding module of DRNN.
9. DRNN algorithms (LSTM, BPTT) are used to build the training model using the 70 percent train and 20 percent validation data.
10. The 10 percent test data is tested on the training model
11. Evaluation using precision, recall, and f-Measure
12. Average results are produced

Table 4-7: Training File Format

Token	POS Tag	Class type
بریگیڈیئر	PNN	S_DESIGNATION
ایڈ	PNN	B_PERSON
بٹلر	PNN	E_PERSON
سنہ	PNN	B_DATE
دوہزارچھ	PNN	E_DATE
میں	PSP	NOR
ہلمند	PNN	S_LOCATION
کے	PSP	NOR
فوجی	JJ	NOR
کمانڈر	NN	NOR
تھے	VBF	NOR
-	PU	NOR

[14] https://github.com/zhongkaifu/Txt2Vec

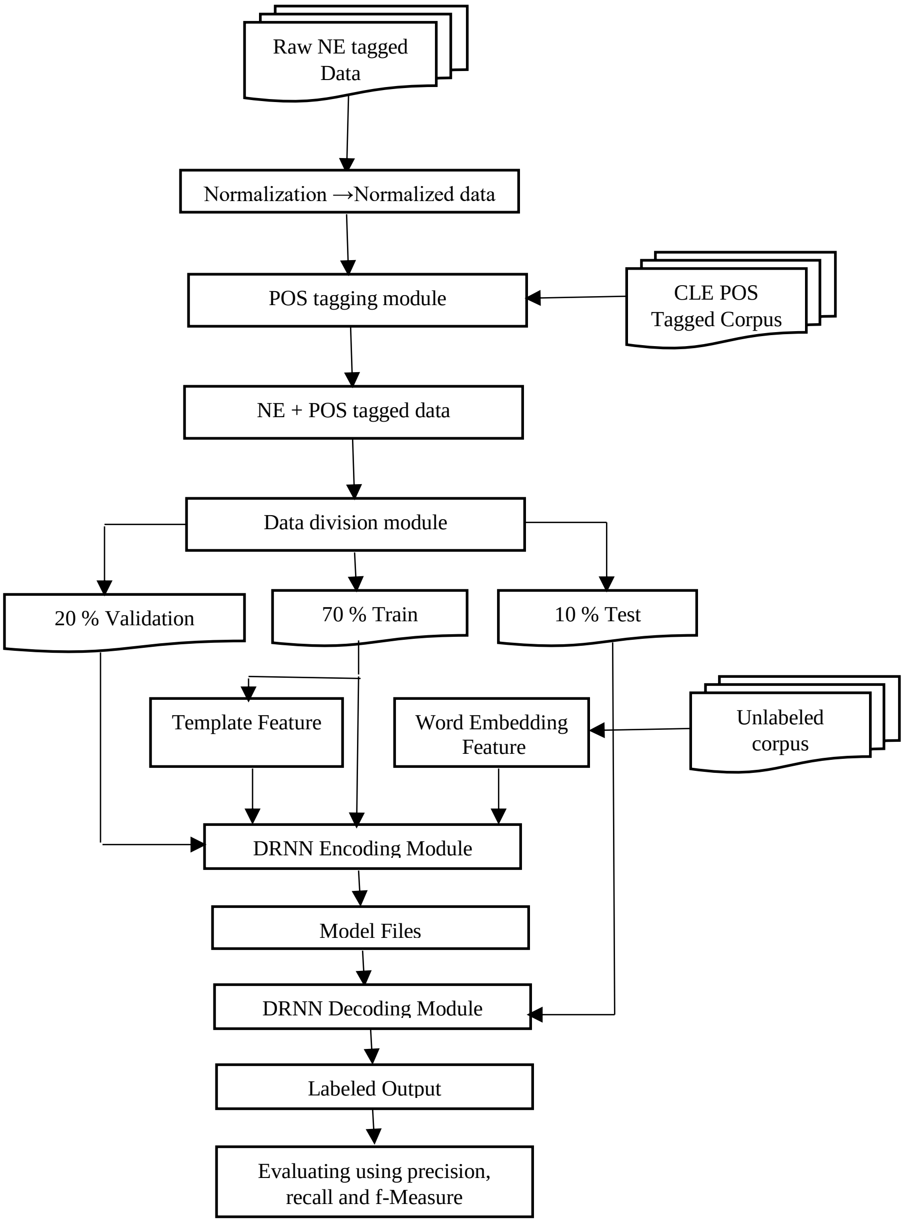


Figure 4-2: Graphical Representation of DRNN Training and Testing Process

4.3.3 Performance Evaluation Measures

In this research work, for performance evaluation of our proposed systems, we used precision, recall, and f-measure matrices.

4.3.4 Results and Discussion

Our proposed DRNN methods were tested against the CRF model mentioned in (Malik & Sarwar, 2015) and the ANN model mentioned in (Malik, 2017) for NER in Urdu and outperformed it by a significant margin. We evaluated our DRNN model on the benchmark IJCNLP-2008 dataset with seven named entity classes instead of three of the baseline approaches. We performed 10-fold cross-validation experiments for CRF and DRNN approach. For DRNN we follow the same ratio of data split as mentioned in (Dai, Li, & Xu, 2016) e.g. training (70 percent), validation (10 percent) and testing (20 percent). We used various structures of DRNN to generate results. The various structures of DRNN used include LSTM-Forward, LSTM Bi-Direction, BPTT-forward, and BPTT Bi-Direction.

Results show that our proposed DRNN supersedes baseline approaches on the benchmark data. Table 4-8 shows average precision, recall, and F-Measure of the proposed and baseline approach on the INCNLP-2008 dataset.

Table 4-10 shows average precision, recall, and F-Measure of proposed DRNN and baseline CRF models on UNER news dataset. The total number of TIME entity in sports news domain of UNER news dataset is just Ten. From machine learning perspectives, this total is very less and can adverse the performance of classifiers. Therefore, due to the low distribution of TIME entity across this domain, we considered only six named entities from sports news domain instead of seven. Due to this reason, our proposed DRNN model supersedes baseline approach relatively with high margin on the sports domain compared to national and international news domains.

Table 4-8: Average Precision, Recall & F-Measure of Proposed and Baseline approaches on IJCNLP-2008 Dataset.

Approach	Model	Precision	Recall	f-Measure
Baseline	Linear Chain CRF	43.89	61.25	47.61
	Artificial Neural Network	44.66	62.72	47.09
Proposed	LSTM-Forward RNN	54.74	59.78	51.7
	LSTM Bi-Direction RNN	58.93	61.01	57.14
	BPTT-Forward RNN	57.69	56.85	53.55
	BPTT Bi-Direction RNN	55.38	56.35	51.8

Similarly, we also checked the performance of our proposed and baseline models on Jahangir et al., datasets. Statistics shows that DRNN model beats baseline models on this dataset too. Table 4-9 shows average precision, recall, and F-Measure of proposed DRNN and baseline models on Jahangir et al., dataset.

Table 4-11shows individual entities average precision, recall and F-Measure of an IJCNLP-2008 dataset while from

Table 4-10 to Table 4-15 shows individual entities average precision, recall and F-Measure of UNER news dataset and Jahangir et al dataset.

Table 4-9: Average precision, Recall & F-Measure of proposed and baseline model on Jahangir et al dataset

Approach	Model	Precision	Recall	f-Measure
Baseline	Linear Chain CRF	71.89	69.79	70.19
	Artificial Neural Network	73.38	72.2	70.22
Proposed	LSTM-Forward RNN	81.97	79.45	79.84
	LSTM Bi-Direction RNN	79.7	75.22	76.32
	BPTT-forward RNN	79.93	76.02	77.05
	BPTT Bi-Direction RNN	77.48	71.78	72.83

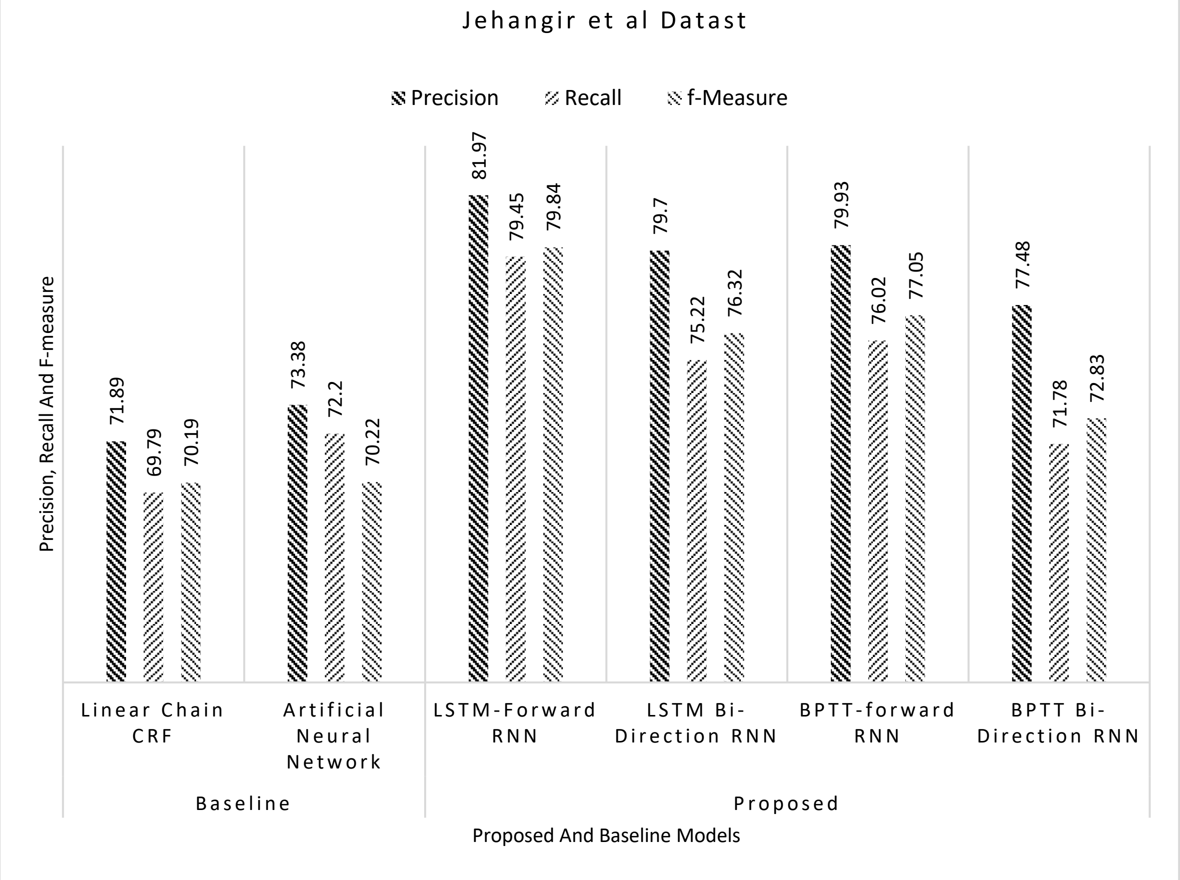


Figure 4-3: Graphical Representation of Jahangir et al dataset results

Table 4-9 presents average precision, recall, and f-Measure of DRNN, CRF and ANN Models on Jahangir et al dataset while Figure 4-3 graphically depicts its results. The highest f-Measure value is 89.54 reported by LSTM forward RNN while the lowest value is 70.19 reported by CRF. BPTT forward RNN performed comparatively and reported 77.05 f-Measure value.

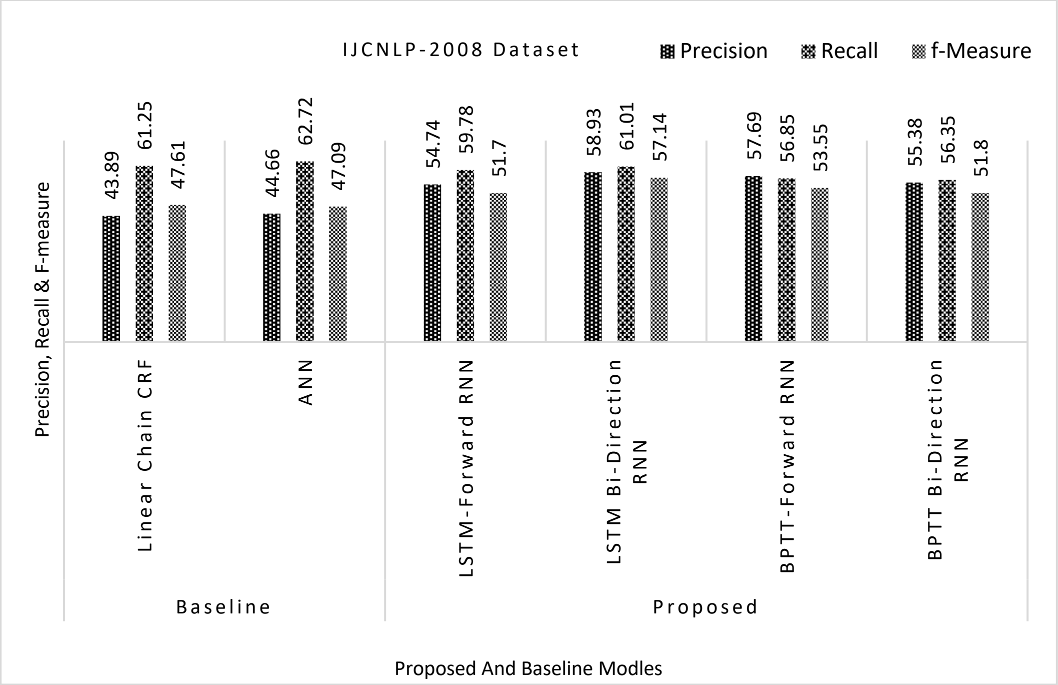


Figure 4-4:Graphical Representation of Average precision, Recall & F-measure of Proposed and Baseline approach on IJCNLP dataset

Figure 4-4 graphically depicts average precision, recall, and f-Measure of proposed and baseline approaches on IJCNLP dataset. On IJCNLP dataset the maximum f-Measure value reported is 57.14, reported by LSTM Bi-Direction RNN while the lowest f-Measure reported is 47.09 reported by ANN.

Table 4-10: Average precision, Recall & F-Measure of proposed and baseline model on UNER News dataset

Domain	Approach	Model	Precision	Recall	f-Measure
National	Baseline	Linear Chain CRF	70.54	66.84	67.27
		Artificial Neural Network	3.16	69.88	68.85
	Proposed	LSTM-Forward RNN	74.56	69.18	70.93
		LSTM Bi-Direction RNN	75.71	68.67	70.95
		BPTT-Forward RNN	72.59	66.2	68.01
		BPTT Bi-Direction RNN	75.01	66.06	68.8
International	Baseline	Linear chain CRF	64.06	56.98	56.56
		Artificial Neural Network	64.48	59.05	56.71
	Proposed	LSTM-Forward RNN	65.67	61.63	62.53
		LSTM Bi-Direction RNN	65.35	59.06	60.59
		BPTT-Forward RNN	60.8	57.72	58.44
		BPTT Bi-Direction RNN	68.97	62.22	64.13
Sports	Baseline	Linear chain CRF	81.22	79.63	79.81
		Artificial Neural Network	80.05	78.61	77.82
	Proposed	LSTM-Forward RNN	92.38	86.9	89.4
		LSTM Bi-Direction RNN	92.71	86.61	89.3
		BPTT-Forward RNN	90.87	85.58	87.99
		BPTT Bi-Direction RNN	90.11	83.35	86.34

Table 4-11: Entity Wise Statistics of average precision, recall and F-Measure of baseline and LSTM forward approach on IJCNLP-2008 dataset

Model\ Entity	ANN			Linear Chain-CRF			LSTM-Forward RNN		
	Precision	Recall	f-Measure	Precision	Recall	f-Measure	Precision	Recall	f-Measure
Person	59.4	72.21	63.82	60.86	81.84	67.86	45.79	54.54	46.41
Location	67.71	93.42	77.41	77.90	96.05	85.57	67.79	73.74	69.88
Organization	35.09	46.17	31.01	21.45	37.29	23.12	13.33	40	20
Designation	39.73	57.42	40.63	30.30	45.37	29.58	58.53	89.58	67.3
Number	31.94	54.04	33.67	31.90	49.69	34.90	60.97	38.35	38.87
Date	34.06	53.07	36	40.96	57.25	44.61	43.33	32.55	33.76

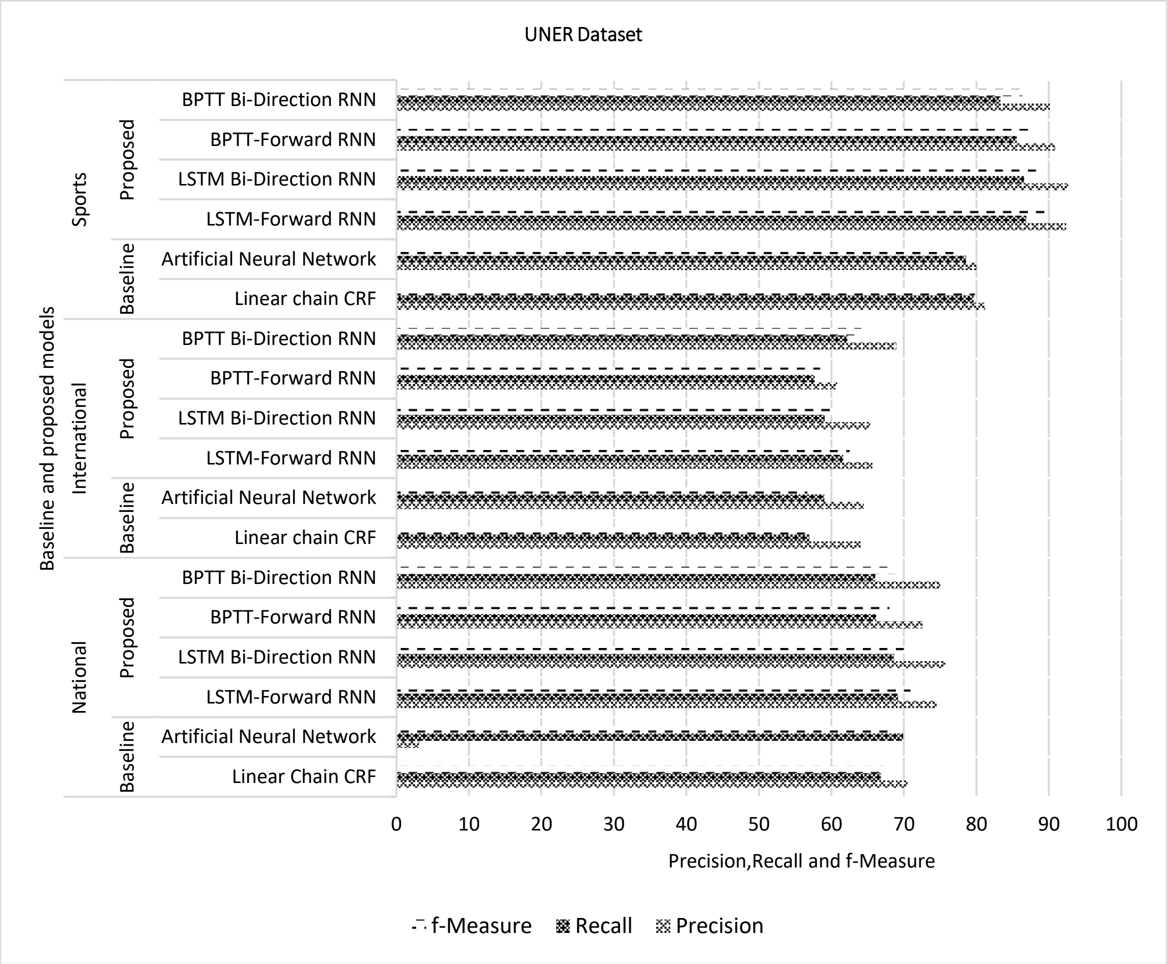


Figure 4-5: Graphical Representation of Average Precision, Recall and F-Measure of proposed and baseline model on UNER News dataset

Table 4-12: Entity Wise Statistics of average precision, recall, and F-Measure of the proposed approach on IJCNLP-2008 dataset

Model\ Entity	LSTM-Bi-RNN			BPTT-Forward RNN			BPTT-Bi RNN		
	Precision	**Recall**	**f-Measure**	**Precision**	**Recall**	**f-Measure**	**Precision**	**Recall**	**f-Measure**
Person	46.6	56.8	46.79	37.8	51.16	42.08	43.41	65.26	48.42
Location	68.1	75.33	71.19	70.96	77.93	74.13	70.15	67.42	67.66
Organization	61.11	45.56	50.79	50	45.84	47.62	63.69	36.25	42.76
Designation	68.89	77.9	71.64	69.11	76.72	65.87	48.43	79.08	59.05
Number	60.64	45.36	49.39	56.03	33.09	39.68	58.85	46.39	49.9
Date	43.61	53.78	47.09	68.34	34.88	45.92	45.59	34.67	36.37

On IJCNLP dataset for person and location entities the maximum f_measure values are 67.86 and 85.57 recorded by CRF, for organization and designation entities the maximum f_measure values are 50.79 and 71.64 reported by LSTM Bi-RNN for Number entity the maximum f_measure value is 49.9 recorded by BPTT-Bi-RNN while for date entity the maximum f-measure value is 47.09 recorded by LSTM-Bi-RNN.

Table 4-13: Average Precision, Recall and f-Measure of Person, Location and Organization in UNER and Jahangir et al datasets

Dataset	Model	Person			Location			Organization		
		Pre:	Rec:	f-M	Pre:	Rec:	f-M	Pre:	Rec:	f-M
UNER (Sport Domain)	LSTM-Forward	89.6	94.13	91.81	92.53	91.94	92.23	92.31	75	82.76
	LSTM-Bi-RNN	90.29	95.6	92.87	91.35	91.94	91.64	90.91	78.12	84.03
	BPTT-Forward RNN	89.61	92.82	91.19	89.34	91.94	90.62	85.45	73.44	78.99
	BPTT-Bi RNN	88.99	93.64	91.26	89.34	91.94	90.62	90.74	76.56	83.05
	Linear Chain-CRF	95.96	97.68	96.75	94.77	97.41	95.97	49.09	35.06	38.55
	ANN	94.94	97.07	95.9	93.5	96.75	94.94	48.84	33.66	37.03
UNER (National Domain)	LSTM-Forward RNN	75.18	80.71	77.85	69.85	70.93	70.39	81.82	83.62	82.71
	LSTM-Bi-RNN	71.65	84.01	77.34	69.77	69.77	69.77	86.95	81.42	84.09
	BPTT-Forward RNN	67.58	81.47	73.88	70.75	69.38	70.06	85.87	77.26	81.34
	BPTT-Bi RNN	70.68	81.98	75.91	68.01	71.71	69.81	84.69	81.17	82.89
	Linear Chain-CRF	69.11	68.96	68.75	72.48	74.87	73.39	71.61	67.55	69.06
	ANN	73.3	73.64	73.17	78.37	79.5	78.07	75	71.95	72.41
UNER (Inter: Domain)	LSTM-Forward RNN	49.17	69.59	57.62	70.87	80.69	75.46	79.44	58.37	67.29
	LSTM-Bi-RNN	50.24	61.99	55.5	69.76	85.64	76.89	74.41	64.08	68.86
	BPTT-Forward RNN	52	60.82	56.07	72.34	84.16	77.8	71.23	63.67	67.24
	BPTT-Bi RNN	56.4	69.59	62.3	70.56	86.63	77.77	75.59	65.71	70.3
	Linear Chain-CRF	63.91	64.82	62.58	63.68	80.02	70.28	72.22	63.97	66.48
	ANN	58.54	78.98	63.94	71.89	74.99	71.96	74.81	70.54	70.89
Jahangir et al.	LSTM-Forward RNN	81.39	83.16	81.28	78.72	83.74	81.01	79.01	77.77	77.55
	LSTM-Bi-RNN	80.31	80.1	79.96	79.65	83.98	81.62	74.69	76.64	75.18
	BPTT-Forward RNN	76.63	83.93	79.72	80.69	82.99	81.72	73.83	72.74	72.85
	BPTT-Bi RNN	78.59	81.34	79.58	78.33	84.93	81.45	74.25	71.85	72.53
	Linear Chain-CRF	71.04	70.4	70.07	70.45	75.62	72.82	73.72	65.88	68.83
	ANN	74.86	72.79	72.3	71.49	89.06	78.59	75.81	57.94	62.56

(Pre: Precision, Rec: Recall and f-M: F_Measure)

Table 4-14: Average Precision, Recall and f-Measure of Title and Number in UNER and Jahangir et al datasets

Dataset	Model	Title			Number		
		Precision	Recall	F_Measure	Precision	Recall	F_Measure
UNER Sport Domain)	LSTM-Forward RNN	96.15	83.33	89.28	90.68	88.12	89.38
	LSTM-Bi-RNN	100	76.67	86.79	92.79	88.44	90.56
	BPTT-Forward RNN	100	83.33	90.91	90.32	87.5	88.89
	BPTT-Bi RNN	95.65	73.33	83.02	90.55	86.88	88.68
	Linear Chain-CRF	94.94	94.12	93.70	87.66	96.34	91.53
	ANN	94.03	92.88	92.25	84.98	95.39	89.49
UNER (National Domain)	LSTM-Forward RNN	73.27	60.66	66.37	68.92	82.26	75
	LSTM-Bi-RNN	76.47	63.93	69.64	68.87	83.87	75.63
	BPTT-Forward RNN	69.16	60.66	64.63	65.61	83.06	73.31
	BPTT-Bi RNN	68.75	54.1	60.55	70.34	82.26	75.83
	Linear Chain-CRF	70.41	59.94	64.5	64.32	80.22	70.43
	ANN	70.85	51.79	58.31	55.44	78.46	63.22
UNER (Inter: Domain)	LSTM-Forward RNN	60.78	63.27	62	65.43	67.09	66.25
	LSTM-Bi-RNN	53.19	51.02	52.08	68.29	70.89	69.57
	BPTT-Forward RNN	59.52	51.02	54.94	63.22	69.62	66.27
	BPTT-Bi RNN	62.22	57.14	59.57	67.09	67.09	67.09
	Linear Chain-CRF	63.66	45.94	50.28	52.50	70.45	58.10
	ANN	60.5	55.24	53.22	65.51	63.85	60.5

Table 4-15: Average Precision, Recall and f-Measure of Date and Time in UNER and Jahangir et al datasets

Dataset	Model	Date			Time		
		Precision	Recall	F_Measure	Precision	Recall	F_Measure
UNER Sport Domain)	LSTM-Forward RNN	93.02	88.89	90.91	---	----	---
	LSTM-Bi-RNN	90.91	88.89	89.89	---	----	---
	BPTT-Forward RNN	90.48	84.44	87.36	---	----	---
	BPTT-Bi RNN	85.37	77.78	81.4	---	----	---
	Linear Chain-CRF	64.95	57.17	58.81	---	----	---
	ANN	64.01	55.9	57.32	---	----	---
UNER (National Domain)	LSTM-Forward RNN	74.47	41.67	53.44	78.38	64.44	70.73
	LSTM-Bi-RNN	80.39	48.81	60.74	75.86	48.89	59.46
	BPTT-Forward RNN	77.27	40.48	53.13	71.88	51.11	59.74
	BPTT-Bi RNN	84.31	51.19	63.7	78.26	40	52.94
	Linear Chain-CRF	70.07	60.45	63.92	75.77	55.89	60.84
	ANN	77.91	62.67	67.45	81.22	71.13	69.36
UNER (Inter: Domain)	LSTM-Forward RNN	84	62.38	71.59	50	30	37.5
	LSTM-Bi-RNN	87.69	56.44	68.68	53.85	23.33	32.56
	BPTT-Forward RNN	76.54	61.39	68.13	30.77	13.33	18.6
	BPTT-Bi RNN	90.91	59.41	71.86	60	30	40
	Linear Chain-CRF	81.66	55.82	63.57	50.77	17.83	24.60
	ANN	75.79	49.25	56.52	75.79	49.25	56.52

This study shows that with the use of various DRNN models' architecture with sparse and dense features we can obtain better NER tagging results for Urdu compared to the traditional CRF with only language-independent features. Results of (Malik and Sarwar, 2015), The baseline work of this study based on linear chain CRF and vanilla ANN, lacks comprehensiveness. Their work does not make use of language-dependent features, for example, POS tags, thus the tagging accuracy of the CRF model may suffer. Also, DRNN assigns a class label to the word in real time fashion e.g. assigns tag on the strategy when and where a word is seen while CRF assigns tags after reading the whole sequence (Yao et al., 2014). In CRF the input and output are directly connected without any middle layers while in RNN inputs are connected with output through recurrent cells (Huang, Xu, & Yu, 2015). RNN holds each word of training data as a high dimensional real-valued vector and in this vector representation, chances that similar words will be close to each other increases. And this phenomenon maintains the relationship between similar words. Thus, the scenario of word vector space representation goes to better performance for analogous words in analogous linguistic context(Yao et al., 2014).

Our Ten percent test data comprises of total 876 NEs, entity wise statistics can be found in Table 4-16.

Table 4-16: Test Data NE Statistics

Person	Location	Organization	Designation	Number	Date	Time
171	201	245	49	79	101	30

After testing we observed that our proposed model confused mostly person NEs with location and organization NEs, similarly Organization NE is confused with person and location NEs. Details of most confused tags using RNN-BPTT Bidirectional is given in Table 4-17.

Table 4-17: Confusion Matrix of BPTT Bi-Direction RNN on International News Domain of UNER dataset

	Person	Location	Organization	Designation	Number	Date	Time
Person	119	16	24	8	1	3	0
Location	8	174	7	1	11	0	0
Organization	48	25	161	5	6	0	0
Designation	13	3	5	28	0	0	0
Number	9	8	2	2	53	1	4
Date	10	20	6	0	3	60	2
Time	4	1	8	1	5	2	9

The majority of these confused tag errors are due to the phenomenon that in Urdu a single word can acquire a different meaning in different context. For example, the Urdu word (دو, du) can be a part of Number entity (دوہزار, Two Thousand), it can be a part of Date entity e.g (سنہ دو ہزار پانچ, 2005 AD), it can be a part of Time Entity e.g (دو بجے, two o Clock) and it can also be used as a normal term. In our original test data, the word (دو, du) appears twenty times. It acquired 7 times date tag, 7 times number tag, two times Time tag and four times NOR tag. After testing the same word acquired one-time person and organization tag, eight times number tag, five times date tag, one-time time tag and four times NOR tag. Table 4-18 shows original statistics of words that acquire multiple tags in a different context in test data while Table 4-19 show statistics after testing.

Table 4-18: Original Statistics of multiple tags acquiring word in original Test data

Test Token	Person

We also found that these models have a competitive performance with each other. We make use of word embedding as a feature along with both language dependent and non-dependent features and hypothesize that this accounts for the improved performance over Linear chain CRF and ANN.

About this Chapter

This chapter explores the third contribution namely the development and release of UNER dataset. In this chapter, we provided the detailed information related to UNER dataset development process and its comparison with the already existed dataset. Additionally, we also highlighted the various segment representation techniques with which this new dataset is compatible.

Urdu Named Entity Dataset

Gold standard data is extremely important for automatic Urdu language processing. The performance of ML approaches in the context of NLP is strongly dependent on the availability of good quality of linguistic resources. Supervised ML model utilizes pre-labeled data in training phase to induce a pattern from it therefore, it is obligatory to put up related pre-labeled training data and in a bulky quantity (Brodley & Friedl, 1999). The Urdu, the national language of Pakistan and the most widely spoken and understandable language of South Asia suffers from a shortage of linguistic resources for the development of sophisticated NLP tools and conducting experiments.

In whole south Asia, the Center for Language Engineering (CLE)[15] in Pakistan is the only organization whose entire focus is on the development of various linguistic corpora to promote research in Pakistani as well as many other Urdu like languages in the region. To encourage the researchers of ULP research community, CLE has set up numerous linguistic corpora for automated Urdu language processing. Particulars of the obtainable CLE linguistic corpora can be ascertained on the source URL of CLE store: http://www.cle.org.pk/clestore/index.htm. The only limitation of CLE linguistic resources is its processing charges.

5.1 Importance of Tagged corpora

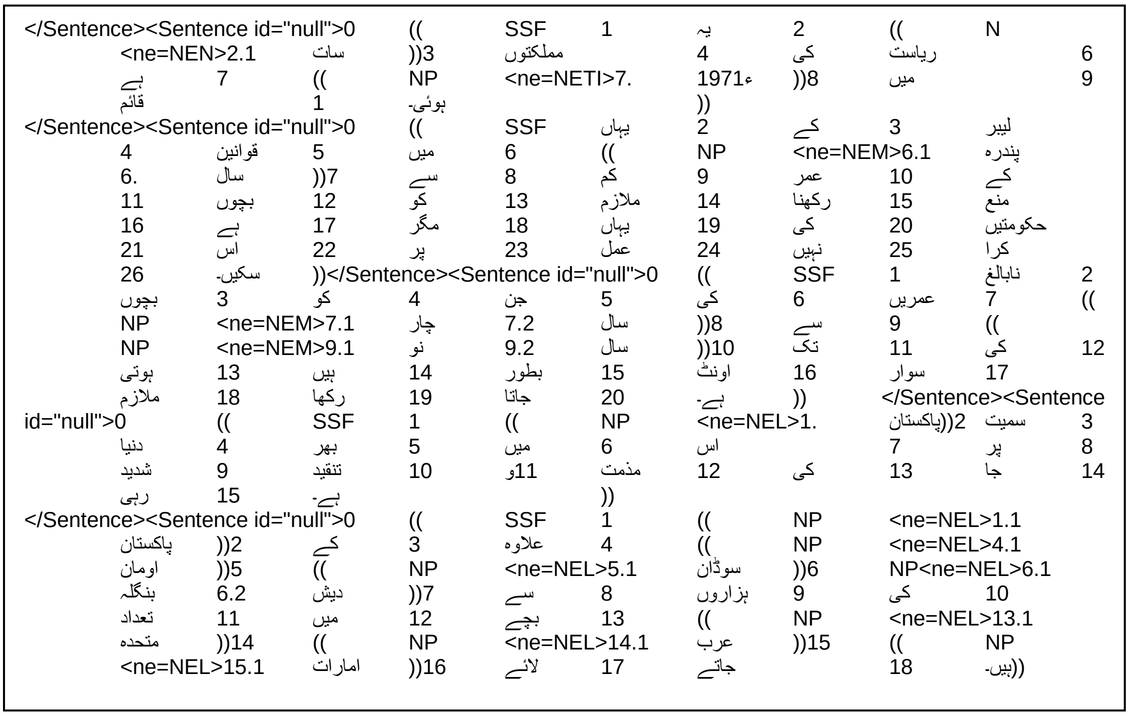
Compared to un-annotated corpus a tagged corpus is well-lined of expedient information. This linguistic information obscured in theses tagged corpus can be exploited to anticipate the detailed quantitative analysis of text data with the help of appropriate ML approaches. The tags in these tagged corpora can be conceived a piece of code appended with text to represent an individual feature or set of features.

[15] http://www.cle.org.pk/

5.2 IJCNLP-2008 NE tagged dataset

In 2008 the IJCNLP take initiative steps, and invited NLP researchers from NLP research community to build automated NER system as IJCNLP-08 shared task for five most popular and most dominantly spoken language of South Asia. The five languages chosen are Bengali, Hindi, Telugu, Oriya, and Urdu. For the development of automated NER system, all participants are restricted to use the training data released by IJCNLP for the same task. The IJCNLP released training data is manually annotated data with twelve NE classes and its size is about 40,000 words. The research group namely Center for Research in Urdu Language Processing (CRULP) at National University of Computer and Emerging Sciences in Pakistan and IIT Hyderabad, India have jointly made efforts to create the IJCNLP-2008 dataset, after the creation, they donated it to the NER workshop of IJCNLP (Hussain, 2008). The IJCNLP-2008 dataset is considered the ever first Urdu dataset that is manually annotated with explicitly named entity classes. This dataset is freely available for conducting research.

The IJCNLP-2008 dataset contains a lot of information about sentences structure, words, and named entities. In this dataset text is not arranged sentences wise therefore the annotators used an explicit "</Sentence>" tag to show the begin of a sentence. Similarly, within a sentence, a unique index number is assigned to each word also the named entities within sentences are enclosed in double parentheses like "((NP <ne=NEL>1. پاکستان))". Here the NE tag "NP <ne=NEL>" represents that this entity belongs to Location class while the term (" پاکستان") shows the actual entity. Similarly, the numbers attached with each word are its index number within a sentence. Screenshot of original text within IJCNLP-2008 dataset can be seen in Figure 5-2

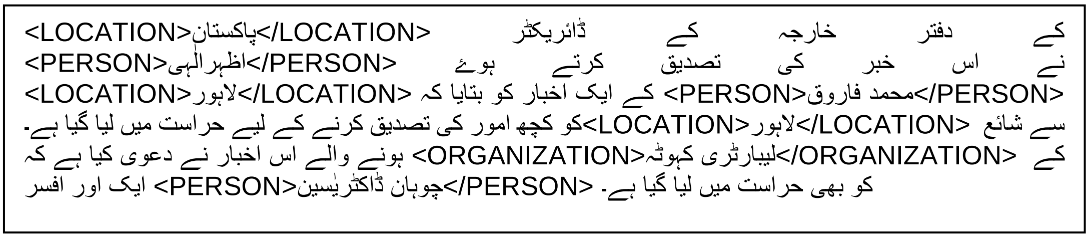


```
</Sentence><Sentence id="null">0    ((    SSF    1    یہ    2    ((    N
    <ne=NEN>2.1    سات    ))3    مملکتوں        4    کی    ریاست        6
    ہے    7    ((    NP    <ne=NETI>7.    1971ء    ))8    میں        9
    قائم        1    ہوئی۔            ))
</Sentence><Sentence id="null">0    ((    SSF    یہاں    2    کے    3    لیبر
    4    قوانین    5    میں    6    ((    NP    <ne=NEM>6.1    پندرہ
    6.    سال    ))7    سے    8    کم    9    عمر    10    کے
    11    بچوں    12    کو    13    ملازم    14    رکھنا    15    منع
    16    ہے    17    مگر    18    یہاں    19    کی    20    حکومتیں
    21    اس    22    پر    23    عمل    24    نہیں    25    کرا
    26    سکیں۔    ))</Sentence><Sentence id="null">0    ((    SSF    1    نابالغ    2
    بچوں    3    کو    4    جن    5    کی    6    عمریں    7    ((
    NP    <ne=NEM>7.1    چار    7.2    سال    ))8    سے    9    ((
    NP    <ne=NEM>9.1    نو    9.2    سال    ))10    تک    11    کی    12
    ہوتی    13    ہیں    14    بطور    15    اونٹ    16    سوار    17
    ملازم    18    رکھا    19    جاتا    20    ہے۔    ))    </Sentence><Sentence
id="null">0    ((    SSF    1    ((    NP    <ne=NEL>1.    پاکستان))2    سمیت    3
    دنیا    4    بھر    5    میں    6    اس        7    پر    8
    شدید    9    تنقید    10    و11    مذمت    12    کی    13    جا    14
    رہی    15    ہے۔            ))
</Sentence><Sentence id="null">0    ((    SSF    1    ((    NP    <ne=NEL>1.1
    پاکستان    ))2    کے    3    علاوہ    4    ((    NP    <ne=NEL>4.1
    اومان    ))5    ((    NP    <ne=NEL>5.1    سوڈان    ))6    NP<ne=NEL>6.1
    بنگلہ    6.2    دیش    ))7    سے    8    ہزاروں    9    کی    10
    تعداد    11    میں    12    بچے    13    ((    NP    <ne=NEL>13.1
    متحدہ    ))14    ((    NP    <ne=NEL>14.1    عرب    ))15    ((    NP
    <ne=NEL>15.1    امارات    ))16    لائے    17    جاتے        18    ہیں۔))
```

Figure 5-2: Screenshot of IJCNLP dataset

5.3 Jehangir et al Dataset

The Second NE tagged dataset which we as guideline, is a portion of the dataset which is used in the research work of (Jahangir et al., 2012). The corresponding author of the work (Jahangir et al., 2012) donated their dataset after requesting the dataset for research purpose usage. Since the dataset used in the work of Jahangir et al., is no more available from the source, therefore we refer this dataset as Jahangir et al., dataset. Jahangir et al., the dataset is a dataset of about 31860 words with total 1526 named entities. The dataset is annotated with four named entity classes. The statics are extracted from the available version of the dataset with authors of this work. Screenshot of the Jahangir et al., the dataset can be observed in Figure 5-1.



کے دفتر خارجہ کے ڈائریکٹر <LOCATION>پاکستان</LOCATION>
نے اس خبر کی تصدیق کرتے ہوئے <PERSON>اظہرالٰہی</PERSON>
<PERSON>محمد فاروق</PERSON> کے ایک اخبار کو بتایا کہ <LOCATION>لاہور</LOCATION>
سے شائع <LOCATION>لاہور</LOCATION>کو کچھ امور کی تصدیق کرنے کے لیے حراست میں لیا گیا ہے۔
کے <ORGANIZATION>لیبارٹری کہوٹہ</ORGANIZATION> ہونے والے اس اخبار نے دعوی کیا ہے کہ
کو بھی حراست میں لیا گیا ہے۔ <PERSON>چوہان ڈاکٹرلیٰسین</PERSON> ایک اور افسر

Figure 5-2: Screenshot of Jahangir e al dataset

5.4 The UNER Dataset

Named entity recognition and classification is a very crucial task in Urdu. One challenge among the others which makes the Urdu NER task complex is the non-availability of enough linguistic resources. The NER research for English and other Western languages has a long tradition and a significant amount of work has been done to solve NER problems in these languages. From resource availability aspect, Western languages are counted resource plentiful languages. On the other hand, Urdu lags far behind in terms of resources.

Before the release of UNER dataset, ULP researchers working on NER was limited to two datasets, the IJCNLP-2008 NE tagged dataset and Jahangir et al., dataset (Jahangir et al., 2012). Therefore, in this thesis, we reported the development of NE tagged dataset for automated NER research in Urdu, especially with machine learning (ML) perspectives. The new developed Urdu NER dataset contains about 48000 words, comprising of 4621 named entities of seven named entity classes. The contents source of this new dataset is BBC Urdu and initially contains data from sport, national and international news domain. This new dataset can be used for training and testing purpose of various statistical and machine learning models such as e.g. hidden Markov model (HMM), maximum entropy (ME), Conditional random field (CRF), recurrent neural network (RNN) and so forth for conducting computational NER research in Urdu. Our goal is to make this dataset freely and widely acquirable and to promote other researchers to exercise it as a criteria testbed for experimentations in Urdu NER research.

5.4.1 Development

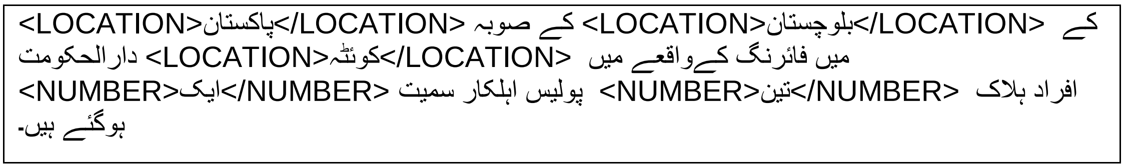
Linguistic resources for most southeastern languages are not radially available due to which these languages are termed scared resource languages. Urdu is a southeastern language and is spoken in a vast area of sub-continent. It is a low resource language. Due to resource scarceness plenty of research work has not been actioned for Urdu language (Mukund et al., 2010).

The dataset we present contains all text from BBC Urdu cyberspace. The current version of UNER NE tagged dataset contains text from three news domains e.g. 1) national news 2) international news 3) sports news. In the future, we will include text from other popular news domains e.g. entertainment, science, business, health not only form BBC Urdu but from other sources such as Express news, Dunia news etc. The size of our current NER dataset is about 48 thousnad words with total of 4621 named entities. Entities in the text are manually tagged in the guideline of IJCNLP-2008 and Jahangir et al., dataset. Initially, only seven named entity classes are used in tagging. The seven named entity classes used in tagging include PERSON, LOCATION, ORGANIZATION, DESIGNATION, NUMBER, DATE, TIME. After manual tagging, samples from the three domains are reviewed through Urdu linguistic experts from two different organizations, and changes mentioned by them have incorporated accordingly in the whole dataset. During the development process all the entities are tagged from right to left and also text is stored sentence level. The symbols "-" dash and "?" are used as sentence separators. During the tagging, the entity is enclosed in start and end tags. E.g. The entity (پاکستان, Pakistan) where it occurs in the text is tagged with start and end tags of LOCATION such as <LOCATION>پاکستان</LOCATION>. For all the seven-entity class labels, the same approach is adopted. For storage purpose, we used notepad with UTF-8 encoding system. Text in files is organized sentence wise because most of machine learning models take inputs sentence wise. E.g. CRF, RNN, DRNN and so on.

The UNER NE tagged dataset we reported is freely available[16] for research purposes. Although UNER NE tagged dataset can be used for rule-based work, but its structure and organization is more feasible for machine learning based approaches. The two main fascinating aspects of the UNER dataset is its larger size and it's content storage in preprocessed form. Since larger training data is mandatory for better performance of supervised ML models and the UNER datasets address this requirement of ML models to some extent. Similarly, before training process modeling training data in the required for the format of ML is necessary. However, the existence of unnecessary data such as special characters, emojis, hyperlinks, XML tags makes training data modeling a challenging task. So far, the UNER dataset is concerned its contents are preprocessed and very minimum efforts are required to model it in any format. We hope that our NER dataset will help in promoting ML-based research in Urdu particularly in NER task. The distinguishing features of the UNER dataset are listed below:

- Freely available
- Large size comparatively to IJCNLP-2008 and Jahangir el al datasets
- In annotation, seven entity classses are used
- Domain wise contents storage while other datasets lack this
- Easy data modeling as the text is stored in preprocessed form. E.g. It does not contain irrelevant text such as XML tags, emojis, headers, links, and special characters

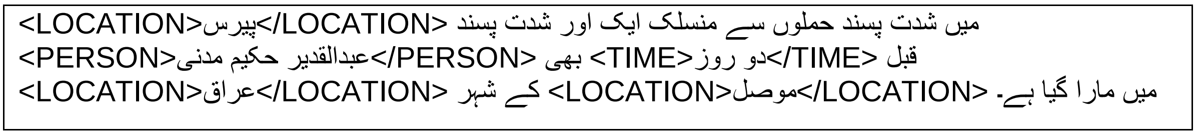
Below Figure 5-3, Figure 5-4 and Figure 5-5 shows screenshots of a single sentence from each news domain in which named entities are labeled with its corresponding class label.



<LOCATION>پاکستان</LOCATION> کے صوبہ <LOCATION>بلوچستان</LOCATION> کے
دارالحکومت <LOCATION>کوئٹہ</LOCATION> میں فائرنگ کےواقعے میں
<NUMBER>ایک</NUMBER> پولیس اہلکار سمیت <NUMBER>تین</NUMBER> افراد ہلاک
ہوگئے ہیں-

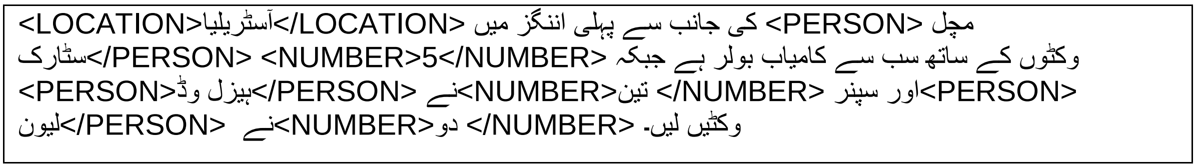
Figure 5-3: Screenshot of National News Domain

[16] https://drive.google.com/open?id=0B2kus5E2eIJQNUNzRjR5eFBMZkk



<LOCATION>پیرس</LOCATION> میں شدت پسند حملوں سے منسلک ایک اور شدت پسند
<PERSON>عبدالقدیر حکیم مدنی</PERSON> بھی <TIME>دو روز</TIME> قبل
<LOCATION>عراق</LOCATION> کے شہر <LOCATION>موصل</LOCATION> میں مارا گیا ہے۔

Figure 5-4: Screenshot of International News Domain



<LOCATION>آسٹریلیا</LOCATION> کی جانب سے پہلی اننگز میں <PERSON> مچل
سٹارک</PERSON> <NUMBER>5</NUMBER> وکٹوں کے ساتھ سب سے کامیاب بولر ہے جبکہ
<PERSON>ہیزل وڈ</PERSON> نے <NUMBER>تین </NUMBER> اور سپنر <PERSON>
لیون</PERSON> نے <NUMBER>دو </NUMBER> وکٹیں لیں۔

Figure 5-5: Screenshot of Sports News Domain

Details of our presented UNER dataset can be found in Table 5-1 and Table 5-2. Consolidated statistics such as total number of words, a total number of comprising named entities and the total number of sentences are provided in Table 5-1. Domain wise consolidated statistics of each named entity class are provided in Table 5-2.

Table 5-1: Consolidated Statistics of UNER dataset

Total of No. of Words	48673
Total No. of sentences	1744
Total No. of Named Entities	4621

Table 5-2: Domain wise consolidated statistics of each entity class

Entity\Domain	**National**	**International**	**Sport**	**Total**
Person	401	201	605	**1207**
Location	390	360	455	**1205**
Organization	400	210	53	**663**
Designation	167	70	42	**279**
Number	270	132	589	**991**
Date	81	74	48	**203**
Time	40	23	10	**73**
Total	**1749**	**1088**	**1809**	**4621**

Table 5-3 provides lists of typical named entity types that are considered during the construction process of our NE tagged dataset along with its description. We are intend to balance the dataset with consideration of genre and proportion of each entity class. Table 5-2 clearly reflects that the proportion of DATE and TIME entity classes are quite small compared to others because the occurrence and mentioning these two entities in national, international and sports news are not customary. After the annotation process, the whole dataset is stored in 150 notepad documents using the UTF-8 encoding scheme.

Table 5-3: List of Generic Urdu Named Entity Types with the kind of Entities they refer.

Type	**Tag**	**Sample Category**
Person	<PERSON>	Individuals, small groups
Location	<LOCATION>	Territory, land, kingdom, mountains, site, locality etc
Organization	<ORGANIZATION>	firms, a group of players, Political parties, bureau etc
Designation	<DESIGNATION>	Various designations e.g. Professor, Dean, Mufti, Captain etc.
Number	<NUMBER>	Counts e.g. Hundred, Ten Thousand One, 10 million etc.
Date	<DATE>	Date stamps
Time	<TIME>	Clock time stamps

Table 5-4 provides domain wise document detail of UNER dataset. The national news domain contents are stored in 60 notepad documents, sports domain contents are stored in 50 documents while international news domain contents are stored in 40 documents.

Table 5-4: Domain wise No. of Documents

Domain	File No.	No. of Document
National	1-60	60
Sports	61- 110	50
International	111- 150	40
Total		150

Table 5-5 presents the example of the seven named entity classes of the UNER dataset. For example, the word اقوام متحدہ represent organization entity and therefore it is annotated with organization entity class label. Similarly, the Urdu word نو مارچ represents date entity and hence it is tagged with date entity class. In the same way, the word عمران خان acquires person label, وزیراعلیٰ attains designation label, 5 acquires number label and the word دو بجے attains time entity class.

Table 5-5: Example of various NE tags

Class	Tag	Example
Person	<PERSON>	<PERSON>عمران خان</PERSON>
Location	<LOCATION>	<LOCATION>کراچی</LOCATION>
Organization	<ORGANIZATION>	<ORGANIZATION> اقوام متحدہ</ORGANIZATION>
Designation	<DESIGNATION>	<DESIGNATION>وزیراعلیٰ </DESIGNATION>
Number	<NUMBER>	<NUMBER>5</NUMBER>
Date	<DATE>	<DATE>نو مارچ</DATE>
Time	<TIME>	<TIME>دو بجے</TIME>

Figure 5-6 represents UNER dataset graphically. From the graph, it is clear that more than half of the total entities consist of person and location entities e.g. 26% occurrence of each. The number entity makes the third major share (22%) of it while the organization entity occurrence is 14% in UNER dataset. The lowest share is 2%, representing the time entity while the percentage figures 6% and 4% represents designation and date entities respectively.

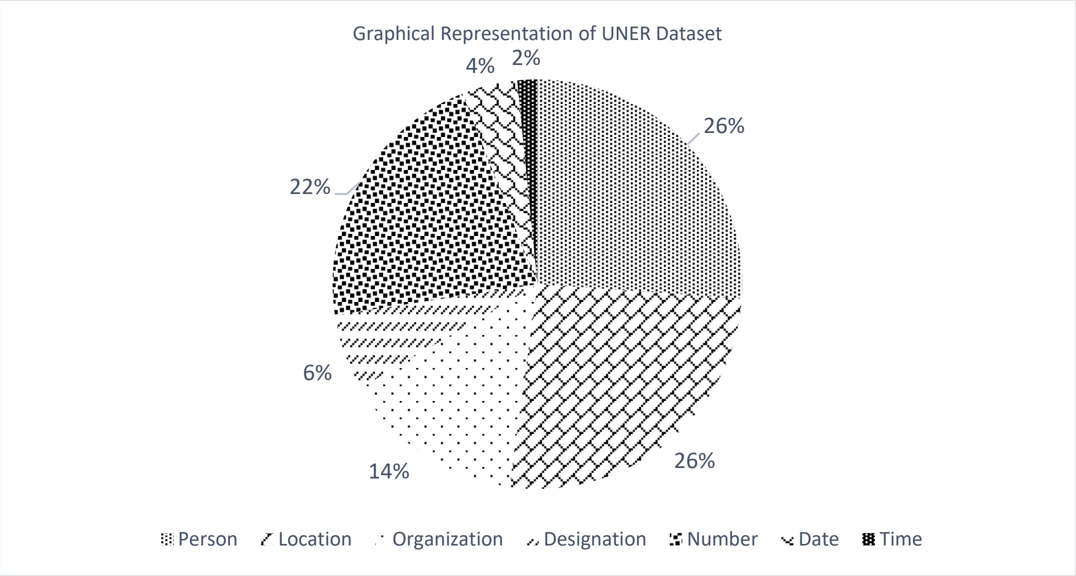


Figure 5-6: Graphical Representation of UNER Dataset

5.4.2 Segment Representation Techniques

Most state-of-the-art ML models require training data to be formulated in some feasible segment representation (SR) technique(Keretna, Lim, Creighton, & Shaban, 2015). SR techniques have been applied to a variety of NLP tasks such as POS, NER and word segmentation (Wu, 2014). Selection of an expedient SR approach has a significant impact on the detection results of classification models. Normally in classification task, SR of text is carried out as a pre-processing step. The underlying theme of incorporating SR approaches is to cater all words in a sentence with a label and tag. The main objective of the label is to depict the class that the word dwells to, while the tag points the positioning of the word in the class, as classes, can be shaped employing more than one word. In case of named entity task example of various classes can be PERSON, LOCATION, ORGANIZATION, DESIGNATION, NUMBER, DATE, TIME etc. while “Begin”, “Middle” and “End” are the example of labels. In literature, the process of text representation into its segments is referred with various names like chunk representation, chunk encoding, chunk tagging. In literature, the most widely adopted SR techniques reported thus far included: IOBES (Inside.Outside/Begin/End/Single), IO(Inside/Outside), IOB2 (Inside/Outside/Begin version-2), IOE2 (Inside/Outside/End version-2), and SBME (Single/Begin/Middle/End) (Keretna et al., 2015).

The most widely used SR approaches in NER task are IOB2, SBME, and IOBES. The UNER dataset is fully compatible with the above-mentioned SR representation approaches and one can easily represent the text of UNER dataset into any above-mentioned SR representation formats. Table 5-6 provides an overview of the various SR format of text in UNER dataset.

Table 5-6: Example of Various SR approaches in the task of NER

Token	**SBME**	**IO**	**IOB2**	**IOBES**
الیکشن	B_Organization	I_Organization	B_Organization	B_Organization
کمیشن	E_Organization	I_Organization	I_Organization	E_Organization
کے	NOR	O	O	O
وکیل	S_Designation	I_Designation	B_Designation	S_Designation
سلمان	B_Person	I_Person	B_Person	B_Person
اکرم	M_Person	I_Person	I_Person	I_Person
راجہ	E_Person	I_Person	I_Person	E_Person
نے	NOR	O	O	O
دلائل	NOR	O	O	O
دیتے	NOR	O	O	O
ہوئے	NOR	O	O	O
کہا	NOR	O	O	O

In case of SBME segment representation approach “S” means that this entity is single, “B” represent the beginning of an entity, “M” represents the middle portion of an entity, the letter “E” represents the ending of an entity while NOR represents that token is a regular term. In IO SR representation, all named entities are tagged with “I” tag while all other regular terms acquires the “O” tag, in IOB2 scheme begin of an entity is mentioned with “B” tag all other parts of the concerned entity are tagged with “I” tag and the “O” tag is used for all other regular terms similarly in an IOBES scheme the “S” represents isolated or individual entities, in this scheme beginning of an entity is marked with “B” while the ending boundaries of entities are marked with “E” tag and middle terms are marked with “I” tag.

The IO scheme is the first SR representation scheme and is considered the simplest one. In IO representation, the only two labels used are “I” and “O”, this scheme lacks NE boundary recognition capability of two consecutive classes. To overcome the limitations of IO, the scheme of IOB1 was presented. In the IOB scheme, all words acquire some sort of tags. Beginning of any is mentioned with the letter “B”, while all other words of an entity of any particular class are represented with “I” and all regular terms are represented with the tag “O”. The latest SR scheme is IOBES, it is similar to other schemes except in this new scheme two additional labels are added. The additional labels are E and S, E represents the ending boundary of an entity while S is used to show single entities.

5.5 Comparative analysis of UNER dataset

In Table 5-7 comparative statistics of UNER and other available named entity dataset are provided. The UNER dataset comprised of 48673 words while IJCNLP and Jahangir et al possess 40408 and 31860 words respectively. Similarly, from a total number of named entities the UNER dataset holds more named entities when compared with IJCNLP and Jehangir et al dataset. The UNER dataset contains total of 4621 named entities while the other two contains 1521 and 1114 named entities respectively. So far, the total number of sentences are concerned the UNER dataset contains more sentences as compared to other two datasets. Our dataset is comprised of total 1744 sentences while the IJCNLP and Jahangir et al dataset hold 1097 and 1315 sentences.

Table 5-7: Comparative statistics of UNER dataset with IJCNLP and Jehangir et al datasets

Dataset	No. of Words	No. of Sentences	No. of NEs
UNER News Dataset	48,673	1,744	4,621
Jahangir et al.,	31,860	1,315	1,526
IJCNLP-2008	40,408	1,097	1,115

Table 5-8 provides domain wise statistics of UNER dataset. The sports news portion of UNER dataset is comprised of total 15577 words, 602 sentences and a total of 1802 named entities, the national news portion of it is comprised of 19589 words, 607 sentences and a total of 1749 entities while the international news portion is containing 13507 words, 535 sentences, and 1070 named entities.

Table 5-8: Domain wise statistics of UNER dataset

Dataset	No. of Words	No. of Sentences	No. of NEs
Sports	15,577	602	1,802
National	19,589	607	1,749
International	13,507	535	1,070
Total	**48,673**	**1,744**	**4,621**

Table 5-9 presents domain wise entity statistics of UNER dataset. All the three portion the UNER dataset contains total of 1207 person entities, 1205 location entities, 663 organization entities, 203 date entities, 73-time entities, 991 number entities, and 279 designation entities.

Table 5-9: Domain wise Entity distribution of UNER dataset

Domain\ Entity	Per:	Loc:	Org:	Dt:	Time	Num:	Desig:	Total
Sports	605	455	53	48	10	589	42	1,802
National	401	390	400	81	40	270	167	1,749
International	201	360	210	74	23	132	70	1,070
Total	**1,207**	**1,205**	**663**	**203**	**73**	**991**	**279**	4,621[17]

Table 5-10 provides detailed comparative statistics of the UNER and other available NE tagged datasets. The number occurrence of person entity in UNER dataset is 1207 while in IJCNLP and Jahangir et al datasets its occurrence is 277 and 380 times. The location entity occurrence in UNER dataset is 1205 times while in IJCNLP and Jahangir et al datasets its occurrence is 490 and 756 times. Similarly, the UNER dataset contains 203-times date entity while IJCNLP and Jahangir et al datasets its occurrence is 123 and 101 times. The Jahangir et al dataset lacks number and designation entities while IJCNLP dataset lacks the time entities.

Table 5-10: Entity wise statistics of each dataset

Dataset	Per:	Loc:	Org:	Dt:	Time	Num:	Desig:
Jahangir et al.,	380	756	282	101	7	---	---
UNER News	1,207	1,205	663	203	73	991	279
IJCNLP-2008	277	490	48	123	---	108	69

We believe that UNER dataset is a rich dataset and has the ability to compensate the indigence of NER and NLP research, utilizing the present-day Urdu.

5.4 Chapter Summary

These days the state of the art approaches that are widely adopted around the globe for the development of NER tools in almost all languages including Urdu, are machine learning approaches. The core reason behind its wide usage is based on four features: a) the capability of automatic learning b) the degree of accuracy c) the speed of processing and d) generic nature. The basic need for ML approaches for training and testing is the availability of pre-NE tagged dataset. As far as Urdu is concerned, it is termed as resource-poor language. Therefore, in this work we tried to contribute in Urdu language resource with a large enough newly created NE tagged dataset. Significant efforts were made to build this huge NE tagged dataset compared to existing one with text from multi news domains. The fascination aspect of the UNER dataset is its size as well as very rich NE contents. These two aspects make UNER dataset more feasible

[17] *(Per: Person, Loc: Location, Dt: Date, Num: Number, Desig: Designation)*

for ML techniques. We hope that this new dataset will spark a light in the ULP research community and will attract researchers in the future to promote research in ULP.

About this Chapter

In this chapter, we confront the conclusions of this thesis regarding the results prevailed in POS and NER, the two most basic NLP tasks with help of machine learning and deep learning approaches, to highlight the accomplished objectives and prima contribution. Additionally, several latent future works in the subject area is accounted, that is to say, to improve the feature set for POS and NER task and extending our framework by adding more NLP tasks such supervided learning based Urdu word segmentation, Automatic parsing, and automatic stemming modules.

Conclusion and Future Work

This thesis reports the development of a generic Urdu NLP framework for POS and NER, considered as two fundamental NLP tasks. Additionally, the development of newly created dataset for NER task from machine learning perspectives is also reported.

6.1 Conclusion

In this thesis, we have proposed generic Urdu NLP framework for carrying two most basic and interrelated NLP tasks e.g. the POS and NER, with the help of machine learning and deep learning models.

The first module of our work provides a solution to the Urdu POS task. We presented the CRF model as an alternative to the state-of-the-art model (SVM) to Urdu POS tagging. We demonstrated the impact of our proposed feature set on the performance of CRF experimentally by comparing it with SVM of baseline. Our proposed model recorded setting performance in the context of Urdu POS tagging. Since in this study, we have evaluated the performance of CRF with both context word features and linguistic features, and the overall average results confirm that CRF with these features set outperform SVM with considerable margin, Therefore, from results, we conclude that:

- Feature-based classifier is the best choice for Urdu POS task.
- Secondly, SVM showed better results for POS tags whose frequency was less in training data while CRF showed better results for POS Tags whose frequency is higher. Therefore, from the results we conclude that SVM performs better when the training data is smaller while CFR performs better in case of larger training data.

In the second module we reported the development of Urdu NER system based on Deep Recurrent Neural Network (DRNN) learning algorithms with various model structures. Results show that the proposed DRNN based approach outperforms existing work that employs CRF based approaches.

From the results of the proposed models, we conclude that:

- The deep learning approach that makes use of word embeddings can be successfully applied to morphologically rich languages without performing any language-specific morphological analysis
- We showed that joint use of Word Embedding and POS features are useful for NER in Morphological rich languages
- Since Urdu is morphology rich language, therefore, we believe that it can easily be applied to other morphologically rich languages like Pashto, Punjabi, Sindhi, Arabic
- We used word embedding as a feature alongside both language-dependent and non-dependent features and hypothesise that this accounts for the improved performance over linear chain CRF.

The third module describes the development of our UNER Urdu named entity tagged dataset development. From the literature, we observed that though Urdu is the national language of Pakistan and also the most widely spoken language of the world lacking standard linguistic tools and resources. Therefore, we tried to contribute to Urdu language resource with a large enough newly created NE tagged dataset with the goal to make this dataset freely and widely acquirable, and to promote other researchers to exercise it as a criteria testbed for experimentations in Urdu NER research. Significant efforts were made to build this huge NE tagged dataset compared to existing NE dataset with text from multi news domains. The fascination aspect of the UNER dataset is its size as well as its very rich NE contents. These two aspects make UNER dataset more feasible for ML techniques. We hope that this new dataset will spark a light in ULP research community and will attract researchers in future to promote research in ULP.

The application areas which can be benefited from our proposed framework include information retrieval, machine translation, question answering, document summarization, location identification in GPS system, information extraction, speech recognition, text to speech, word sense disambiguation, semantic processing, parsing, conversation, voice search and plagiarism detection.

6.2 Future Work

In the current research work, we sustained provocative baselines for POS and NER -the two most basic NLP tasks, that can now be put-upon by other researchers for comparability and further enhancement. Our proposed framework is itself protractible and can easily conciliate further advanced NLP modules, bringing in it functional on genuine data.

In future, we are interested in extending our framework by adding more NLP tasks, such as automatic Urdu word segmentation, automatic parsing and automatic stemming modules with the larger objective to develop a sophisticated Urdu to English translation system.

So far as Urdu word segmentation is concerned, the IR systems performance are majorly dependent on word segmentation as most of the search engines are based on keyword search. From literature review we have observed that researchers from ULP research community showed a very little research interest. Very minimal research work has been reported. Also all the research work carried is based on rule-based approaches. Therefore, in the future, there is a large opportunity to investigate this task with ML perspectives.

Similarly, in future work, we plan (a) to create a billion-word Urdu POS corpus by crawling the Urdu web and annotating it automatically with Urdu POS using our CRF based model, and (b) to use the POS task to perform numerous NLP tasks including machine translation.

The existence of large enough labeled linguistics corpora is paramount necessary for carrying diverse NLP task. In this research work, we contributed to the ULP research community with the development of UNER dataset, though the size of UNER dataset is twice compared to the size of existing available datasets. But this is not significant when compared to linguistic resources of English and other Westernly languages. Therefore, in the future, we are also interested in extending the size of UNER dataset up to 200K words by including contents from other news domains such as entertainment, science, religion and much more.

References

Abbas, Q. (2014). *Semi-semantic part of speech annotation and evaluation.* Paper presented at the proceeding of the 8th Linguistic Annotation Workshop, Dublin, Ireland, August 23-24 2014.

Adeeba, F., & Hussain, S. (2011). *Experiences in building the Urdu WordNet.* Paper presented at the proceedings of the 9th Workshop on Asian Language Resources collocated with IJCNLP, Chiang Mai, Thailand

Al-Shammari, E. T. (2008). *Towards an Error-Free Stemming.* Paper presented at the IADIS European Conf. Data Mining.

Alotaibi, F., & Lee, M. G. (2014). *A Hybrid Approach to Features Representation for Fine-grained Arabic Named Entity Recognition.* Paper presented at the COLING.

Antony, P., & Soman, K. (2010). *Kernel based part of speech tagger for kannada.* Paper presented at the Machine Learning and Cybernetics (ICMLC), 2010 International Conference on.

Anwar, W., Wang, X., Li, L., & Wang, X.-l. (2007). *A statistical based part of speech tagger for Urdu language.* Paper presented at the International Conference on Machine Learning and Cybernetics, 2007

Atwell, E. (2008). Development of tag sets for part-of-speech tagging.

Benajiba, Y., & Rosso, P. (2008). *Arabic named entity recognition using conditional random fields.* Paper presented at the of Workshop on HLT & NLP within the Arabic World, LREC.

Biemann, C. (2006). *Unsupervised part-of-speech tagging employing efficient graph clustering.* Paper presented at the Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: student research workshop.

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997). *Nymble: a high-performance learning name-finder.* Paper presented at the the fifth conference on Applied natural language processing.

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). Description of the MENE Named Entity System as used in MUC-7.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics, 21*(4), 543-565.

Brodley, C. E., & Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of artificial intelligence research, 11*, 131-167.

Capstick, J., Diagne, A. K., Erbach, G., Uszkoreit, H., Leisenberg, A., & Leisenberg, M. (2000). A system for supporting cross-lingual information retrieval. *Information processing & management, 36*(2), 275-289.

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11*(Jul), 2079-2107.

Chiong, R., & Wei, W. (2006). *Named entity recognition using hybrid machine learning approach.* Paper presented at the 5th IEEE International Conference on Cognitive Informatics, 2006.

Collins, M., & Singer, Y. (1999). *Unsupervised models for named entity classification.* Paper presented at the the joint SIGDAT conference on empirical methods in natural language processing and very large corpora.

Dai, Z., Li, L., & Xu, W. (2016). CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases. *arXiv preprint arXiv:1606.01994*.

Daniel, J., & James, H. (2009). Speech and Language processing: An introduction to natural language processing. *Computational Linguistics and Speech Recognition, 2nd Ed., Prentice Hall*.

Das, A., Ganguly, D., & Garain, U. (2017). Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 16*(3), 18.

Daud, A., Khan, W., & Che, D. (2016). Urdu language processing: a survey. *Artificial Intelligence Review*, 1-33. doi: 10.1007/s10462-016-9482-x

Demir, H., & Ozgur, A. (2014). *Improving Named Entity Recognition for Morphologically Rich Languages Using Word Embeddings.* Paper presented at the ICMLA.

Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing, 3*.

Ekbal, A., Haque, R., & Bandyopadhyay, S. (2008). *Named Entity Recognition in Bengali: A Conditional Random Field Approach.* Paper presented at the the International Joint Conference on Natural Language Processing (IJCNLP).

Ekbal, A., Haque, R., Das, A., Poka, V., & Bandyopadhyay, S. (2008). *Language Independent Named Entity Recognition in Indian Languages.* Paper presented at the the International Joint Conference on Natural Language Processing (IJCNLP).

Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP), 8*(4), 14.

Florian, R., Henderson, J. C., & Ngai, G. (2000). *Coaxing confidences from an old friend: probabilistic classifications from transformation rule lists.* Paper presented at the Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13.

Gali, K., Surana, H., Vaidya, A., Shishtla, P., & Sharma, D. M. (2008). *Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition.* Paper presented at the IJCNLP.

Giménez, J., & Marquez, L. (2004). *SVMTool: A general POS tagger generator based on Support Vector Machines.* Paper presented at the In Proceedings of the 4th International Conference on Language Resources and Evaluation.

Goller, C., & Kuchler, A. (1996). *Learning task-dependent distributed representations by backpropagation through structure.* Paper presented at the Neural Networks, 1996., IEEE International Conference on.

Graça, J. V., Ganchev, K., Coheur, L., Pereira, F., & Taskar, B. (2011). Controlling complexity in part-of-speech induction. *Journal of artificial intelligence research, 41*, 527-551.

Hardie, A. (2003). *Developing a tagset for automated part-of-speech tagging in Urdu.* Paper presented at the the Corpus Linguistics 2003 conference. UCREL Technical Papers Volume 16. Department of Linguistics, Lancaster University, UK.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780.

Hoefel, G., & Elkan, C. (2008). *Learning a two-stage SVM/CRF sequence classifier.* Paper presented at the the 17th ACM conference on Information and knowledge management.

Hovy, D., & Spruit, S. L. (2016). *The social impact of natural language processing.* Paper presented at the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Hussain, S. (2004). *Letter to sound rules for Urdu text to speech system.* Paper presented at the Workshop on Computational Approaches to Arabic Scriptbased Languages, COLING.

Hussain, S. (2008). *Resources for Urdu Language Processing.* Paper presented at the IJCNLP.

Ijaz, M., & Hussain, S. (2007). *Corpus based Urdu lexicon development.* Paper presented at the the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan.

Jabbar, A., Iqbal, S., & Khan, M. U. G. (2016). *Analysis and Development of Resources for Urdu Text Stemming*. Paper presented at the LANGUAGE & TECHNOLOGY.

Jahangir, F., Anwar, W., Bajwa, U. I., & Wang, X. (2012). *N-gram and gazetteer list based named entity recognition for urdu: A scarce resourced language.* Paper presented at the 10th Workshop on Asian Language Resources.

Jawaid, B., & Ahmed, T. (2009). *Hindi to Urdu conversion: beyond simple transliteration.* Paper presented at the Conference on Language and Technology.

Jawaid, B., Kamran, A., & Bojar, O. (2014). *A Tagged Corpus and a Tagger for Urdu.* Paper presented at the LREC.

Jawaid, B., & Ondřej, B. (2012). *Tagger Voting for Urdu.* Paper presented at the 24th International Conference on Computational Linguistics.

Kazama, J. i., & Torisawa, K. (2007). *Exploiting Wikipedia as external knowledge for named entity recognition.* Paper presented at the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

Keretna, S., Lim, C. P., Creighton, D., & Shaban, K. B. (2015). Enhancing medical named entity recognition with an extended segment representation technique. *Computer methods and programs in biomedicine, 119*(2), 88-100.

Khan, W., Daud, A., Nasir, J. A., & Amjad, T. (2016). A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait journal of Science, 43*(4), 66-84.

Kumar, P., & Kiran, V. R. (2008). A Hybrid Named Entity Recognition System for South Asian Languages. *NER for South and South East Asian Languages*, 83.

Kumar Saha, S., Sarathi Ghosh, P., Sarkar, S., & Mitra, P. (2008). Named entity recognition in Hindi using maximum entropy and transliteration. *Polibits*(38), 33-41.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. Paper presented at the eighteenth international conference on machine learning, ICML.

Lai, D. W., & Surood, S. (2013). Effect of service barriers on health status of aging South Asian immigrants in Calgary, Canada. *Health & social work, 38*(1), 41-50.

Lu, Z., Li, L., & Xu, W. (2015). *Twisted Recurrent Network for Named Entity Recognition.* Paper presented at the Bay Area Machine Learning Symposium.

Malik, M. K. (2017). Urdu Named Entity Recognition and Classification System Using Artificial Neural Network. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 17*(1), 2.

Malik, M. K., & Sarwar, S. M. (2015). urdu named entity recognition and classification system using conditional random field. *Science International 5*(27), 4473-4477.

McCallum, A., & Li, W. (2003). *Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.* Paper presented at the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4.

Muaz, A., Ali, A., & Hussain, S. (2009). *Analysis and development of Urdu POS tagged corpus.* Paper presented at the Proceedings of the 7th Workshop on Asian Language Resources.

Mukund, S. (2012). *An NLP Framework for Non-Topical Text Analysis in Urdu--A Resource Poor Language*: ERIC.

Mukund, S., Srihari, R., & Peterson, E. (2010). An Information-Extraction System for Urdu---A Resource-Poor Language. *ACM Transactions on Asian Language Information Processing (TALIP), 9*(4), 1-43.

Mukund, S., & Srihari, R. K. (2009). *NE tagging for Urdu based on bootstrap POS learning.* Paper presented at the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies.

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

Naz, F., Anwar, W., Bajwa, U. I., & Munir, E. U. (2012). Urdu part of speech tagging using transformation based error driven learning. *World Applied Sciences Journal, 16*(3), 437-448.

Naz, S., Umar, A. I., Shirazi, S. H., Khan, S. A., Ahmed, I., & Khan, A. A. (2014). Challenges of Urdu Named Entity Recognition: A Scarce Resourced Language. *Research Journal of Applied Sciences, Engineering and Technology, 8*(10), 1272-1278.

Oudah, M., & Shaalan, K. (2016). NERA 2.0: Improving coverage and performance of rule-based named entity recognition for Arabic. *Natural Language Engineering*, 1-32. doi: 10.1017/S1351324916000097

Oudah, M., & Shaalan, K. F. (2012). *A Pipeline Arabic Named Entity Recognition using a Hybrid Approach.* Paper presented at the COLING.

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Platts, J. T. (1874). *A grammar of the Hindustani or Urdu language*: WH Allen.

Poibeau, T., & Kosseim, L. (2001). Proper name extraction from non-journalistic texts. *Language and computers, 37*(1), 144-157.

Rahman, T. (1997). The medium of instruction controversy in Pakistan. *Journal of Multilingual and Multicultural Development, 18*(2), 145-154.

Rau, L. F. (1991). *Extracting company names from text.* Paper presented at the Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on.

Raymond, C., & Riccardi, G. (2007). *Generative and discriminative algorithms for spoken language understanding.* Paper presented at the INTERSPEECH.

Raza, A., & Hussain, S. (2010). *Automatic diacritization for urdu.* Paper presented at the Proceedings of the Conference on Language and Technology.

Riaz, K. (2008). *Baseline for Urdu IR evaluation.* Paper presented at the Proceedings of the 2nd ACM workshop on Improving non english web searching.

Riaz, K. (2010). *Rule-based named entity recognition in Urdu.* Paper presented at the In the 2010 Association for Computational Linguistics

Named Entities Workshop, Association for Computational Linguistics.

Roberts, A., Gaizauskas, R. J., Hepple, M., & Guo, Y. (2008). *Combining Terminology Resources and Statistical Methods for Entity Recognition: an Evaluation.* Paper presented at the the Conference on Language Resources and Evaluation (LRE'08), .

Roth, D., & Zelenko, D. (1998). *Part of speech tagging using a network of linear separators.* Paper presented at the Proceedings of the 17th international conference on Computational linguistics-Volume 2.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation: California Univ San Diego La Jolla Inst for Cognitive Science.

Sajjad, H., & Schmid, H. (2009). *Tagging Urdu text with parts of speech: A tagger comparison.* Paper presented at the Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.

Sak, H., Senior, A., & Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling.* Paper presented at the Fifteenth annual conference of the international speech communication association.

Sassano, M. (2014). *Deterministic Word Segmentation Using Maximum Matching with Fully Lexicalized Rules.* Paper presented at the the European Association for Computational Linguistics (EACL).

Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Learning, 4*(2).

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85-117.

Schmidt, R. L. (1999). *Urdu, an Essential Grammar*: Psychology Press.

Seok, M., Song, H.-J., Park, C.-Y., Kim, J.-D., & Kim, Y.-s. (2016). Named Entity Recognition using Word Embedding as a Feature. *International Journal of Software Engineering and Its Applications, 10*(2), 93-104.

Shaalan, K. (2014). A survey of arabic named entity recognition and classification. *Computational Linguistics, 40*(2), 469-510.

Shaalan, K., & Raza, H. (2009). NERA: Named entity recognition for Arabic. *Journal of the American Society for Information Science and Technology, 60*(8), 1652-1663.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., . . . Lanctot, M. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489.

Singh, U., Goyal, V., & Lehal, G. S. (2012). *Named Entity Recognition System for Urdu.* Paper presented at the COLING.

Song, S., Zhang, N., & Huang, H. (2017). Named entity recognition based on conditional random fields. *Cluster Computing*, 1-12.

Sundheim, B. M. (1996). *Overview of results of the MUC-6 evaluation.* Paper presented at the workshop on held at Vienna, Virginia:, 1996.

Tafseer, A., Urooj, S., Hussain, S., Mustafa, A., Parveen, R., Adeeba, F., . . . Butt, M. (2015). *The CLE Urdu POS Tagset.* Paper presented at the LREC 2014, Ninth International Conference on Language Resources and Evaluation.

Thenmalar, S., Balaji, J., & Geetha, T. (2015). Semi-supervised Bootstrapping approach for Named Entity Recognition. *arXiv preprint arXiv:1511.06833*.

Tjong Kim Sang, E. F., & De Meulder, F. (2003). *Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition.* Paper presented at the the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4.

Williams, D., & Hinton, G. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-538.

Wong, D. F., Chao, L. S., & Zeng, X. (2014). iSentenizer-: Multilingual sentence boundary detection model. *The Scientific World Journal, 2014*(1), 1-10.

Wu, Y.-C. (2014). A top-down information theoretic word clustering algorithm for phrase recognition. *Information Sciences, 275*, 213-225.

Yao, K., Peng, B., Zweig, G., Yu, D., Li, X., & Gao, F. (2014). *Recurrent conditional random field for language understanding*. Paper presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), .

Zhang, Z., Sun, Z., Liu, J., Chen, J., Huo, Z., & Zhang, X. (2016). Deep Recurrent Convolutional Neural Network: Improving Performance For Speech Recognition. doi: rXiv:1611.07174v2 [cs.CL] 27 Dec 2016

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS), 23*(4), 550-560.

Žitnik, S., Šubelj, L., & Bajec, M. (2014). SkipCor: Skip-mention coreference resolution using linear-chain conditional random fields. *PloS one, 9*(6), 1-14.

Appendix

Table 6-1:Tag wise Confusion matrix of SVM model on BJ dataset

	PN	NN	ADJ	VB	EXP	AA	ADV	CA	PM	TA	SC	I	Q	G	INT
PN	7089	4617	262	174	132	52	26	25	24	9	7	5	4	2	1
	G	ADJ	NN	---	---	---	---	---	---	---	---	---	---	---	---
G	949	5	1	---	---	---	---	---	---	---	---	---	---	---	---
	NN	PN	ADJ	VB	AA	EXP	CA	ADV	Q	INT	G	I	TA	PP	SC
NN	40943	1653	444	309	78	49	30	24	17	10	8	6	5	8	4
	P	CC	NN	---	---	---	---	---	---	---	---	---	---	---	---
P	20674	3	1	---	---	---	---	---	---	---	---	---	---	---	---
	U	NN	---	---	---	---	---	---	---	---	---	---	---	---	---
U	74	6	---	---	---	---	---	---	---	---	---	---	---	---	---
	VB	TA	NN	AA	P	PN	ADJ	ADV	Q	AKP	CA	---	---	---	---
VB	13639	3405	1743	981	310	201	88	23	7	2	3	---	---	---	---
	SM	---	---	---	---	---	---	---	---	---	---	---	---	---	---
SM	1082	---	---	---	---	---	---	---	---	---	---	---	---	---	---
	PM	NN	PN	EXP	CA	PD	---	---	---	---	---	---	---	---	---
PM	2853	1097	64	11	3	1		---	---	---	---	---	---	---	---
	PP	PD	P	ADJ	NN	Q	OR	---	---	---	---	---	---	---	---
PP	5417	1040	358	19	8	2	1	---	---	---	---	---	---	---	---
	CC	---	---	---	---	---	---	---	---	---	---	---	---	---	---
CC	3832	---	---	---	---	---	---	---	---	---	---	---	---	---	---
	ADJ	NN	PN	VB	PP	Q	EXP	ADV	CA	G	AA	I	---	---	---
ADJ	7968	1974	619	59	38	34	21	13	6	5	3	2	---	---	---
	CA	NN	PN	VB	EXP	ADJ	ADV	---	---	---	---	---	---	---	---
CA	3171	365	166	58	20	4	6	---	---	---	---	---	---	---	---
	RP	NN	---	---	---	---	---	---	---	---	---	---	---	---	---
RP	186	6	---	---	---	---	---	---	---	---	---	---	---	---	---
	SC	ADV	PN	NN	VB	---	---	---	---	---	---	---	---	---	---
SC	4754	52	30	25	11	---	---	---	---	---	---	---	---	---	---
	SE	---	---	---	---	---	---	---	---	---	---	---	---	---	---
SE	2932	---	---	---	---	---	---	---	---	---	---	---	---	---	---
	ADV	NN	ADJ	VB	AA	TA	PN	CC	SC	---	---	---	---	---	---
ADV	2620	187	58	28	16	11	9	16	1	---	---	---	---	---	---
	EXP	PN	NN	CA	---	---	---	---	---	---	---	---	---	---	---
EXP	193	171	5	2	---	---	---	---	---	---	---	---	---	---	---
	I	SC	NN	PP	TA	PN	---	---	---	---	---	---	---	---	---
I	2570	797	110	14	4	3	---	---	---	---	---	---	---	---	---

NEG

Table 6-2: Tag wise Confusion matrix of CRF model on BJ dataset

	PN	NN	SM	ADJ	VB	EXP	CA	PM	TA	SC	ADV	RP	I	G	U	PP	SE	QW	P	NEG	Q
PN	7802	1784	404	294	192	95	104	35	24	21	20	15	13	11	16	12	30	4	9	2	2
	G	NN	ADJ	PN	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
G	734	16	8	6	2	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	NN	SM	PN	VB	ADJ	AA	I	CA	EXP	PM	ADV	G	PD	PP	RP	REP	U	P	-	--	-
NN	38950	831	598	285	197	55	30	29	27	24	20	30	7	15	4	12	10	6	-	--	-
	P	PP	VB	NN	CC	ADJ	ADJ	--	--	--	--	--	--	--	--	--	--	--	-	--	-
P	20513	18	11	3	2	1	1	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	U	NN	PN	VB	SM	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
U	48	17	5	6	2	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	VB	NN	SM	PN	AA	TA	ADJ	QW	P	KER	CA	ADV	SC	PM	U	G	--	--	-	--	-
VB	18210	804	232	186	184	169	137	59	51	44	17	9	8	3	4	6	--	--	-	--	-
	SM	NN	PN	ADJ	VB	AA	CA	PP	ADV	TA	SC	--	--	--	--	--	--	--	-	--	-
SM	4723	1379	509	127	93	33	23	4	3	2	1	--	--	--	--	--	--	--	-	--	-
	PM	CA	NN	PN	SM	ADJ	EXP	VB	AA	PP	I	P	--	--	--	--	--	--	-	--	-
PM	3325	181	155	139	39	30	13	10	5	4	6	5	--	--	--	--	--	--	-	--	-
	PP	PD	P	NN	ADJ	PN	VB	Q	AA	--	--	--	--	--	--	--	--	--	-	--	-
PP	5423	124	40	14	9	7	5	2	1	--	--	--	--	--	--	--	--	--	-	--	-
	CC	PN	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
CC	3555	2	2	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	ADJ	NN	PN	SM	VB	CA	Q	PP	EXP	PM	ADV	SC	TA	GR	RP	G	CC		-		-
ADJ	8297	810	376	148	131	40	18	17	14	13	9	8	5	8	6	12	2		-		-
	CA	NN	PN	ADJ	SM	VB	EXP	AA	PM	Q	PP	SC	--	--	--	--	--	--	-	--	-
CA	2941	226	141	48	38	22	18	8	5	3	4	6	--	--	--	--	--	--	-	--	-
	RP	PN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
RP	174	6	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	SC	I	NN	ADV	PN	ADJ	VB	TA	--	--	--	--	--	--	--	--	--	--	-	--	-

SC	4170	41	27	22	16	5	4	2	--	--	--	--	--	--	--	--	--	--	-	--	-
	SE	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
SE	2930	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	ADV	NN	ADJ	PN	SM	VB	AA	TA	CC	SC	U	--	--	--	--	--	--	--	-	--	-
ADV	2326	106	59	42	20	18	10	7	12	3	4	--	--	--	--	--	--	--	-	--	-
	EXP	PN	NN	ADJ	CA	PM	VB	SM	--	--	--	--	--	--	--	--	--	--	-	--	-
EXP	233	71	31	5	4	3	2	2	--	--	--	--	--	--	--	--	--	--	-	--	-
	I	SC	NN	PP	PN	ADJ	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
I	3271	140	39	9	6	4	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	NEG	NN	PN	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
NEG	2006	8	6	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	TA	VB	NN	AA	PN	SM	PM	--	--	--	--	--	--	--	--	--	--	--	-	--	-
TA	5707	654	17	8	5	4	3	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	AP	NN	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
AP	1080	4	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	Q	ADV	ADJ	NN	PN	VB	G	--	--	--	--	--	--	--	--	--	--	--	-	--	-
Q	2031	56	27	26	8	5	3	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	PD	PP	PN	I	SM	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
PD	1740	55	9	2	1	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	WALA	PN	NN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
WALA	500	7	6	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	KP	QW	VB	NN	PN	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
KP	149	27	13	4	3	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	GR	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
GR	743	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	REP	NN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
REP	1091	3	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	A	NN	PN	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
A	363	6	3	3	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-

	KD	PN	NN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
KD	414	3	3	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	AA	VB	NN	PN	SM	ADJ	CA	TA	ADV	QW	SC	PM	--	--	--	--	--	--	-	--	-
AA	4407	367	162	76	38	19	12	9	6	3	2	8	--	--	--	--	--	--	-	--	-
	QW	VB	KP	NN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
QW	341	14	2	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	KER	VB	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
KER	441	50	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	OR	ADJ	NN	PN	VB	SM	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
OR	231	15	12	11	3	3	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	AKP	NN	VB	PN	ADJ	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
AKP	435	11	5	4	4	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	INT	NN	ADJ	PN	RP	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
INT	76	7	5	4	1	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	AD	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
AD	88	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
	NN	FR	ADJ	PN	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-
FR	12	10	7	3	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-	--	-

Table 6-3: Tag wise confusion matrix of CRF model on CLE dataset

	NN	VBF	JJ	NNP	VBI	AUXA	PRP	PSP	PDM	PRR	Q	SCP	SC	CC
NN	28659	176	137	63	32	22	22	18	7	12	15	3	12	5
	NNP	NN	JJ	VBF	AUXA	PRP	RB	VBI	AUXT	PRT	PDM	OD	CC	PSP
NNP	3276	1192	124	52	13	11	9	16	7	6	10	3	8	8
	PSP	PRP	PRT	VBF	SCK	NN	NNP	CC	AUXA	--	--	--	--	--
PSP	17172	28	15	11	10	9	6	2	3	--	--	--	--	--
	VBI	NN	NNP	VBF	JJ	AUXA	PSP	--	--	--	--	--	--	--
VBI	1338	211	41	35	25	9	8	--	--	--	--	--	--	--
	VBF	NN	AUXT	AUXA	JJ	PSP	AUXP	VBI	NNP	AUXM	SCP	SC	PRT	--
VBF	10292	287	157	102	88	21	17	11	27	7	4	8	5	--
	CC	SCP	NNP	PSP	NN	SC	--	--	--	--	--	--	--	--
CC	2944	250	20	6	4	1	--	--	--	--	--	--	--	--
	JJ	NN	VBF	NNP	CC	AUXA	CD	RB	PDM	AUXT	PRP	VBI	PSP	PU

JJ	7117	703	112	56	20	15	16	7	6	10	4	12	4	8
	PRP	PDM	PSP	NN	NNP	VBF	SC	AUXA	--	--	--	--	--	--
PRP	4153	162	58	47	16	6	2	4	--	--	--	--	--	--
	AUXA	VBF	VBI	NN	AUXT	JJ	NNP	AUXM	PSP	--	--	--	--	--
AUXA	2760	190	25	13	10	6	4	3	4	--	--	--	--	--
	PU	PSP	--	--	--	--	--	--	--	--	--	--	--	--
PU	10365	4	--	--	--	--	--	--	--	--	--	--	--	--
	SC	SCP	RB	NN	NNP	JJ	PRT	CC	PSP	--	--	--	--	--
SC	1732	96	17	10	6	4	6	2	4	--	--	--	--	--
	RB	NN	JJ	VBF	NNP	SCP	SC	VBI	PRR	--	--	--	--	--
RB	1173	143	42	18	16	13	10	6	8	--	--	--	--	--
	PDM	PRP	JJ	NNP	NN	CD	PRR	VBF		--	--	--	--	--
PDM	1992	98	13	6	4	3	4	2		--	--	--	--	--
	CD	NN	JJ	NNP	NNP	AUXA	FF	VBI	PSP	PU				
CD	2097	96	46	16	16	7	4	3	6	4				
	PRT	SC	VBF	NN	--	--	--	--	--	--	--	--	--	--
PRT	1457	28	3	4	--	--	--	--	--	--	--	--	--	--
	APNA	PRF	--	--	--	--	--	--	--	--	--	--	--	--
APNA	580	1	--	--	--	--	--	--	--	--	--	--	--	--
	PRR	NN	NNP	JJ	VBF		--	--	--	--	--	--	--	--
PRR	685	10	6	4	6		--	--	--	--	--	--	--	--
	AUXT	VBF	CC	--	--	--	--	--	--	--	--	--	--	--
AUXT	3408	69	3	--	--	--	--	--	--	--	--	--	--	--
	PRD	PRR	PDM	PRP	--	--	--	--	--	--	--	--	--	--
PRD	102	34	2	1	--	--	--	--	--	--	--	--	--	--
	Q	NN	JJ	NNP	AUXA	--	--	--	--	--	--	--	--	--
Q	1320	37	16	4	4	--	--	--	--	--	--	--	--	--
	AUXM	VBF	AUXA	AUXT	NN	JJ	--	--	--	--	--	--	--	--
AUXM	451	32	6	5	4	1	--	--	--	--	--	--	--	--
	SYM	NN	NNP	JJ	--	--	--	--	--	--	--	--	--	--
SYM	253	14	4	3	--	--	--	--	--	--	--	--	--	--
	NEG	NN	JJ	NNP	--	--	--	--	--	--	--	--	--	--
NEG	952	6	4	4	--	--	--	--	--	--	--	--	--	--
	AUXP	VBF	NN	AUXA	--	--	--	--	--	--	--	--	--	--
AUXP	478	19	4	2	--	--	--	--	--	--	--	--	--	--
	SCK	VBF	PSP	--	--	--	--	--	--	--	--	--	--	--
SCK	651	15	8	--	--	--	--	--	--	--	--	--	--	--
	OD	NN	JJ	NNP	VBF	PRP	AUXA	--	--	--	--	--	--	--
OD	407	51	25	6	8	4	1	--	--	--	--	--	--	--
	CD	FF	NN	NNP	JJ	PDM	VBF	VBI	PSP	PRP	--	--	--	--
FF	69	57	32	31	13	6	12	6	4	4	--	--	--	--

	PRS	JJ	NN	NNP	VBI		--	--	--	--	--	--	--	--
PRS	483	7	5	2	4	--	--	--	--	--	--	--	--	--
	PRF	PRP	NN	NNP	--	--	--	--	--	--	--	--	--	--
PRF	80	4	3	4	--	--	--	--	--	--	--	--	--	--
	VALA	NN	--	--	--	--	--	--	--	--	--	--	--	--
VALA	293	2	--	--	--	--	--	--	--	--	--	--	--	--
	SCP	NNP	NN	RB	--	--	--	--	--	--	--	--	--	--
SCP	296	3	2	1	--	--	--	--	--	--	--	--	--	--
	FR	JJ	NN	NNP	CD	RB	--	--	--	--	--	--	--	--
FR	41	10	8	5	4	1	--	--	--	--	--	--	--	--
	NN	AUXA	NNP	--	--	--	--	--	--	--	--	--	--	--
PRE	9	2	2	--	--	--	--	--	--	--	--	--	--	--
	NN	INJ	NNP	JJ	CC	AUXA	VBF	RB	PU	--	--	--	--	--
INJ	48	31	16	11	5	4	9	2	2	--	--	--	--	--
	QM	NNP	NN	--	--	--	--	--	--	--	--	--	--	--
QM	5	4	1	--	--	--	--	--	--	--	--	--	--	--