An Urdu Semantic Tagger – Lexicons, Corpora, Methods, and Tools

Jawad Shafi

Supervisors: Dr. Paul Rayson (Lancaster University)

Dr. Rao Muhammad Adeel Nawab (COMSATS)

School of Computing and Communications InfoLab21

Lancaster University

This dissertation is submitted for the degree of

Doctor of Philosophy

January 2020

I would like to dedicate this thesis to my loving *parents*, and *wife*.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Jawad Shafi

January 2020

Acknowledgements

First and foremost I would like to thanks to Allah SWT who is the source of all the knowledge in this world, and imparts as much as He wishes to any one He finds suitable. My deepest gratitude and acknowledgement goes to my supervisors, Dr. Rao Muhammad Adeel Nawab and Dr. Paul Rayson, who have supported me throughout my work on this dissertation with their patience and knowledge. Without their guidance, expertise, motivation and support this Ph.D. dissertation would not have been completed otherwise. One could not wish for more kind, accessible and friendlier supervisors than them.

My special thanks to Mr. Muhammad Sharjeel, Mr. Hafiz Muhammad Rizwan Iqbal, and Dr. Scott Piao who helped me a lot throughout my work and gave very useful suggestions regarding it. I offer my sincere thanks to my elders (Hazrat Muhammad Abdul Qadir Owaise Sb DB, Hazret Abdul Jabbar Khan Sb DB, Mufti Shahid Muneeb sb DB, Sh. Muhammad Naseem Sb) for their special attention and prayers. Without their support, motivation and guidance it would not have been possible to successfully complete this Ph.D. thesis. I am thankful to my parents for their support, prayers, love and care throughout my life and they have played a vital role in achieving this milestone, indeed. My wife has always been a wonderful being to me and extended her whole-hearted support especially during my Ph.D. studies, which I could not have been completed without her. To my daughters (Mahrosh, Mashaim) and son (Muhammad Ibrahim); all of you have played your part very well as your naughtiness always made me relaxed. Acknowledgement goes to my sisters (Samina Faisal, and Amina Anees: have cheered me up in difficult moments, celebrated with me for my achievements, and never blamed me for being often far away from them), brothers, in-laws, grand-mother, relatives, nephews (Subhan Faisal) as well as nieces, my students for their continuous support and prayers. Also many thanks to my friends and colleagues; Dr. Touseef Tahir, Dr. S.A. Abid, Dr. Abdul Waheed, Umer Farooq, Mohsin Hafeez, Mansoor Siddique, Goher Ayoub, Samiullah, Tayyab, Tanzeel, Bilal, Umer Sheikh, Maj. Sheraz Ikram, Taimoor, Saqib, Zaheer, Tahir-ul-islam, Haider and Dr. Sarfraz Iqbal. Last but not least my special thanks and gratitude to all those who helped me to complete Ph.D.

I am thankful to COMSATS University Islamabad, Pakistan for funding this Ph.D. under the Split Site Ph.D. Program and to Lancaster University, U.K. for their tremendous resources and help.

Finally, my Ph.D. was one of the tough but best experiences of my life as it gave me the possibility to: research, teach, spend balanced life, travel, work with scientists from top research institutes, get in touch with both far West and far East cultures, and meet special people who will always be part of my life.

Jawad Shafi; January 2020

Abstract

Extracting and analysing meaning-related information from natural language data has attracted the attention of researchers in various fields, such as Natural Language Processing (NLP), corpus linguistics, data sciences, etc. An important aspect of such automatic information extraction and analysis is the semantic annotation of language data using semantic annotation tool (a.k.a semantic tagger). Generally, different semantic annotation tools have been designed to carry out various levels of semantic annotations, for instance, sentiment analysis, word sense disambiguation, content analysis, semantic role labelling, etc. These semantic annotation tools identify or tag partial core semantic information of language data, moreover, they tend to be applicable only for English and other European languages. A semantic annotation tool that can annotate semantic senses of all lexical units (words) is still desirable for the Urdu language based on USAS (the UCREL Semantic Analysis System) semantic taxonomy, in order to provide comprehensive semantic analysis of Urdu language text. This research work report on the development of an Urdu semantic tagging tool and discuss challenging issues which have been faced in this Ph.D. research work. Since standard NLP pipeline tools are not widely available for Urdu, alongside the Urdu semantic tagger a suite of newly developed tools have been created: sentence tokenizer, word tokenizer and part-of-speech tagger. Results for these proposed tools are as follows: word tokenizer reports F_1 of 94.01%, and accuracy of 97.21%, sentence tokenizer shows F_1 of 92.59%, and accuracy of 93.15%, whereas, POS tagger shows an accuracy of 95.14%. The Urdu semantic tagger incorporates semantic resources (lexicon and corpora) as well as semantic field disambiguation methods. In terms of novelty, the NLP pre-processing tools are developed either using rule-based, statistical, or hybrid techniques. Furthermore, all semantic lexicons have been developed using a novel combination of automatic or semi-automatic approaches: mapping, crowdsourcing, statistical machine translation, GIZA++, word embeddings, and named entity. A large multi-target annotated corpus is also constructed using a semi-automatic approach to test accuracy of the Urdu semantic tagger, proposed corpus is also used to train and test supervised multi-target Machine Learning classifiers. The results show that Random k-labEL Disjoint Pruned Sets and Classifier Chain multi-target classifiers outperform all other classifiers on the proposed corpus with a Hamming Loss of 0.06% and Accuracy of 0.94%. The best lexical coverage of 88.59%, 99.63%, 96.71% and 89.63% are obtained on several test corpora. The developed Urdu semantic tagger shows encouraging precision on the proposed test corpus of 79.47%.

Despite good results of the proposed tools, methods, lexicons and corpora, however, the following limitations have been observed. A word tokenization method did not handle out-of-vocabulary words in morpheme matching process of space omission problem. Sentence tokenization is rule based and are not able to dealt with non-sentence boundary markers and period marker used between different abbreviations. Whereas, the POS tagger did not completely handle unknown words. Multi-target classifiers did not explore feature extraction approaches and has only been tested on a small dataset. Finally, future work will need to focus on the creation of multi-word semantic lexicons.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ACAMRIT	Automatic Content Analysis of Market Research Interview Transcripts
ACASD	Automatic Content Analysis of Spoken Discourse
AI	Artificial Intelligence
AIj	Ali Ijaz
ANN	Artificial Neural Network
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASSIST	Automatic Semantic Assistance for Translators
BBC	The British Broadcasting Corporation
BilBOWA	Bilingual Bag-of-Words without Word Alignments
BNC	British National Corpus
BoW	Bag of Words
BR	Binary Relevance

CC	Creative Commons
CFIE	Corporate Financial Information Environment
CLAWS	the Constituent Likelihood Automatic Word-tagging System
CL	Computational Linguistics
CLE	Center for Language Engineering
CLex	Computational Lexicon
CM	Crude Morphemes
COUNTER	COrpus of Urdu News TExt Reuse
CRF	Conditional Random Field
DeReKo	Deutschen Referenz Korpus– The Mannheim German Reference Corpus
DMM	Dynamic Maximum Matching
EAGLES	Expert Advisory Group on Language Engineering Standards
EM	Exact Match
EMILLE	Enabling Minority Language Engineering
En-Ur	English-Urdu
ESL	English Semantic Lexicon
EST	English Semantic Tagger
EUSAP	English-Urdu Sentence Aligned Parallel Corpus
FST	Finish Semantic Tagger

GATE	General Architecture for Text Engineering
GUI	Graphical User Interface
GUSAI	Graphical User Semantic Annotation Interface
HC	Hans Christensen
HL	Hamming Loss
HLT	Human Language Technology
HMM	Hidden Markov Model
HTML	Hypertext Markup Language
HTST	the Historical Thesaurus Semantic Tagger
ICT	Information and Communications Technology
LC	Label Combination
LDOCE	Longman Dictionary of Contemporary English
LLOCE	Longman Lexicon of Contemporary English
MaEn	Maximum Entropy
MD	Muaz Dataset
METER	MEasuring TExt Reuse
MFS	Most Frequent Sense
MIT	the Massachusetts Institute of Technology
MLE	Maximum Likelihood Estimation

ML	Machine Learning
MWE	Multi Word Expression
NB	Naïve Bayes
NER	Named entity Recognition
NLP	Natural Language Processing
NLTK	the Natural Language Toolkit
NSBM	Non Sentence Boundary Markers
POS	Part-Of-Speech
PT	Problem Transformation
RAkELd	Random k-labEL Disjoint Pruned Sets
REVERE	Requirements Reverse Engineering to Support Business Process Change
RF	Random Forest
RST	Russian Semantic Tagger
SAI	Semantic Annotation Interface
SBM	Sentence Boundary Marker
ST	Sentence Tokenizer
SVM	Support Vector Machine
TnT	Trigrams-and-Tag
TT	Tree Tagger

txt	Text
UAW-WSD-18	Urdu All Word WSD
UCM	Ultra Crude Morphemes
UCREL	the University Centre for Computer Corpus Research on Language
UKTB	URDU.KON-TB treebank
UK	United Kingdom
ULS-WSD-18	Urdu Lexical Sample WSD
UMC	Urdu Monolingual Corpus
UMLi	Urdu Mono-Lingual
UNER	Urdu Named Entity Recognition
UNLTools	Urdu Natural Language Tools
UNLT	Urdu Natural Language Toolkit
UPPC	Urdu Paraphrase Plagiarism Corpus
URL	Uniform Resource Locator
US Tagger	The Urdu Semantic Tagger
USAS	the UCREL Semantic Analysis System
USC	Urdu Summary Corpus
UTF	Unicode Transformation Format
WSD	Word Sense Disambiguation

WT	Word Tokenizer
XML	Extensible Markup Language
ZWNJ	Zero Width Non Joiner

Chapter 1

Introduction

This Ph.D. thesis describes the theory, motivation, development, and evaluation of semantic tagging resources developed for the Urdu language. These resources provide a framework required for the Urdu semantic tagger pipeline; in other words, they are natural language processing tools, lexicons, corpora and methods which are used by a computer to perform semantic tagging. This thesis describes and evaluates these resources, outline their further development, and suggest applications for them. The thesis places this work in the context of a new Urdu semantic tagger for the development of various types of natural language processing and human language technology applications involving the Urdu language.

1.1 Context and Motivation

Semantic tagging is a dictionary-based process of identifying and labelling the meaning of natural language text. Semantic tagging is useful for the fine-grained analysis of words, therefore, a relevant task for several research areas and practical applications, for instance Natural Language Processing (NLP), Human Language Technology (HLT), data science, machine translation, information retrieval, corpus linguistics, sentic computing, bi-lingual/multi-lingual extraction of multi-words, mono-lingual/cross-lingual information extraction, classification of language, and so on. In recent research, different types of semantic tagging tools (or semantic taggers) have been suggested and developed to carry out various levels of semantic analysis.

Some types of semantic tagging tools have been designed to identify topics of a given text [15]. Others are used to extract specific or partial information, for example, types of named entities or events [198, 241] or a common sense based framework for concept level opinion mining/ sentiment analysis [44]. Another type of semantic tagging tool is designed to identify semantic categories for all lexical units (words and multi-word expressions) using a predefined semantic taxonomy. In order to support semantic information extraction and analysis from language data, the latter types of tools require richer semantic lexical resources and provide a broader level of sense disambiguation, and thus, are challenging to create. In this research work, main focus will be on developing the benchmark NLP pre-processing tools, dictionaries, corpora and methods for a semantically rich text analytical tool.

Several semantically rich lexical resources and annotation tools are available for monolingual analysis, particularly for English e.g. WordNet [134, 149], but very few resources or tools exist that can be used to carry out semantic analysis for multilingual text, such as, EuroWordNet [237], BabelNet [149], and USAS[1] [180], which have many applications in the development of intelligent NLP and HLT systems. For example, the original English USAS semantic annotation tool (or English semantic tagger) has been applied in numerous research studies such as entrepreneurship [65], software engineering [227], empirical language analysis [171], requirements engineering [182], historical semantic analysis via HTST 1.1 [166], to train a Chatbot [218], and several others [23, 214]. Moreover, USAS [180] has been ported previously

[1] USAS: the UCREL (University Centre for Computer Corpus Research on Language) Semantic Analysis System, HTST: the Historical Thesaurus Semantic Tagger

to cover many other languages[2] (Arabic, Finnish, Russian, Chinese, Welsh, Italian, Portuguese, Czech, Dutch and Spanish) with a unified semantic annotation scheme. Following this established framework i.e. *USAS* [180] therefore, in this research work primarily focus will be the development of a coarse-grained *all-words* semantic tagging tool rather than annotating fine-grained word senses as in WordNet.

Originally developed for the English semantic tagging task, USAS [180] is a commonly used semantic field oriented analysis system. Compared to Word Sense Disambiguation (WSD) systems, it does not disambiguate between fine-grained word sense definitions, but rather, it assigns a semantic category (or categories) to each word or phrase by employing a unified semantic annotation taxonomy. USAS is also different from those systems which extract other types of information (named entity recognition, semantic role labelling, etc), in that it assigns semantic field tag(s) to every lexical unit in a running text. The required resources and methods in the development and evaluation of the USAS [180] system are: (i) a set of semantic field tags[3] (see Table 2.4, for major semantic field tags), (ii) single and multi-word semantic lexicons, (iii) semantic field disambiguation methods, and (iv) a software framework (for more details on these see Section 2.4.1).

With the web transforming into a multi-lingual hub, the NLP research community has also diverted its focus to the development of multi-lingual tools. As a consequence, USAS [180] has been ported for various languages (mentioned previously) based on semantic lexicons using a unified semantic annotation scheme. However, the focus is primarily towards Western and East Asian languages. Unfortunately, much less effort has been devoted to South-Asian languages particularly Urdu, and

[2] http://ucrel.lancaster.ac.uk/usas/ - Last visited: 9-October-2018

[3] The USAS semantic fields are originally based on the Longman Lexicon of Contemporary English taxonomy, with 21 major semantic fields which expand into 232 sub-fields: http://ucrel.lancaster.ac.uk/usas/USASSemanticTagset.pdf - Last visited: 29-October-2018

there is a dearth of semantic resources and annotation tools for Urdu, which is a common and widely spoken language of the world [216].

1.2 Importance and Characteristics of the Urdu Language

Urdu is one of the most popular languages spoken around the globe and an official language of Pakistan. There is a dire need to develop basic NLP text annotation and analysis resources for this highly under-resourced language for several reasons; (i) it has 400 million speakers around the world [41, 1], (ii) digital text is readily available through on-line repositories and is rapidly increasing day by day [4], (iii) it has ethnic and geographically diverse speakers, (iv) a wide South-Asian diaspora [41], (v) it is a *lingua franca* for the South-Asian business community in Pakistan and in the South Asian community in the U.K [194], and (vi) one of the widely spoken language in the United Kingdom [22].

Urdu is an Indo-Aryan[4] (or Indic) language derived from Sanskrit/Hindustani language [33], has been heavily influenced by Arabic, Persian [33] and less by Turkic (Chagatai[5]) languages for literary and technical vocabulary [216], and is written from right to left in Nastaliq style [59, 216]. Urdu is a highly inflectional and morphologically rich language [202], including many multi-word expressions. Moreover, it is a free word order language [59, 143, 196] and does not use capitalised letters for upper and lower case discrimination. Moreover, the script is context sensitive i.e. letters change their shape depending on the adjoining letters.

[4] https://en.wikipedia.org/wiki/Indo-Aryan_languages#cite_note-ethnologue-4 - Last visited: 13-April-2019

[5] https://en.wikipedia.org/wiki/Urdu - Last visited: 20-March-2019

1.3 Problem and Significance

To develop high-quality large-scale freely available resources for the under-resourced Urdu language is non-trivial, since it is a challenging, expensive, slow, laborious, difficult and time-consuming task. Therefore, relatively little research work has been reported on the development of large scale semantically annotated corpora, NLP pre-processing tools, and methods for Urdu (see Chapter 2); most of the work in the field has been done for English. Furthermore, large semantic lexical resources based on the USAS based semantic classification scheme (see Section 1.1) have not been attempted before for Urdu. This thesis addresses this gap in the research by presenting the Urdu Semantic Tagger (hereafter the US Tagger) by incorporating semantic lexicons (single and multi-word), pre-processing tools (tokenizers, Part-Of-Speech (POS) tagger, and lemmatizer), and methods (semantic field disambiguation and multi-target classifiers) which use semantic fields as the organizing principle and are thus a unique resource created for the Urdu language. Furthermore, the US Tagger is tested on a newly developed semantically annotated corpus.

In addition to describing and evaluating the US Tagger, semantic resources, and supporting tools, this thesis will also outline their further development and suggest new applications for them. The US Tagger can be practically applied in many Urdu NLP and HLT applications and tailored for various purposes, as will become evident in this thesis.

The US Tagger is the thirteenth non-English semantic tagger in the UCREL Semantic Analysis System (USAS) (see Section 1.1) framework. At present, there are equivalent semantic taggers based on semantic lexicons available for twelve languages, and the framework is continuously being expanded to cover new languages. The findings of this thesis (see Section 1.5), in regard to both the lexicon development where different automatic or semi-automatic approaches have been used, Urdu natural language processing tools, semantically annotated corpus as well as multi-target classifiers, the software development of the US Tagger, will benefit this work, especially when the USAS framework is extended to other languages which, like Urdu, are highly inflectional and morphologically rich. Moreover, now that there are equivalent semantic taggers available for many languages, this opens up exciting possibilities for the development of various multi-lingual applications in addition to mono-lingual Urdu applications.

1.4 Objective and Research Goals

The aim of this research work is to develop an Urdu semantic tagger (or tagging tool) which can perform semantic analysis of Urdu text, by investigating whether and how it is possible to create semantic resources for Urdu semantic tagger which are compatible with the existing English semantic tagger pipeline. In this regard, the following research goals can be formulated:

- Explore the in-depth problem of automatic semantic tagging task for Urdu text in order to see what new methods and frameworks are required.
- Develop efficient algorithms and methods as well as extract rules for automatically detecting word and sentence boundary as well as to assign POS tags to Urdu language text.
- Develop large-scale supporting resources (e.g. lexicons, word lists, and annotated corpora) for Urdu word, sentence segmentation and POS tagging.
- Develop annotated training and testing corpora for multi-target classifiers and to evaluate the US Tagger.
- Create an Urdu semantic tagset for Urdu semantic tagging task.

- Develop Urdu semantic lexicons (single and multi-word) using automatic and semi-automatic approaches as well as supporting resources and to determine how extensive are these lexicons in terms of lexical coverage.

- Evaluate new methods for the semantic tag disambiguation task for Urdu text.

- Development of a new software framework for the US Tagger and its evaluation.

1.5 Contributions

The main contributions of this research work are:

1. **Development of various Urdu natural language tools for the semantic tagging task along with supporting resources.**

   The main lexical unit for Urdu semantic tagger is sentence and word/token. Once properly tokenized, these units are assigned POS tags to remove lexical and semantic ambiguity. The grammatical tags are assigned to tokenized data. Urdu text is written in a script which normally has no spaces between words. The word boundary recognition problem in Urdu text tokenization faces two main challenges; (i) the space insertion problem, where there is extra space between two different words (so need to remove space to form a single token) and (ii) the space omission problem, where there is no space between two different words (so need to insert space to detect two different tokens).

   Contribution: State-of-the-art techniques have been developed for Urdu text tokenization to solve space omission problem. The one which is adopted in this thesis is a character *bi*-gram morpheme match base approach which generates all possible sequences of tokens of the input text. Then using trigram maximum likelihood estimation (MLE) to select the most optimised list, with back-off to *bi*-gram MLE with a Laplace smoothing estimation to avoid the data-sparseness. Evaluation is performed on a self-created corpus. Space insertion is solved using a dictionary look-up approach. Furthermore, the morphemes and complex words dictionaries are generated either automatically or semi-automatically. In addition to this, a large training and testing corpus for word tokenization task has also been presented. On the other hand rule based techniques have been developed for Urdu sentence tokenization task. These are rules, dictionary look-up and regular expressions. Furthermore, a manual sentence annotated dataset has been developed for the evaluation of Urdu sentence tokenizer.

Contribution: To assign grammatical categories or tags to a tokenized word, various off-the-shelf Urdu POS taggers (or tagging methods) have been presented. Therefore, to train and test statistical POS taggers a corpus of 200K words has been annotated using semi-automatic approach to assign CLE Urdu POS tags [225]. Furthermore, 80% of the data is used to train two different statistical models, the tri-gram Hidden Markov Model, and Maximum Entropy statistical taggers. Furthermore, Laplace and Lidstone smoothing estimation methods for unknown words have also been explored.

2. **Creation of Urdu semantic tagset.**

   Porting a USAS semantic classification scheme into another languages is not an easy task. The selection of an appropriate semantic classification scheme will have effects on the quality of semantic lexicons and eventually on semantic tagger accuracy.

   Contribution: Machine translation and bilingual dictionaries have been used to automatically translate an English semantic tagset into the Urdu language.

Automatically translated Urdu tagset is further manually verified by two annotators.

3. **Development of Urdu semantic lexicons, English-Urdu sentence aligned parallel corpus and Urdu monolingual corpus.**

   The US Tagger relies heavily on semantic lexical resources as its knowledge source. One important task for the US Tagger is to generate similar single-word and multi-word lexical resources. However, manually creating them is time consuming, laborious, slow, expensive and may be subject to annotator biases. A major challenge is to create these resources with less effort and in a short time-span.

   Contribution: In this research various methods for rapidly constructing large-scale and high-quality Urdu semantic lexicons (single-word and multi-word) have been proposed. These automatic or semi-automatic approaches for constructing semantic lexicons for the Urdu language are: (i) mapping, (ii) crowdsourcing, (iii) machine translation, (iv) GIZA++, (v) word embedding, and (vi) named entities. Four (Crowdsourcing, machine translation, word embedding, and named entities) of these methods have not been used before for the creation of a semantic tagger in a new language, and in addition their combination is also novel. In addition to this, a large English-Urdu sentence aligned parallel corpus and an Urdu monolingual corpus has also been generated for GIZA++ and word embedding approaches.

4. **Development of a multi-target semantically annotated corpus and multi-target classification methods.**

   In the final evaluation step in order to test the US Tagger performance and the lexical coverage, a benchmark semantically annotated corpus is required.

Contribution: To develop a large scale semantically annotated corpus by collecting text from various domains and then annotated using a semi-automatic approach. The corpus is annotated at word level with the USAS classification scheme. A tagged word can have one to nine Urdu semantic field tags associated with it. These tags have been used to indicate multiple membership categories from the USAS semantic taxonomy. i.e. different components of one sense. Furthermore, an inter-annotator agreement is also calculated on the proposed multi-target corpus. To demonstrate how a proposed corpus can be used for the development and evaluation of Urdu semantic tagging methods, various features are extracted (local, topical and semantic) from the newly created corpus and applied seven different supervised multi-target classifiers on them. Furthermore, the same test corpus is used to evaluate the US Tagger accuracy, precision and lexical coverage.

5. **Development of the Urdu semantic tagger, semantic tag disambiguation methods and evaluation.**

   Based on Urdu semantic lexicons, a semantic annotation tool is required to provide semantic analysis of Urdu language text. Moreover, as in the case of grammatical tagging, the task of the US Tagger is subdivided primarily into two phases, (i) tag assignment – attaching a set of potential semantic tags to each word or token, and (ii) tag disambiguation – selecting the contextually appropriate from the set provided in the first phase. Evaluation of the annotation tool and lexicon is also required to measure its performance.

   Contribution: The US Tagger has also been developed in this thesis by integrating Urdu semantic lexicons, NLP tools, and disambiguation rules. Furthermore, various baseline statistical and knowledge based approaches have been applied to improve semantic tag disambiguation, i.e. POS and general-likelihood. The

US Tagger is then used to evaluate Urdu semantic lexicons and annotated text on various corpora.

1.6 Organization of the Thesis

The remainder of this thesis consists of the following five chapters as follows:

- **Chapter 2:** Background and Related Work

  This chapter describes the background required to establish a semantic tagging task for Urdu text. To start with, this thesis outlined the general framework for the field of semantic tagging by introducing the most related concepts. Thereafter, an overview of the state-of-the-art techniques and resources for bilingual and multilingual semantic tagging task has been provided. Afterwards, this chapter describes USAS, another semantic tagging tool originally developed for the English language which is based on the idea of semantic fields or tags, as a model for the development of the Urdu counterparts. In addition to this, an overview of existing state-of-the-art lexical resources have been presented. Moreover, an overview of existing NLP tools, Urdu word and sentence tokenizers, POS taggers, and corpora and other resources required to build a framework for the US Tagger has been given. Finally, this chapter conclude by giving a brief account of measures used to evaluate the tools as well as tagger and multi-target classifiers.

- **Chapter 3:** Urdu Natural Language Tools

  With the preliminaries dealt with, this chapter begins with an overview of sentence tokenizer, word tokenizer, and part-of-speech tagger. Following that it began by looking at the initial phases of the research and development process, and, subsequently, it provide a brief summary of the development and the structure of the Urdu natural language tools to place the work in its immediate context. Although these tools are not the main focus of the US Tagger, however, it is essential to develop these pre-processing tools for the Urdu language, since there are no word or sentence tokenizers and POS taggers which are freely and publicly available. Furthermore, it is not possible to perform semantic tagging and the application of various semantic disambiguation methods without the availability of these pre-processing tools. Thereafter, this chapter provide a detailed description of the principles and practices which have been followed when creating these tools. In addition, this chapter also provided the detailed process of creating the supporting dictionaries and corpora which are required for these pre-processing tools.

This chapter concludes with an evaluation process for the newly created Urdu natural language tools. The results have demonstrated that the newly created Urdu natural language tools performed well on several test corpora.

- **Chapter 4:** Semantically Annotated Corpus and Methods

  This chapter reports the development of the semantically annotated corpus which is developed for the evaluation process of the US Tagger. Each word or multi-word expression (MWE) in the US Tagger output may appear with multiple possible semantic field tags to show the different meanings which can be taken in different contexts, and these are left in the output in rough likelihood order if disambiguation methods cannot resolve the correct sense. For such systems, multi-target classifiers can be potentially beneficial, where the word(s) may be associated with multiple labels or tags. Subsequently, several features have been extracted and applied various state-of-the-art multi-target classifiers to the semantic disambiguation task and this can be seen as an important step towards more robust wide coverage candidate semantic tag assignment before any final disambiguation.

Finally, a detailed statistics of different techniques applied on multi-target classifiers have given.

- **Chapter 5:** Urdu Semantic Tagset, Lexicons, the US Tagger and its Evaluation

  This chapter describes the Urdu semantic tagset, lexical resources and software framework of an Urdu semantic tagger along with its evaluation process. This chapter begun by looking at the initial phases of the research and development process, and, subsequently, it provide a short overview of the development and the structure of the Urdu semantic tagset in order to place the work in its immediate context. Thereafter, a detailed description of the principles and practices is provided which has been followed when creating the Urdu semantic lexical resources.

  This chapter also reports the results of the US Tagger when integrated with six Urdu semantic lexical resources, disambiguation methods, and with various natural language tools (see Chapter 3). This chapter also briefly summarizing the US Tagger framework which is developed for the evaluation process of Urdu semantic tagging task. This evaluation is carried out on the corpus mentioned in Chapter 4 as well as on the most frequent words of the Urdu monolingual corpus (see Chapter 2).

  These experiments measure the lexical coverage and accuracy by indicating the number of words which are covered by the single and multi-word semantic lexicons as well as by indicating how well these lexicons and tools perform when they are integrated into the US Tagger, respectively. Finally, this chapter analysed the errors which occurred in the US Tagger evaluation process.

- **Chapter 6:** Conclusions and Further Directions

  This chapter provides the conclusions. The first section comprises a summary of the thesis. Thereafter, research questions have been revisited. Finally, the chapter concluded by suggesting further work on the semantic lexical resources and also envisage new applications for the US Tagger.

1.7 Dissemination and Exploitation

1.7.1 Published Work

Publications produced during this research work are as follows:

- "Scott Piao, Paul Rayson, Dawn Archer, Francesca Bianchi, Carmen Dayrell, Mahmoud El-Haj, Ricardo-María Jiménez, Dawn Knight, Michal Kren, Laura Löfberg, Rao Muhammad Adeel Nawab, Jawad Shafi, Phoey Lee Teh and Olga Mudraya. (2016) Lexical Coverage Evaluation of Large-scale Multilingual Semantic Lexicons for Twelve Languages. In proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC2016), Portoroz, Slovenia, pp. 2614-2619."

1.7.2 Submitted Papers

- "Jawad Shafi, R.M.A. Nawab, Paul Rayson, H. Rizwan Iqbal. Urdu Natural Language Toolkit (UNLT). Natural Language Engineering (NLE)."
- "Jawad Shafi, R.M.A. Nawab, Paul Rayson. Semantic Tagging for the Urdu Language: Annotated Corpus and Multi-Target Classification Methods. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)."

Chapter 2

Background and Related Work

2.1 Introduction

In Chapter 1, various types of semantic annotation tools have been described: (i) some are designed to identify the topic or themes while others are designed to extract specific partial information from given texts and (ii) others are designed to identify semantic categories of all lexical units based on a given classification scheme. An in-depth discussion of the semantic annotation tools proposed for the first type of task will be beyond the scope of this chapter. Therefore, this survey is restricted to the issue of semantically rich text analytical tools, methods, and resources (based on latter type of task) on a natural language text which is the focus of this research work.

The rest of this chapter is divided into four parts. In the first part (see Section 2.2) basic background is given by defining the most important related concepts which are required to describe an Urdu semantic tagger. In the second part, related work is subdivided into eight sub parts (Sections 2.3.1 to 2.3.8), these sub-parts describe corpora as well as techniques for WSD and semantic tagging tasks, lexical resources, types of natural language toolkits for English and European languages, Urdu word tokenization, sentence tokenization and POS tagging techniques for Urdu, and characteristics of datasets developed for several Urdu NLP tasks. The third part (Sections 2.4) presents USAS as well as the English semantic tagger along with its components and its further extension. Finally, the last part (see Section 2.5) gives an overview of the commonly used evaluation measures which have been used to evaluate the performance of the semantic tagger, multi-target classifiers, and Urdu NLP tools.

2.2 Fundamental Concepts

This part introduces the most important concepts related to the semantic tagging task. It starts from the most general concept, which is computational linguistics, and then moves on to more successively specialized ones.

2.2.1 Computational Linguistics

Computational Linguistics (CL) is an interdisciplinary field concerned with the statistical or rule-based modelling of natural language text from a computational perspective, as well as the study of appropriate computational approaches to linguistic questions. CL is a relevant task for a wide range of research areas and practical applications, for instance NLP, Human Computer Interaction (HCI), text mining, data science etc. However, the focus of this research work is in the field of NLP.

2.2.2 Natural Language Processing

Natural language processing is a sub-field of computer science, and artificial intelligence concerned with the interactions between computers and humans using natural languages. It deals in particular with how to program computers to process and analyse large amounts of natural language data. The most commonly researched areas in the NLP field are syntax annotation (POS tagging, word/sentence tokenization, stemming, parsing, terminology extraction), semantics (WSD, sentiment analysis, machine translation, semantic tagging), discourse (text summarization, discourse analysis, coreference resolution), and speech (speech recognition, speech segmentation, and text-to-speech). However, the focus of this research work will be on semantic tagging of natural language text rather than speech.

2.2.3 Text Annotation

Text can also be provided with additional linguistic information, called *annotation*, or it can be defined as the practice of adding interpretative linguistic information to a text [111]. There are different types of text annotation. However, this thesis deals with computational linguistics annotations, which will be discussed in the following subsections. Other types of annotation are the textual and extra-textual annotation, orthographical annotation etc.

2.2.4 Part-Of-Speech Tagging

The most basic type of linguistic annotation is POS tagging which is also known as grammatical tagging or morpho-syntactic annotation. An annotation program automatically assigns each lexical unit in a text with a tag that indicates its part of speech. The information about the part of speech is valuable for a number of NLP sub-fields, for instance, WSD and semantic tagging, and so on [165]. In this research work, the POS tagging has been used to resolve semantic tag ambiguity.

2.2.5 Semantic Tagging

Semantic tagging can be defined as a dictionary-based process of identifying and labelling the coarse-grained meaning of words in a given text. In research [75], this process parallels that of grammatical tagging except that it is more abstract and more difficult to achieve. Semantic tagging has received increasing attention during recent decades and various tools (semantic taggers) have been developed for this purpose for different languages. Semantic taggers have several applications such as terminology extraction, machine translation, bilingual and multilingual MWE extraction, monolingual and multilingual information extraction, as well as in automatic generation, interpretation, and classification of language (for more applications see Section 2.4.1). There are different techniques (see Section 2.3) to carry out the semantic tagging task. However, in this thesis, approach to carrying out semantic tagging is based on semantic fields (see Section 2.2.6). Other semantic tagging and annotation techniques are defined in Section 2.3. Moreover, the main research focus of this thesis is on creating computational linguistic resources for semantic tagging and the semantic tagger.

2.2.6 Semantic Fields

Semantic fields can be defined as "a theoretical construct which groups together words that are related by virtue of their being connected at some level of generality with the same mental concept" [74]. Words which belong to the same semantic field can be synonyms, antonyms, hyponyms[1], meronyms[2], or expressions that are associated with each other in one way or another. The semantic tagging tool which have been reported in this thesis is based on semantic fields as tags.

[1] A word that is more specific than a given word.

[2] A word that names a part of a larger whole.

2.2.7 Word and Sentence Tokenization

Tokenization is the act of breaking up a sequence of strings into pieces of words or sentences that are known as tokens. Tokens can be individual words, multi-words or even whole sentences. In this work, the tokens become the input for the US Tagger so that particular words can be tagged with semantic fields or help to resolve semantic tag ambiguity.

2.3 Related Work

The research field closely related to semantic tagging task is WSD (see Section 2.2.6) [180]. Therefore, in this section, various corpora and methods developed for WSD and semantic tagging tasks have been presented. Thereafter, this part will focus on related work on lexical resources, natural language toolkits, word and sentence tokenization methods, POS tagging methods and datasets because these have also been developed in this research.

2.3.1 Corpora and Techniques for Word Sense Disambiguation

2.3.1.1 Corpora

To develop large-scale freely available standard evaluation resources to investigate the problem of WSD is a non-trivial task. In previous literature, efforts have been made to develop benchmark corpora for the WSD task. An in depth discussion of all the WSD corpora will be beyond the scope of this study. Therefore, this subsection only present some of the most prominent studies.

The most prominent effort in developing standard evaluation resources for WSD task is a series of SensEval competitions[3]. The outcome of these competitions is a

[3] http://www.senseval.org/ - Last visited: 18-February-2019

set of benchmark corpora for the WSD task. The SensEval competitions on WSD task have been organized from 1998 to 2004. The competitions focused on two main types of WSD: (i) all-words WSD task and (ii) lexical sample WSD task. The languages for which WSD corpora have been developed are: English, Basque, Italian, Japanese, Korean, Spanish, Swedish, Chinese and Romanian. The lexical resources or dictionaries that are used in the development of WSD corpora include WordNet. SensEval WSD corpora are large and freely available for research purposes [146].

In previous literature, other than SensEval, efforts have been made to develop WSD corpora for English and other languages such as the SEMCOR WSD Corpus [104], Google WSD Corpus [251], and DutchSemCor WSD corpus [238]. However, for the Urdu WSD task, only two corpora have been found in previous research, an Urdu sense tagged corpus [234] and Urdu Lexical Sample WSD (ULS-WSD-18) corpus [202]. The Urdu sense tagged corpus [234] is developed for the Urdu all-words WSD task and contains 17K manually sense annotated sentences with 2,285 unique senses by a single annotator over a period of 10 months. ULS-WSD-18 corpus has been developed for the lexical sample WSD task and contain 7,185 manually sense tagged sentences for 50 target words (senses of tagged words are extracted from a hand crafted dictionary called Urdu Lughat Board [32]) by three different annotators.

2.3.1.2 Techniques

WSD research is closely related to a work reported here, as a consequence different WSD techniques have been used to resolve semantic tag ambiguity such as those mentioned in [180]. Therefore, in this section, an overview of WSD techniques are provide.

Over the years, many different WSD techniques have been proposed, and they can be classified into the following four categories: (i) Artificial Intelligence (AI) techniques, (ii) Knowledge-based techniques, (iii) Corpus-based techniques, and (iv) Hybrid techniques [146, 180].

Prominent efforts to tackle WSD based on AI techniques began in the early 1970s via large-scale language understanding [163, 88]. For example, Wilks [245] described a "preference semantics" system, using selectional restrictions and lexical semantics (*case frames*[4]) to find a set of senses for a word in a sentence.

Knowledge-based WSD techniques use lexical resources to provide contextual knowledge which is essential to determining the appropriate sense(s) of polysemous[5] words. These resources can be thesauri [199], machine-readable dictionaries [173], or computational lexicons [134, 180]. A wider survey of these resources can be found in Section 2.3.3.

Current state-of-the-art techniques for the resolution of word sense ambiguity stem from the field of Machine Learning (ML). These ML (or corpus-based) WSD techniques can be primarily classified into: (i) unsupervised, (ii) semi-supervised, and (iii) supervised.

Unsupervised techniques have the potential to acquire contextual information directly from knowledge acquisition [72] i.e. senses can be deduced from untagged raw text using similarity measures[6] [130] based on the idea that occurrences of the same sense of a word will have similar neighbouring words. Example techniques for unsupervised WSD are co-occurrence and spanning tree-based graphs [6], word clustering [36], and recently developed neural network language models [161].

[4] These contain information about words, their relation to other words, and their roles in individual sentences

[5] Words having many meanings

[6] Clustering word occurrences and then classifying new occurrences into the induced clusters.

Semi-supervised ML WSD techniques usually train a classifier with a small set of labelled examples and then bring further improvements in the process of iterative learning i.e. a classifier is retrained, and this learning process continues until convergence. There have been a number of studies which have used semi-supervised ML WSD techniques, for instance, [153] used label propagation algorithm for WSD, whereas [251] used sequence learning neural network to differentiate different senses.

Supervised single-label classification techniques apply where each word is only associated with a single label or class, that is, they assign the appropriate sense to a target word. There have been a number of research studies where single-label ML classification techniques are applied for English and European language WSD tasks, for example, [5] used decision lists [197], whereas, [137] used C4.5 (decision tree) and concluded that it outperformed all the other single-label ML techniques, simple Naïve Bayes is applied in [40], [232] (based on neural networks), k-nearest neighbour [61]. A complete overview and discussion of all single-label classification techniques will be beyond the scope of this section. Therefore, this section present the single-label classification studies adopted for Urdu. Only two such studies have been found in the previous literature: (i) Abid et al. [4], and (ii) Naseer and Hussain [145].

The authors in [4] developed a lexical-sample based WSD system using single-label classifiers including, Naïve Bayes, Decision Tree, and Support Vector Machines with POS tags and bag-of-words as features. Twenty named entities are used to evaluate the system performance. The reported F_1 scores for Naïve Bayes, decision tree, and support vector machines are: 71%, 34%, and 34% respectively.

Another study is conducted by [145] using Naïve Bayes classifiers for the development of lexical-sample WSD system. The authors resolved ambiguity on four words including three verbs and one noun. Bag-of-words and POS tags are used as features and the reported highest F_1 score is 95.15%.

The final of four categories of techniques describe here is hybrid approach, representing studies using a combination of the various above-mentioned techniques. A number of research studies have been carried out using hybrid ensemble techniques, for instance, [222] used LDOCE (Longman Dictionary of Contemporary English [173]) with information derived from corpora etc.

It can be observed from the above discussion that a number of WSD techniques have been used for sense resolution. However, these techniques have several shortcomings as follows: (i) AI techniques are tested on a single or only a few sentences, therefore, their effectiveness on real text is impossible to determine, (ii) knowledge-based methods are a useful way to represent linguistic or lexicographic knowledge of word sense ambiguity, and they have produced good results. However, they are not very robust as natural language is a dynamic phenomenon i.e. new words and senses are added and old ones become archaic or outdated, thus, they lack complete coverage as new words or senses may not exist in these resources, (iii) lexical resources are readily available for English and other European languages, but not for under-resourced Urdu[7] language, (iv) semi-supervised ML techniques have a major drawback in that they lack a method for selecting optimal values for classifiers i.e. the number of iterations and labelled examples [151]. Further, these types of techniques are tested on small corpora [146], (v) unsupervised ML techniques automatically acquire contextual information and are often erroneous and noisy [4], thus degrading system performance, (vi) hybrid techniques require several resources, which is difficult for resource-poor languages, and (vii) supervised single-label classifiers can assign only one tag or label.

[7] A recent study [213] involved Urdu semantic lexicons (both single and multi-words) of 2K entries, however, it is lacking wide lexical coverage.

2.3.2 Corpora and Techniques for Semantic Tagging

2.3.2.1 Corpora

A number of studies in the literature have devoted a great deal of research effort for the development of semantic annotation, such as, Semantic Role Labelling, Named Entity Recognition, Content Analysis, sentiment analysis etc. Usually, these semantic annotation systems have used annotated corpora and more recently BabelNet[8] [149] to induce or cluster different meanings or senses [138]. The most prominent effort in developing standard evaluation resources for various semantic annotation tasks are the series of SemEval competitions for English and other languages [146, 147].

The outcome of these competitions (from 2007 to date) are a set of benchmark corpora with semantic annotations for various NLP tasks, Information Extraction, Sentiment Analysis and Opinion Mining (a.k.a sentic computing [43]), Textual Semantic Similarity, Word Semantic Similarity, Question Answering etc. (SemEval-2012[9], SemEval-2013[10], SemEval-2014[11], SemEval-2015[12], SemEval-2016[13], SemEval-2017[14], and SemEval-2018[15]) for a variety of languages including English, French, Italian, Dutch, Chinese, Arabic and several others. Table 2.1 summarizes the corpora involved in the SemEval workshop series along with their properties.

[8] Multilingual semantic network created from the algorithmic integration of WordNet and Wikipedia.

[9] https://www.cs.york.ac.uk/semeval-2012/index.html - Last visited: 18-February-2019

[10] https://www.cs.york.ac.uk/semeval-2013/ - Last visited: 18-February-2019

[11] http://alt.qcri.org/semeval2014/ - Last visited: 18-February-2019

[12] http://alt.qcri.org/semeval2015/ - Last visited: 18-February-2019

[13] http://alt.qcri.org/semeval2016/ - Last visited: 18-February-2019

[14] http://alt.qcri.org/semeval2017/ - Last visited: 18-February-2019

[15] http://alt.qcri.org/semeval2018/ - Last visited: 18-February-2019

Table 2.1 Summary of the available SemEval corpora

Workshop	No. of Tasks	Corpus for domain	Languages*
SemEval-2007	18	cross-lingual, frame extraction,	ar,
		information extraction,	en,
		lexical sample, metonymy,	zh,
		semantic annotation, semantic	ca,
		role labelling, sentiment analysis,	es,
		lexical substitution,	tr
		semantic relations,	
		time expression, WSD	
SemEval-2010	17	co-reference, cross-lingual,	ca, it,
		ellipsis, information extraction,	zh, ja
		lexical substitution, metonymy,	nl,
		semantic relations, semantic role labelling,	en,
		sentiment analysis, textual entailment,	fr,
		noun compounds, parsing,	es,
		time expressions, WSD	de

SemEval-2012	8	common sense reasoning, lexical simplification, relational similarity, spatial role labelling, semantic dependency parsing, semantic and textual similarity, role labelling, sentiment analysis, time expression, WSD	zh, en
SemEval-2013	14	temporal annotation, sentiment analysis, spatial role labelling, noun compounds, lexical sample phrasal semantics, time expressions, response analysis, biomedical, cross and multilingual WSD, word sense induction, cross and multilingual WSD,	ca, de, it, es, fr, en de
SemEval-2014	10	compositional distributional semantic, grammar induction for spoken dialogue systems, L2 writing assistant, supervised semantic parsing, clinical text	en, es, fr, de,

		analysis, sentiment analysis in Twitter, multilingual semantic textual similarity, cross-level semantic similarity, sentiment analysis, semantic dependency parsing	nl,
SemEval-2015	17	text similarity and question answering, time and space, sentiment analysis, WSD, word sense induction, text similarity learning semantic relations	en, es, ar, it
SemEval-2016	14	text similarity and question answering, sentiment analysis, supervised semantic parsing, semantic annotation, semantic taxonomy	en, it nl, zh fr, tu es, ru
SemEval-2017	12	multilingual semantic textual similarity, cross and multilingual, text similarity and question answering,	en, it, fr,

		sentiment analysis in Twitter	ja,
		semantic taxonomy,	de,
		fine-Grained sentiment analysis	es
		on financial Microblogs and News,	
		Hashtag wars learning a sense of humour	
		analysis, semantic dependency parsing	
SemEval-2018	12	affect and creative language in tweets,	en,
		co-reference, information extraction,	it,
		lexical semantics ,	es
		and reading comprehension and reasoning,	

*ISO 639-2 two letter codes: en: English, fr: French, it: Italian, cs: Czech, nl: Dutch, ja: Japanese, es: Spanish, ko: Korean, sv: Swedish, ca: Catalan, ro: Romanian, da: Danish, et: Estonian, eu: Basque, zh: Chinese, ar: Arabic, tr: Turkish, de: German, ru: Russian

2.3.2.2 Techniques

Semantic tagging (see Section 2.3.2) is certainly an effective method, but it also faces the difficulty that the same object or concept can be referred to in a number of ways; the identification of the meaning of a word is not necessarily an easy task as defined in [120] (pp. 43-44). By way of illustration, the animal "cat" can also be called kitten, pussy, and mog. This phenomenon is related to synonymy[16]. On the other hand, one single word can refer to a number of concepts, such as the polysemous[17] noun "bass" can refer both to a type of fish, tones of low frequency, and to a type of instrument. Likewise, the homonym[18] word "book" can refer to the noun "to read" as well as to the verb "reservation". Such kinds of ambiguity can sometimes present difficulty to human beings. There is no doubt that human can differentiate the various meaning of such words with the aid of their knowledge of the world. However, this type of task is non-trivial for computer programs and thus have presented a serious challenge to NLP research community.

By way of illustration, if someone is using a query on a search-engine to find information about a certain term and enters into the search field a word "bass", which is a polysemous, than s/he may end up with considerable amounts of unnecessary contents in the search results, such as many information related to the term fish whereas s/he actually wants contents related to a type of musical instrument. In NLP the task of selecting the relevant sense for a lexical unit (word) from multiple senses is referred to as WSD [146]. Semantic tagging (see Section 2.3.2) is another way of carrying out this task. Similarly, if a search engine is used to search the word "crane", it might be the possibility that it would return hits for the word meaning a bird in it, and the search engine might ignore websites containing text relevant to the

[16] Two words that can be interchanged relative to that context.

[17] A word having many meanings.

[18] Two or more words are homonyms if they are pronounced or spelled the same form but have different meanings.

words with a type of construction equipment or to strain out one's neck. For such cases, the semantic tagging task can have significant benefits i.e. it can help to find out all the relevant information and filter out the irrelevant ones.

The approach which have been adopted for the semantic tagging task in this thesis is based on semantic fields (see Section 2.2.6). Words which belong to the same semantic field can be synonyms, antonyms, hyponyms[19], meronyms[20], or expressions that are associated with each other in one way or another. Synonymy "near or close" and antonym "near and far" are relations which exist between two words. The relations can also be hierarchical, as in the case of hyponyms and meronyms, in which some words have a more general meaning whereas some have a more specific meaning, when they are referring to the same entity. Hyponyms is the "kind of" relation. For example, the most general term "garment" is on the top level of this hierarchy, and it is referred to as the hypernyms and the more specific terms "coat" on the level below are referred to as the hyponyms. The second level terms, in turn, are hypernyms of even more specific terms "parka" on the third level. By comparison, meronymy is the "part of" relation, where phenomena are analysed into parts. Here the superordinate term "shirt" refers to the complete entity, whereas the terms on the lower levels represent its parts "sleeve" on the following level and then "cuff" on the subsequent level. Consequently, the words "garment","coat", "parka", "shirt", "sleeve", and "cuff" as well as, for instance, the words "attire", "hem", "trousers", "undress", "dressed", "stark naked", and "haute couture" could all be considered to belong to the same semantic field. If a semantic tag (label) is attached, to every word in a text indicating the semantic field into which each falls, it will then be able to extract all the related words from a text by querying on the specific semantic field. There is a problem however, in the classification of words, since not all of them

[19] A word that is more specific than a given word.

[20] A word that names a part of a larger whole.

always fall conveniently into the predefined semantic fields. The authors in [74] (pp. 58–59) point out with the example word "sportswear". This word could be classified in the semantic field of clothing equally well as in the semantic field of sports. These types of systems are discussed in more detail in Section 2.4.1.1.

A collection of words classified into semantic fields can be designated as a "semantic annotation system or semantic tagging". Semantic annotation systems are something of a compromise between, on the one hand, attempting to mirror how words are believed to be organized into relationships in the human mind, and on the other hand, the need for usable annotated corpora and reference works by linguists and other scholars [75] (pp. 54-55). The authors in [74] further observed that the majority of existing semantic annotation systems consist of very similar basic categories, but they differ from each other in terms of hierarchy (in other words, the structure of the categories) and in terms of granularity (in other words, the level of detail; how many categories the system distinguishes). The semantic tagging approach which will be describe in the thesis is based on semantic fields.

Texts can be annotated with semantic field information in three different ways depending on the level of automation [75] (pp. 62). The first option is to attach all annotations in the text manually. The second option, computer-assisted tagging, represents a semi-automatic form of manual tagging which is supported by a computer-readable lexicon containing possible semantic fields for given words. Such systems may also contain a limited amount of automatic WSD mechanisms. In this case, the computer is used to assign candidate semantic field tags to all the words in a text on which there is already information, and it leaves for manual treatment only those words that it does not recognize or which remain ambiguous after the application of disambiguation methods. The third option is a fully automatic semantic tagger. This is a program which assigns the correct semantic fields automatically to all the known words in a text without any manual intervention and without leaving any words ambiguous. The semantic tagging approach dealt with in this thesis utilizes the third option.

NLP Researchers Rayson and Stevenson (2008) [184] classified semantic tagging systems (or semantic field annotation) into four types of methods, (i) Artificial Intelligence (AI) based, (ii) Corpora based, (iii) Knowledge based, and (iv) Hybrid.

The first approach for semantic tagging is based on artificial intelligence approach and is popular in the 1970s, but declined after the 1980s, when they found to be impractical for large-scale language understanding [180].

The second approach is based on tagged corpora. Tagged corpora have also been used to induce or cluster different senses or meanings, aiming to identify and assign certain types of semantic information required by specific tasks. These types of semantic annotations have been researched in [54] and [15] identify the topic or themes of a given text. There are yet further studies [198, 241] which are conducted to extract specific or partial information, such as named entities, categories of relations between the specific named entities, and/or types of events.

A third approach of semantic tagging is via another group of knowledge-based sense inventories (WordNet, BabelNet, etc.), and for these semantic annotation can be used to assign fine-grained word senses [134]. WordNets have been developed for English, other European, and several Asian languages. These resources have also been ported to provide multilingual word sense inventories [146] for more detailed information on these resources see Section 2.3.3.

The fourth approach is based on hybrid based i.e. which are a combination of the previously mentioned methods (AI, corpus based or knowledge based). The semantic tagging system which has been reported in this thesis is based on hybrid approach, as it uses the knowledge-based and corpus based approaches in combination.

Other semantic tagging research aims to assign each content word with a semantic category using a component-based semantic classification scheme, for instance, tagging the word “mother” as [HUMAN, FEMALE, ADULT] and “paprika” as [NON-HUMAN, VEGETABLE], and so on. A number of research studies based on this concept have been reported previously, including [119].

2.3.3 Lexical Resources

To develop large-scale freely available lexical resources to investigate the problem of semantic analysis is a difficult task. However, there has been a number of research efforts in the past, where researchers have devoted a great deal of attention for developing benchmark evaluation lexical resources for semantic annotation task, although most are for the English and European languages. These lexical resources are thesaurus, machine-readable dictionaries, computational lexicons, and several others[21] [77]. A complete comparison of all these lexical resources used in semantic annotation task is beyond the scope of this thesis. Therefore, a short overview of those resources which are more commonly used in the semantic tagging task are as follow (for more details see [7]).

A thesaurus provides a relationship between words like, synonymy (for instance, a bus is a synonym of coach), antonym (e.g. good is an antonym of bad), and, possibly, further relations [105]. The thesaurus compiled by [199] (Roget’s International Thesaurus) is a famous example and the latest addition contains 250,000 word entries, which are organized in six classes and 1,000 categories. It is most widely used in WSD semantic annotation task [146] and to calculate semantic similarities [89].

[21] https://dkpro.github.io/dkpro-wsd/lsr/ - Last visited: 18-February-2019

Machine-readable dictionaries have become a popular source of knowledge for NLP processing since the 1980s. Among others, the LDOCE [173] is among the list of most widely used manually created machine-readable dictionary before the diffusion of WordNet [135]. However, LDOCE as a semantic resource is not as widespread[22] [207].

Computational Lexicons (CLex), are divided into two types: (i) fine-grained, and (ii) coarse-grained CLex. Fine-grained CLex, are often considered one step beyond commonly available machine-readable dictionaries as they encode rich semantic networks of concepts, called synsets. Among others, the WordNet [134, 135] manually created semantic lexicon provides gloss (textual definition of the synset with usage examples), and lexical and semantic relations (these relations connect pairs of word senses and synsets) for each synset. Presently, it is most predominant and considered a *de facto* standard in computational lexicons for semantic annotation e.g. WSD, thus, a most-used resource for English [146]. The latest version of WordNet 3.0, contains 155,000 words organized in over 117,000 synsets. Moreover, there has been a number of attempts where WordNet is developed for several other languages, EuroWordNet [237] provided an interlingual alignment between national wordnets, thus make WSD possible in several other languages. However, with the increase of multi-lingual digital text on the web research community are targeting multi-lingual settings, as a consequence BabelNet [148] is automatically created by linking Wikipedia to the most popular English WordNet CLex. Furthermore, BabelNet is a multi-lingual lexicalized semantic network which provides concepts and named entities in a multi-lingual setting and connected with large amounts of semantic relations, i.e. Babel synsets. WordNet [134, 135], EuroWordNet [237], and BabelNet

[22] http://www.ilc.cnr.it/EAGLES96/rep2/node18.html#SECTION03124000000000000000 - Last visited: 18-February-2019

[148] is used to assign fine-grained semantic concepts, which is often well beyond what may be needed in many NLP and HLT applications [88, 180, 146].

In contrast to fine-grained CLex, another group of lexicons containing lexical units classified with a set of predefined coarse-grained semantic fields, these type of CLex are known as USAS English semantic lexicons [180], created manually by a group of linguistics experts in the Benedict project[23]. In these lexicons (single-word and multi-word) each word is assigned with pre-defined semantic categories based on the lexicographically-informed semantic classification scheme. These lexicons are different from other CLex, since they do not provide word meaning definitions or fine-grained word senses, rather, they help to assign semantic fields based on the LLOCE (*Longman Lexicon of Contemporary English*) [129]. For the multi-lingual setting, efforts have been carried out to port semantic lexical resources in numerous languages such as Finnish [118], Russian [140] by means of manual efforts, however, manually developing semantic lexical resources for new languages from scratch is a slow, and expensive task, that may lead to erroneous annotation [213].

There are several other studies where efforts have been carried out to create new lexical resources from existing resources by finding transitive translation chains of words across several bilingual dictionaries [34]. In other remarkable attempts authors in [243] and [110, 109] have extracted dictionaries from corpora and different algorithms. There exist two main automatic methods to construct WordNets. The first method translates the synsets of WordNet to any of the target languages [25, 157]. The second method builds target language WordNets, then aligns it with the English WordNet [208]. In another study the authors in [160] used a metric to automatically evaluate machine translation and a corpus approach to build a lexical resource.

Directly related to a work which is research here, the growing body of automatic and semi-automatic approaches to generate USAS multilingual semantic lexicons are

[23] Under the EU funded IST-2001-34237, and two previous UK-funded projects.

the experiments reported in [165, 68]. The authors in [165] automatically generated semantic lexicons by transferring semantic tags from the existing USAS English semantic lexicon entries to their translation equivalents in various European languages via dictionaries and bilingual lexicons. Authors in [68] used crowdsourcing approaches by employing native language experts and non-experts, to generate a list of coarse-grained senses using a USAS based multilingual classification fields, and have generated coarse-grained semantic lexicons for different European languages. In [213] word-to-word alignment is performed on parallel corpora to extract Czech language semantic lexicon. In [213] multilingual WordNet is used to build the bilingual semantic lexicon for the Malay language by porting the semantic lexicons via synset IDs. For Arabic semantic lexicons, the authors in [136] used a combination of automatically and manually generated semantic lexicons.

2.3.4 Existing Natural Language Processing Toolkits

In previous studies, a number of NLP toolkits have been developed to solve common problems in language processing [47, 24, 35]. A complete comparison and discussion of all available toolkits will be beyond the scope of this study. Therefore, this section will present the comparison of five most popular, commonly and widely used, large-scale, multi-functional language processing toolkits that are built and distributed by academic projects: (i) Natural Language ToolKit (NLTK)[24] [221], (ii) Apache OpenNLP[25] [108], (iii) Stanford CoreNLP[26] [126], (iv) General Architecture for Text Engineering (GATE)[27]) [55] and (v) LingPipe[28] [45, 14].

[24] http://www.nltk.org/ - Last visited: 18-February-2019

[25] https://opennlp.apache.org/ - Last visited: 18-February-2019

[26] http://stanfordnlp.github.io/CoreNLP/ - Last visited: 18-February-2019

[27] https://gate.ac.uk/ - Last visited: 18-February-2019

[28] http://alias-i.com/lingpipe/ - Last visited: 18-February-2019

NLTK [31] is an open source, general purpose, and widely used NLP toolkit. This toolkit is written in Python and includes a collection of language analysis tools for the English language, including sentence tokenizer, word tokenizer, POS tagger, Named Entity Recognition (NER), text classifier, stemmer, parser, lemmatizer, coreference tagger, dependency parsing, machine translation, sentiment annotator, twitter processing etc. NLTK is easy to learn, well documented, with a collection of statistical, regular expression, rule-based, machine learning and N-gram language models based techniques. It also supports WordNet as a part of its word analysis. This toolkit also supports dozens of datasets[29] and is distributed under the terms of Apache License Version 2.0[30]. NLTK functional tools are used in various applications such as sentiment analysis [152, 206], annotating named entities in Twitter data [69], grammatical error correction [150] and many more.

Apache OpenNLP [108] is an open source and Java based toolkit which supports the most common NLP tasks. The pipeline of this toolkit consists of several text processing tools such as word tokenizer, sentence tokenizer, POS tagger, named entity extraction, chunker, parser, coreference tagger, lemmatizer, summarization, translation, feedback annotator and text classifier. This toolkit provides a large number of pre-built models for different languages. It is a machine learning and dictionary based toolkit with detailed documentation. In addition to the basic NLP tasks mentioned previously, the toolkit also has built-in support for various datasets, required for training/testing of different NLP tools. This NLP toolkit is available under the Apache License, Version 2.0[31]. Apache OpenNLP toolkit has been used by different companies in various applications, such as, for noun phrase coreference resolution [223], in microbiology and genetics [201], in question answering system [28] etc.

[29] http://www.nltk.org/nltk_data/ - Last visited: 18-February-2019
[30] http://www.apache.org/licenses/LICENSE-2.0 - Last visited: 18-February-2019
[31] http://www.apache.org/licenses/LICENSE-2.0 - Last visited: 18-February-2019

The Stanford CoreNLP [126] is a robust, high quality, easy to use as well as domain-specific linguistic analysis toolkit. It also supports text processing for many languages with the highest quality text analytics. This NLP toolkit is also open source, well documented and written in Java. Currently, it consists of many natural language analysis tools such as word tokenizer, sentence tokenizer, POS tagger, lemmatizer, Named Entity Recognition, parser, coreference tagger, sentiment annotator and text classifier. The CoreNLP supports several datasets and operates under the GNU General Public License V3 or later[32]. It is worth mentioning here that CoreNLP tools are trained using supervised machine learning, rule-based, regular expressions based, deep learning based, maximum entropy based, linear chain conditional random field based, neural network based, and probability based models. Again, this NLP toolkit has been used in a wide range of text processing applications e.g. text summarization [200], semantic parsing [19], sentiment classification [228], sentence embedding [244], document classification [249] etc.

GATE [55] is also an open source, well documented, stable, robust, scalable, Java based architecture, development environment and framework for natural language engineering tasks which has been available since the 1990s. It supports all types of computational linguistic tools for various human languages from a small start-up to large corporations, from an undergraduate language processing project to industrial research projects. GATE includes many tools for various NLP tasks such as word tokenizer, sentence tokenizer, POS tagger, classifier, stammer, lemmatizer, parser, chunker, NER, coreference tagger. This toolkit has built-in support for several datasets and licensed under the GNU Lesser General Public License[33]. It is important to note that GATE tools incorporate machine learning, deep learning, neural network, probability, rule-based and regular expression based methods. GATE is widely used

[32] http://www.gnu.org/licenses/gpl.html - Last visited: 29-October-2018

[33] https://www.gnu.org/licenses/lgpl-3.0.en.html - Last visited: 29-October-2018

for various research projects including life sciences and in biomedicine [56], topic and sentiment analysis [128], human computation for knowledge extraction and evaluation [209], information extraction [49], text processing in the cloud [224] etc.

Finally, LingPipe [14, 45] is a set of coherently organized general as well as domain specific tools for processing text using computational linguistics. This toolkit is also written in Java. The toolkit is stable, scalable, robust, reusable, well documented and multi-lingual. This toolkit supports the following tools: word tokenizer, sentence tokenizer, POS tagger, classifier, NER, sentiment analysis, parser, and chunker. LingPipe is mainly a collection of statistical models and incorporates supervised as well as unsupervised machine learning techniques. This toolkit supports online training and also incorporates different datasets for various tasks. It operates under a range of licenses which range from free[34] to perpetual server licenses. LingPipe has been used to carry out document classification of newspaper articles [123], development of biomedical ontologies [156], document summarization [124], author's attribution in legal proceeding of court [162] etc.

In short, the above mentioned toolkits are open source, written mostly in Java, include capabilities for word and sentence tokenization, POS tagging, parsing, chunking, identifying named entities, text classification, stemming, lemmatization, coreference resolution, sentiment analysis etc. These toolkits have built-in support for several datasets, operate under different licensing schemes and support single/multiple languages. These toolkits are applied in various domains and applications (see Table 2.2) which summarizes the characteristics of these toolkits.

[34] http://alias-i.com/lingpipe/web/download.html - Last visited: 29-October-2018

Table 2.2 Comparison of five widely used NLP toolkits

Features	**NLTK**	**OpenNLP**	**CoreNLP**	**GATE**	**LingPipe**
Sentence Tokenizer	Y[@]	Y	Y	Y	Y
Word Tokenizer	Y	Y	Y	Y	Y
POS tagger	Y	Y	Y	Y	Y
Classifier	Y	Y	Y	Y	Y
Stemmer	Y	N[@]	N	Y	N
Lemmatizer	Y	Y	Y	Y	N
Parser	Y	Y	Y	Y	Y
Chunker	Y	Y	N	Y	Y
NER	Y	Y	Y	Y	Y
Coreference	Y	Y	Y	Y	N
Sentiment	Y	Y	Y	Y	Y
Datasets	Y	Y	Y	Y	Y
Lexicon	WordNet	POS lexicon	N	WordNet	N
Code Language	Python	Java	Java	Java	Java
License[#]	alv	alv	gpl	lgpl	arfl, lpl
Languages[*]	en	en,de,es,nl da,pt,se	ar,zh,en fr,de	en,fr,zh,ar,cy hi,ro,ru,it,da ceb,bg	en,zh
Methods[+]	st,rb,ml re,ng	ml,db,me pml	ml,st,rb,re dl,me,crf,nn	ml,dl,nn st,rb,re	st,dl,re ml

[#] alv: Apache License Version 2.0, gpl: GNU General Public License v3 or later, lgpl: Lesser General Public License, arfl: Alias-i ROYALTY free license version 1, lpl: LingPipe Proprietary License v1.2, [@] Y: supported , N: not supported [*]ISO 639-2 two letter codes: en: English, fr: French, de: German, es: Spanish, pt: portugues, da: Danish, nl: Dutch, se: Northern Sami, ar: Arabic, zh: Chinese, cy: Welsh, hi: Hindi, ro: Romanian, ru: Russian, it: Italian, bg: Bulgarian, ceb: Cebuano, [+] st: statistical, rb: rule-based, ml: machine learning, re: regular expression, ng: N-gram, dl: deep learning, me: maximum entropy, crf: conditional random field, pml: perceptron based ml, db: dictionary based, nn: neural network

2.3.5 Existing Urdu Word Tokenization Approaches

In the existing literature, only a few studies are found which have addressed the problem of word tokenization for the Urdu language, these are [177, 66, 113, 190] and a recent one [254]. The study in [177] performs Urdu word tokenization in three phases. First, Urdu words are tokenized based on spaces, thus returning the cluster(s) of valid (single word) and invalid (merged word(s)[35]) Urdu words. Next, a dictionary is checked against valid and invalid words to assure the robustness of the word(s). If the word is present in the dictionary then it will be considered as a valid Urdu word, returning all single words. However, if the word is not matched in the dictionary then it is considered as a merged word, hence, needing further segmentation. In the second phase, the merged words are divided into all possible combinations, to check the validity of each produced combination through dictionary lookup. If it is present in the dictionary it will be considered as a valid word. The first two phases solve the problem of space omission (see Section 3.2), the third phase addresses the space insertion problem by combining two consecutive words and checking them in the dictionary. If the compound word is found in the dictionary, then it will be considered as a single word. This technique of word tokenization is tested on 11,995 words with a reported error rate of 2.8%. However, the efficiency of this algorithm is totally dependent on the dictionary (used to check whether a word is valid or not) and it is practically not possible to have a complete dictionary of Urdu words. Furthermore, if a valid word is not present in the dictionary then this technique will mark it as invalid, which will be wrong.

Durrani and Hussain (2010) [66] have proposed a hybrid Urdu word tokenizer[36] which works in three phases. In the first phase, words are segmented based on space, thus, returning a set of an orthographic word(s).[37] Further, a rule-based maximum matching technique is used to generate all possible word segmentations of the orthographic words. In the second phase, the resulting words are ranked using

[35] Combination of many words

[36] available at: http://homepages.inf.ed.ac.uk/dnadir/Urdu-Segmentation.zip - Last visited: 18-Dec-2019

[37] One orthographic word may eventually give multiple words and multiple orthographic words may combine to give a single word.

minimum word heuristics, *uni*, and *bi*-grams based sequence probabilities. In the first two phases, the authors solved the space omission problem (see Section 3.2). In the third phase, the space insertion problem is solved to identify compound words by combining words using different algorithms. The proposed Urdu word tokenizer is trained on 70K words, whereas it is tested on a very small dataset of 2,367 words reporting an overall error rate of 4.2%. Although the authors have reported a very low error rate, this study has some serious limitations: (i) the evaluation is carried out on a very small dataset, which makes the reported results less reliable in terms of how good the word tokenizer will perform on real-world data, (ii) using a statistical *n*-gram technique which may ultimately lead to data sparseness, and (iii) it does not tokenize Urdu text correctly even for short texts.

Another online CLE Urdu word tokenizer is available through a website[38], which allows tokenization of up to 100 words. Its implementation details are not provided. It reports an accuracy of 97.9%. However, the link is not always available, and its API (Application Programming Interface) is not freely available[39]. The CLE online Urdu work tokenizer is applied on three randomly selected input short texts and they all are incorrectly tokenized with many mistakes.

The research cited in [113] takes an approach to Urdu word tokenization, based on the Hindi language. The authors tokenized Urdu words after transliterating them from Hindi, as the Hindi language uses spaces consistently as compared to its Urdu counterpart. They also addressed and resolved the space omission problem for Urdu in two phases. In the first phase, Urdu grammar rules have been applied to decide if the Urdu adjacent words have to be merged or not. If the grammatical rules analyser provides a definite answer that two adjacent words can be joined or not, then no further processing is required. However, if the rule-based analyser is not confident

[38] http://182.180.102.251:8080/segment/ - Last visited: 24-June-2018

[39] http://www.cle.org.pk/clestore/segmentation.htm - Last visited: 24-June-2018

about two words either it can be joined or not, then the second phase is invoked. In the second phase, Urdu and Hindi *uni*-gram and *bi*-gram bilingual lexical resources are used to make the final decision i.e. either it need to join the two adjacent words or not. This technique of Urdu word tokenization used 2.6 million words as training data, whereas, it is tested on 1.8 million tokens. The results show an error rate of 1.44%. The limitations of this study are: (i) the problem of space insertion has not been addressed, (ii) this approach requires large bilingual corpus which is difficult to create particularly for under-resourced languages like Urdu and Hindi.

Rehman et al. (2013) proposed an Urdu word tokenizer by using rule-based (maximum matching) with *n*-gram statistical approach. This approach to Urdu word tokenization uses several different algorithms to solve the problem of space omission and insertion. Firstly, the forward maximum matching algorithm is used to return the list of individual tokens of Urdu text. Secondly, the Dynamic Maximum Matching (DMM) algorithm returns all the possible tokenized sequences of the Urdu text, segments are ranked and the best one is accepted. Thirdly, DMM is combined with the *bi*-gram statistical language model. These three algorithms are used to solve the space omission problem, whereas, for the space insertion problem, six different algorithms are used. The authors used 6,400 tokens for training and 57,000 tokens for testing. This approach has produced up to 95.46% F_1 score. Furthermore, the algorithms are based on probabilities which may result in zero probability being assigned to some unknown words. The authors have not handled such cases with either back off or other smoothing estimators.

Finally, the most recent and another hybrid word tokenization approach segment Urdu tokens using Conditional Random Field (CRF) model which combines orthographic, linguistic and morphological features of Urdu text. For training this approach has used 90K tokens, whereas, tested on 21K tokens. Reported score are F_1

of 97% and 85% for space omission and space insertion (see Section 3.2.1) prediction tasks respectively. This approach has performed well for space omission problem but unable to predict words of space insertion problem and trained and tested on small corpora.

From the above discussion, it can be observed that very few studies have been carried out to address the problem of Urdu word tokenization. Also, these approaches have many limitations which can be summed up as follows: firstly, the developed tools, training and testing datasets, and resources are not always freely and publicly available to develop, compare and evaluate new and existing methods. Secondly, most of the above techniques are based on *n*-gram statistical models which may assign zero probability to unknown words, thus, leading to a data sparseness. Thirdly, word tokenization approaches are tested on small test datasets. Fourthly, the space insertion problem in a few studies has not been tackled. Fifthly, less contextual *uni*/*bi*-grams contextual language models are used. Finally, the two existing Urdu word tokenizers are tested on three short texts and they failed to properly tokenized them.

2.3.6 Sentence Tokenization Approaches

The problem of Urdu sentence tokenization has not been thoroughly explored and only two studies are found [190, 175] which address the issue. Rehman and Anwar [190] used a hybrid approach that works in two stages. First, a *uni*-gram statistical model is trained on annotated data. The trained model is used to identify word boundaries on a test dataset. In the second step, the authors used heuristic rules to identify sentence boundaries. This study achieved up to 99.48% precision, 86.35% recall, 92.45% F_1, and 14% error rate, when trained on 3,928 sentences, however, the authors did not mention any testing data. Although this study reports an acceptable score, it has some limitations; (1) the error rate is high (14%), (2) the evaluation is carried out on a very small dataset, which makes the reported results less reliable and it is difficult to tell how well the sentence tokenizer will perform on real test data, and (3) the trained model along with training/testing data are not publicly available.

In another study, Raj et al. [175] used an Artificial Neural Network (ANN) along with POS tags for sentence tokenization in two stages. In the first phase, a POS tagged dataset is used to calculate the word-tag probability (P) based on the general likelihood ranking. Furthermore, the POS tagged dataset along with probabilities is converted to bipolar descriptor arrays[40], to reduce the error as well as training time. In the next step, these arrays along with frequencies are then used to train feed forward ANN using back propagation algorithm and delta learning rules. The training and testing data used in this study are 2,688 and 1,600 sentences, respectively. The results show 90.15% precision, 97.29% recall and 95.08% F_1-measure with 0.1 threshold values. The limitations of this study are; the evaluation is carried out on a small set of test data, and the trained model, as well as the developed resources, are not publicly available.

2.3.7 Part-Of-Speech Tagging Approaches

Similar to Urdu word (see Section 2.3.5) and sentence tokenization (see Section 2.3.6), the problem of Urdu POS tagging has not been thoroughly explored. Only six studies [84, 16, 17, 205, 139, 225] which addressed the issue are found.

A pioneering piece of research on Urdu POS tagging is described in Hardie [84]. This work focused on the development of a uni-rule POS tagger, which consists of 270 manual crafted rules. The author used a POS tagset with 350 tags [83]. The training data consists of 49K tokens, whereas, testing is carried out on two different

[40] If P $>0.1 \implies$ P$\equiv -1$, If P $== 0.1 \implies$ P$\equiv 0$, If P $<0.1 \implies$ P$\equiv +1$.

datasets containing 42K and 7K tokens. The reported average accuracy for the 42K tokens is 91.66%, whereas, for the 7K corpus the average accuracy is 89.26% with a very high ambiguity level (3.09 tags per word). However, the POS tagset used in this study has several limitations (see Section 3.4), and therefore, cannot be used for a grammatical tagging task, and having a large number of POS tags with a relatively small training data will affect the accuracy, and manually deducing rules is a laborious and expensive task.

The first stochastic POS tagger for the Urdu language is developed in 2007 [17]. They have proposed a POS tagger based on a *bi*-gram Hidden Markov Model (HMM) with back off to *uni*-gram model. Two[41] different POS tagsets are used. The reported average accuracies for the 250 POS tagset and 90 POS tagset are 88.82% and 92.60% respectively. Both are trained on a dataset of 1,000 words, however, the authors have not provided any information about the test dataset. As before, this study has several limitations; the POS tagset of 250 tags has several grammatical deficiencies (see Section 3.4), the information about the proposed tagset of 90 tags is not available, the system is trained and tested on a very small dataset, which shows that it is not feasible for morphological rich and free word order language i.e. Urdu, and used less contextual *bi*/*uni*-gram statistical models.

Anwar et al. [16] have developed an Urdu POS tagger using *bi*-gram HMM. The authors proposed six *bi*-gram Hidden Markov based POS taggers with different smoothing techniques to resolve data sparseness. The accuracy of these six models varies from 90% to 96%. For each model, they used a POS tagset of 90 tags. However, the authors have not mentioned the size of training/test datasets. This study has several limitations as, like the one in [17] the authors have used a 350 POS tagset, which has several misclassifications (see Section 3.4), the training/testing data split

[41] The first POS tagset contains 250 POS tags [83], whereas, the second one consists of 90 tags (details are not given)

is unknown to readers, and limited smoothing estimators have been used, it used *bi*-gram language model (i.e. less contextual), and suffix information has not been explored.

The authors in [205] trained Trigrams-and-Tag (TnT) [38], Tree Tagger (TT) [210], Random Forest (RF) [211] and Support Vector Machine (SVM) [78] POS taggers, using a tagset containing 42 POS tags. All these stochastic Urdu POS taggers are trained on a 100K word dataset, whereas for testing only 9K words are used. The reported accuracy for TnT, TT, RF, and SVM are 93.40%, 93.02%, 93.28% and 94.15% respectively. In terms of limitations, they used a POS tagset of 42 tags which has several grammatical irregularities (see Section 3.4).

In another study [139], stochastic Urdu POS taggers are presented i.e. TnT and TT tagger. These taggers are trained and tested on two different datasets with the following statistics: (i) First dataset consists of 101,428 tokens (4,584 sentences) and, 8,670 tokens (404 sentences) for training and testing respectively, and (ii) the second dataset consists of 102,454 tokens (3,509 sentences) and 21,181 tokens (755 sentences) for training and testing respectively. The reported accuracy for the first dataset is 93.01% for TnT tagger, whereas 93.37% for TT tagger. For the second dataset, TnT tagger produced 88.13% accuracy and TT had 90.49% accuracy. Similar to other studies, it employed a POS tagset which has several grammatical problems (see Section 3.4), meaning that it is no longer practical for Urdu text.

The authors in [225] have proposed Urdu POS tagger[42] which is based on Decision Trees and smoothing technique of Class Equivalence, using a tagset of 35 POS tags. It is trained and tested on the CLE Urdu Digest corpus[43], training and test data split is 80K and 20K tokens, respectively. However, this POS tagger is only available through an online interface, which allows tagging of 100 words. It is trained on a relatively

[42] http://182.180.102.251:8080/tag/ - Last visited: 29-October-2018

[43] http://www.cle.org.pk/clestore/urdudigestcorpus100k.htm - Last visited: 29-October-2018

small dataset that is not freely available. The Decision Tree statistical models are less accurate for Urdu text as compared to HMM etc. [205] (see Section 3.4.7).

From the above discussion, it can be observed that a number of Urdu POS taggers have been proposed and developed. However, similar to Urdu word (see Section 2.3.5) and sentence tokenizers (see Section 2.3.6), the majority of existing Urdu POS taggers along with their training and testing datasets are not publicly available. The other limitations of these studies can be summarised as follows: (i) the employed POS tagsets are either incorrect, or obsolete, which show they will malfunction with statistical models, (ii) rule-based POS taggers are difficult to adopt as they are developed for a particular dataset thus, are not easily generalisable across domains, (iii) less contextual statistical language models have been explored, (iv) other smoothing approaches have not been researched, (v) other features to handle unknown words have not been thoroughly explored, (vi) they have been trained/tested either on small or moderate test datasets.

2.3.8 Datasets

In the related literature, several benchmark datasets have been developed for English and other European languages. For example MEasuring Text Reuse (METER) [52], Microsoft N-Grams [240], British National Corpus (BNC[44]) [112], English gigaword corpus[45] [81], AnCora [189] and Deutschen ReferenzKorpus[46] (DeReKo) [107]. However, since Urdu is an under-resourced language, there has been a lack of standard evaluation resources for it. This section aims to present the Urdu datasets that have been developed in recent years. These datasets are broadly categorise into two main types: (i) raw Urdu datasets and (ii) task specific Urdu datasets.

[44] http://www.natcorp.ox.ac.uk/corpus/index.xml?ID=intro - Last visited: 29-October-2018

[45] https://catalog.ldc.upenn.edu/ldc2003t05 - Last visited: 29-October-2018

[46] The Mannheim German Reference Corpus

For Urdu, three raw datasets have been found in the literature: (i) Becker Riaz dataset [29] (henceforth BR), (ii) Enabling Minority Language Engineering (EMILLE) Lancaster [248] dataset, (iii) The Hans Christensen (HC) dataset[47] [50]. Ten task specific Urdu datasets have been found as: (i) Ali Ijaz dataset (henceforth AIj) [13], (ii) Muaz [139] dataset (henceforth MD), (iii) The CLE Urdu Digest POS Tagged Corpus dataset[48] (henceforth CLE) , (iv) Urdu Monolingual Corpus[49] (UMC) [91], (v) Urdu Paraphrase Plagiarism Corpus[50] (UPPC) [142], (vi) COrpus of Urdu News TExt Reuse[51] (COUNTER) [216], (vii) Urdu Named Entity Recognition dataset (UNER[52]) [103], (viii) URDU.KON-TB treebank dataset [2] (henceforth UKTB), (ix) Urdu Summary Corpus [141] (henceforth USC) and (x) lexical sample (ULS-WSD-18 [204]) and all word sense (UAW-WSD-18 [203]) annotated datasets[53] for WSD task.

In 2002, Becker and Riaz [29] developed the first Urdu dataset (BR). The BR dataset includes documents from Web news articles and collected from British Broadcasting Corporation (BBC). It consists of 7,000 documents with over 50,000 words (tokens). This dataset is further used to carry out NLP research of NER [195] and Information Retrieval (IR) [193]. However, this dataset is no longer available on the Web.

The EMILLE Lancaster corpus [248] is a benchmark dataset for South Asian languages (e.g. Bengali, Gujarati, Hindi, Punjabi, Urdu etc.) created within the EMILLE project. The purpose of these datasets are three fold (i) to build a dataset

[47] http://www.corpora.heliohost.org/ Last visited: 29-October-2018

[48] http://www.cle.org.pk/clestore/urdudigestcorpus100ktagged.htm - Last visited: 29-October-2018

[49] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 29-October-2018

[50] http://ucrel.lancs.ac.uk/textreuse/uppc.php - Last visited: 29-October-2018

[51] https://doi.org/10.17635/lancaster/researchdata/96 - Last visited: 23-February-2019

[52] http://ltrc.iiit.ac.in/ner-ssea- 08/index.cgi?topic=5 - Last visited: 29-October-2018

[53] https://comsatsnlpgroup.wordpress.com/ - Last visited: 20-December-2019

of South Asian Languages, (ii) to extend the GATE[54] language engineering architecture and (iii) to develop basic language engineering tools. The total size of these datasets is 67 million tokens (or words) for different languages. However, the Urdu language dataset is comprised of 300 documents containing 512K and 1,640K tokens of spoken and written Urdu respectively. The dataset is distributed among five genres: Education, Housing, Health, Legal and Social Issues. Based on the EMILLE Lancaster dataset, a parallel English-Urdu dataset of 200K words is also constructed and manually annotated with morpho-syntactic POS tagset [83, 84]. This dataset is available for academic research as well as for commercial use[55].

The HC raw dataset [50] is also a collection of 60 different languages (e.g. Arabic, Chinese, Finish, Spanish, Dutch, Urdu, Welsh etc.). The total size of HC dataset is 1,290 million tokens. However, for Urdu, it consists of approximately 7 million words. Urdu text is collected from three sources including; Blogs, Newspapers, and Twitter. This dataset is distributed among twenty-eight domains: Politics, Environment, Food, Arts & Culture, Crime & Law, International News, Local News, Lifestyle & Fashion, Religion, Business & Economy, Science & Technology, Sport, Entertainment, Weather, Travel, Education, Health, Family, Holidays, Recipes, Home & Garden, Transport, Obituaries, Armed Forces, Emergency & Disaster, Leisure Time, and My Life. In addition, a subset of this dataset (2 million tokens) is used to carry out lexical coverage of newly developed semantic lexicons for Urdu [213]. However, this dataset is no longer available on the web for research purposes.

Ali et al. [13] constructed a large Urdu dataset (AIj) for the text classification task, which contains 26K documents with 19.3 million tokens (234K tokens are unique). The documents in the corpus belong to different genres such as Finance, Culture,

[54] https://gate.ac.uk/ Last visited: 29-October-2018

[55] http://catalog.elra.info/product_info.php?products_id=714 - Last visited: 23-February-2019

Sports, News, Personal and Consumer Information. However, the AI dataset is not publicly available.

Another task dependent dataset [139] MD, is a POS tagged corpus for Urdu language[56]. The total size of the corpus is 110K tokens, which belong to various genres including Politics, Health, Education, International Affairs, Humours, Literature and Business, and collected from different sources. This dataset operates under the commercial licensing options.

The CLE POS tagged dataset[57] is 100K tokens in size, consisting of 348 documents with different genres including Politics, Health, Education, International Affairs, Comedy and Fun, Literature and Business. This is annotated using the Urdu POS tagset proposed in a recent research [225]. A sub part (40K tokens) of the CLE dataset is also used to annotate named entity classes in a shared task in a workshop on NER for South and South East Asian Languages [58]. This dataset is also not publicly available and operates under the commercial licensing options.

The UMC[59] is a Urdu POS annotated dataset [91]. It consists of 4.5K documents, which contain 96.4 million tokens from various genres (News, Religion, Blogs, Literature, Science, Education and numerous others) and sources. The UMC dataset is licensed under Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0[60])

The UPPC[61] dataset [142] is developed for detecting paraphrased plagiarism in Urdu language. Wikipedia articles in Urdu are used to create this corpus. It contains 160 documents with approximately 46K tokens. This corpus is licensed un-

[56] http://www.cle.org.pk/clestore/index.htm - Last visited: 29-October-2018

[57] http://www.cle.org.pk/clestore/urdudigestcorpus100ktagged.htm - Last visited: 29-October-2018

[58] http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5 - Last visited: 29-October-2018

[59] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 29-October-2018

[60] https://creativecommons.org/licenses/by-nc-sa/3.0/ - Last visited: 29-October-2018

[61] https://doi.org/10.17635/lancaster/researchdata/67 - Last visited: 24-February-2019

der a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

The COUNTER dataset [216] is created to measure text reuse in the Urdu language. It contains 1,200 documents, 10,841 sentences and 275,387 tokens. Documents in this corpus belong to following domains: National, Foreign, Business, Sports and Showbiz. This dataset is also released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Another, task specific Urdu dataset [103] is developed for Urdu NER task. This dataset consists of 150 documents, with the following statistics: 48,673 tokens, 1,744 sentences, and 4,621 manually tagged name entities. The dataset is collected from various sources with the following genres: National, Sports, and International. This annotated dataset is publicly available for non-profit research work under the Creative Commons License[62].

A dataset, UKTB is constructed for Urdu semi-semantic POS tagging task. This dataset consists of 1,400 POS annotated sentences. The dataset is collected from various Web resources including BBC[63] and Jang[64] newspapers, 400 sentences are also collected from Urdu Wikipedia[65], and contains data from following genres: Local & International News, Social Stories, Sports, Culture, Finance, History, Religion, Travelling, etc. This dataset is also not publicly available but the authors intend to release it publicly in the near future under the Creative Commons Attribution/Share-Alike License 3.0 or higher license.

Another, task specific dataset, USC [141] is also a task dependent dataset. It is used for the facilitation and evaluation of single-document summarization task. It consists of 50 articles (documents), collected from various online sources, mainly

[62] http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5 - Last visited: 29-October-2018
[63] http://www.bbc.com/urdu - Last visited: 29-October-2018
[64] https://jang.com.pk/ - Last visited: 29-October-2018
[65] https://ur.wikipedia.org/wiki/ - Last visited: 29-October-2018

news and blogs, with following statistics: 29,889 (tokens) in original articles, whereas the summarized dataset just has 11,683 tokens. This dataset has the following genres: News, Current Affairs, Health, Sports, Science and Technology, Tourism, Religion, and Miscellaneous. The Urdu summary annotated dataset is publicly available and operates under the Massachusetts Institute of Technology (MIT) License[66].

Finally, two recently released datasets, ULS-WSD-18 [204] and UAW-WSD-18 [203] for all word and lexical word WSD task respectively are worth mentioning here. The UAW-WSD-18 contains 50 target words constructed manually from a sense inventory dictionary called Urdu Lughat. Furthermore, four baseline WSD approaches were applied to the corpus. Whereas, the UAD-WSD-18 dataset contains 5,042 words of Urdu text. However, only 856 ambiguous words are manually tagged using a Urdu Lughat dictionary. Both datasets contains text of following domains: news, religion, blogs, literature, science, and education and are freely available to the research community to under Creative Commons license[67].

Table 2.3 summarizes all Urdu datasets discussed above. As can be observed by looking at the table these datasets are compiled for assorted Urdu language processing tasks (See Table 2.3) and are to a greater or lesser extent domain specific, task dependent, not always publicly available with some remaining license constraints.

2.4 The UCREL Semantic Analysis System

Directly related to a research presented here is the development of coarse-grained semantic tagging tools, such as USAS [180] and several others cited in [62, 140] and [27]. USAS is different from other WSD systems as it assigns tags from a pre-defined coarse-grained semantic field taxonomy rather than fine-grained word meaning.

[66] https://github.com/humsha/USCorpus/blob/master/LICENSE.txt - Last visited: 29-October-2018

[67] https://creativecommons.org/licenses/by-nc-sa/3.0/ - Last visited: 20-January-2020

Table 2.3 Summary of available Urdu datasets

Name	**AV**	**Year of Release**	**Annotation**	**Documents**	**Tokens**
BR	✗	2002	NER,IR	7K	50K
EMILLE	✗	2004	POS tagging	300	2,152K
HC	✗	2012	–	3	7,714K
AIj	✗	2009	Text classification	26K	19,296K
MD	✗	2012	POS tagging, NER	348	110K
CLE	✗	2014	POS tagging	348	100K
UMC	✓	2014	POS tagging	4.5K	96,400K
UPPC	✓	2016	Plagiarism	160	46K
COUNTER	✓	2016	Plagiarism	1.2K	275K
UNER	✗	2008	NER	150	49K
UKTB	✗	2016	POS tagging	Unknown	1.4K*
USC	✓	2016	SA+	50	29.89K
ULS-WSD-18	✓	2019	WSD	1	5,042
UAW-WSD-18	✓	2018	WSD	1	50

AV:Availability, ✗: No, ✓: Yes, * Sentences, + SA: Summarization Analysis

Furthermore, USAS is also different from LaSIE (named entity identification system) [86], in that it does not just focus on one or two specific classes of words, rather, assigns a tag(s) to every word in a running text. Recently, the systems based on USAS semantic fields have been ported to support fine-grained semantic annotation [166] for historic English text. Moreover, the coarse-grained semantic analysis system has been ported to Finnish [116], Russian [140], and for several other European and world languages using a predefined semantic taxonomy [213, 165, 68]. It is a worthwhile task since if similar semantic tagging tools are design for multiple languages, they can potentially provide a bridge for multilingual Machine Translation and WSD systems. Hereby, this section presents the English semantic tagger and its semantic lexicons which have been used as models when developing the US Tagger. Subsequently, other extensions to the USAS framework will also be briefly introduce, which has now evolved into a multilingual semantic annotation system.

2.4.1 English Semantic Tagger

A semantic tagger which performed automatic semantic analysis of English text is known as an English Semantic Tagger (EST[68]) developed at UCREL, Lancaster University. The EST consists of three main components, (i) semantic tagset, (ii) semantic lexicons, and (iii) software module which assigns semantic tags to each lexical unit. The EST assigns semantic tags on the basis of information retrieved from lexical resources after applying various rules and semantic tag disambiguation algorithms which are the core of the EST. The EST has been successfully used for various studies (see Section 1.1 of Chapter 1). In addition, it has been applied to the following research studies[69]:

1. For the analysis of interview transcripts in market research [247];
2. For the stylistic analysis of written and spoken English [246] in Automatic Content Analysis of Spoken Discourse (ACASD) and Automatic Content Analysis of Market Research Interview Transcripts (ACAMRIT) projects;
3. Used in a pilot study of a large corpus of doctor patient interactions [230];
4. Also EST is utilized in the Requirements Reverse Engineering to Support Business Process Change (REVERE) project [181] in research area of software engineering;
5. In Benedict project[70], where an EST and Finnish semantic tagger have been used together to built a context-sensitive search tool for a new type of intelligent electronic dictionary;

[68] Available through the Wmatrix [179] and on-line interface: http://ucrel.lancs.ac.uk/usas/tagger.html - Last visited: 23-February-2019

[69] A complete list of publications and applications using Wmatrix (in which EST is embedded) can be found at http://ucrel.lancs.ac.uk/usas/andhttp://ucrel.lancs.ac.uk/wmatrix/ - Last visited: 19-October-2018

[70] The project reference is IST-2001-34237. For more information, see ftp://ftp.cordis.europa.eu/pub/ist/docs/ic/benedict-ist-results_en.pdf.

6. Used to create historical thesaurus-based semantic tagger for deep semantic annotation [166];
7. To create a historical semantic tagger for English [12];
8. Analysis of personal weblogs in Singapore English [158];
9. Analysis and standardisation of SMS spelling variation [226];
10. Analysis of the semantic content and persuasive composition of extremist media [172];
11. Detecting gender and spelling differences in Twitter and SMS [26];
12. Discourse analysis [159, 11];
13. Finding contextual translation equivalents for words in the Russian and English languages [217];
14. Key domain analysis [183];
15. Metaphors in political discourse [121];
16. Ontology learning [71];
17. Phraseology [82];
18. Political science research [106];
19. Protection of children from paedophiles in on-line social networks [176];
20. Psychological profiling [127];
21. Sentiment analysis [219];

22. Training chatbots and comparing human-human and human-machine dialogues [218], and

23. Deception detection [127].

2.4.1.1 Semantic Tagset

The categories representing different semantic fields are represented with various codes known as semantic fields or tags, and together these semantic tags form a "semantic tagset". The semantic tagset which USAS framework has adopted is loosely based on the categorization used in the LLOCE [129]. As it offered the most appropriate thesaurus-type classification for sense analysis on which the EST has been developed. Furthermore, the tagset has been expanded and amended based on the critical analysis of several previous iterations which are encountered in the course of the research [180].

The USAS semantic tagset has been classified into 21 top level semantic fields/tags which further expand into 232 sub-fields or tags. With the help of the USAS semantic tagset, everything can be categorised that exists in the universe or can be imagined, whether they are concrete entities or abstract concepts i.e. each field contains words which are related to each other. These words can be antonyms, hyponyms, synonyms, or meronyms, and they represent all parts of speech. Table 2.4 shows the top level 21 semantic tags of the USAS semantic tagset. A list of all top level categories and subcategories is presented in Appendix A in English and in Appendix B its counterpart Urdu. The reader is advised to consult these appendices if semantic tags are not explained or clear from context.

The authors Archer et al. [18] (pp. 1-2) have described that a semantic tag consists of various markers. A top level semantic tag always begins with an upper-case letter which indicates the top level semantic category. This upper-case letter

Table 2.4 USAS top level semantic tags

Domain	Description
A	General and abstract terms
B	The body and the individual
C	Arts and crafts
E	Emotional actions, states and process
F	Food and farming
G	Government and the public domain
H	Architecture, buildings, housing and the home
I	Money and commerce
K	Entertainment, sports and games
L	Life and living things
M	Movement, location, travel and transport
N	Numbers and measurement
O	Substances, materials, objects and equipment
P	Education
Q	Linguistic actions, states and process
S	Social actions, states and processes
T	Time
W	The World and our environment
X	Psychological actions, states and processes
Y	Science and technology
Z	Names and grammatical words

is followed by a digit which indicates the first subdivision in the field. One of the simplest possible semantic tags can contain one upper case letter and one number. For instance, the semantic tag for the word “maudlin” is E1 (Emotional Actions, States and Processes: General) and for word “jasmine” is L3 (Plants). Moreover, if there are more sub-divisions, one or two more numbers can be added (such as, the tag for the adjective “exaltation” is E4.1 (Happy/sad: Happy) and the tag for ”yesterday” is T1.1.1 (Time: General: Past). The research cited in [164] has shown that the depth of semantic hierarchical structure is limited to a maximum of three layers since this has been proven to be the most feasible approach. Theoretically, it would be possible to include as many subdivisions of meaning until no further sub-classification is possible, however, semantic field analysis schemes which are too complex may cause problems for practical semantic analysis. That said, the existing semantic categories can be subdivided for a particular task if need be, since the deep hierarchy structure allows to amend the system easily.

In addition, not all lexical units (words) always fall into only one semantic category but rather are *fuzzy sets*– where one word(s) may belong to more than two predefined semantic categories. These multiple memberships of categories are indicated in the context of the USAS framework by a “slash tag” (also known as a "portmanteau tag"). By way of illustration, “classroom” is tagged P1/H2, since it can be considered to belong both to the category “Education in General” (P1) and to the category “Parts of Buildings” (H2).

Unlike many other present-day semantic taxonomies, the USAS semantic tagset is concept-driven rather than content-driven. This means that it aims at providing a conception of the world that is as general as possible, instead of trying to offer a semantic network for specific domains [164]. If or when it is necessary to have a finer-grained taxonomy for a certain task or purpose, it will be relatively easy to expand the present system simply by adding new levels of subcategories or by using more specific slash tags.

2.4.1.2 Semantic Lexicons

The English semantic lexical resources are the knowledge base for the EST. These lexical resources can be divided into two, (i) single word semantic lexicon, and (ii) multi-word semantic lexicon. The single word semantic lexicon stores the information about single words, whereas, the multi-word semantic lexicon hold the information about the multi-words (e.g. United States of America (proper noun), stub out (verb), drop dead (adverb), etc.). These semantic lexical resources are created manually by first adding semantic tags to the dictionaries of the CLAWS (the Constituent Likelihood Automatic Word-tagging System) POS tagger. Thereafter, these semantic lexicons are expanded by adding words which are collected from large text corpora [164]. These semantic lexicons contain basic and inflected forms as there is no reliable lemmatizer[71] available for the English language when the development of the English semantic tagger took place.

The information about the single word semantic lexicon entries can be found in three columns, (i) the first column indicates the word, (ii) the second column indicates its POS tag[72], and (iii) the third column indicates the semantic category. The simplest scenario occurs when the word has just one sense, in which case a word is given along with its POS tag and with only one semantic tag (have been attached to the lexicon entry) for instance, for the word "accidental" (see Table 2.5), where the word is stored in the first column, POS tag (common noun) in second column and with semantic tag (K2 "Music and Related Activities" stored in third column). However, if the single word is ambiguous (it has more than one sense) the different senses are listed in the third column arranged in frequency order, for example, the word "account" (see Table 2.5) which have two senses a verb and noun sense. Table 2.5 shows some more exemplary words of the English semantic lexicon along with its POS and semantic tags.

The information in the multi-word semantic lexicon is presented in two columns. The first column of the lexicon indicates the Multi-Word Expression (MWE) as well as the relevant grammatical and syntactic information, whereas, the second column includes the semantic category. On the other hand, if the MWE is ambiguous, the semantic tags for the different senses are arranged in frequency order. Likewise they have stored inside single word semantic lexicon. Furthermore, all the multi-

[71] A program which converts input words to its root form.

[72] generated by CLAWS POS tagger. The full CLAWS tagset can be found at http://ucrel.lancs.ac.uk/claws7tags.html.

Table 2.5 Single word semantic lexicon of the English semantic tagger

Word	POS tag*	USAS Semantic tags**
access	NN1	M1 A9 A1.1.1
access	VV0	M1 A9 A1.1.1
accessed	VVN	M1 A9 A1.1.1
accessibility	NN1	M6 A9 S1.2.1 A12
accessible	JJ	S1.2.1 A9 A12 M6
accessing	NN1	M1 A9 A1.1.1
accession	NN1	T2 S7.1 A1.8
accessories	NN2	O2 B5 S8 S2
accessorize	VV0	N5 A2.1 B5 N5 A2.1 H5
accidental	NN1	K2
accidentally	RR	A1.4
accompanying	JJ	S3.1
Accord	NP1	Z3
account	NN1	I1 I1.3 I2.1 Q2.2 Y2
account	VV0	Q2.2
accounts	NN2	I1 I1.3 I2.1 Q2.2 Y2

*CLAWS C7– NN2: plural common noun, VV0: base form of lexical verb, VVN: past participle of lexical verb, NN1: singular common noun, JJ: adjective, NP1: singular proper noun, RR: general adverb, ** for more details see Appendix A

word semantic lexicon entries are written into templates, whereby they consist of patterns of words and grammatical and syntactic information presented in the first column. Often they also contain regular expression symbols or "wild cards" that can represent any character or group of characters. Wild-cards help out to the EST to recognize MWE's which have similar structures. For instance, the template "*_* shortage*_*" would capture the expressions "labour shortage" and "fuel shortages" (see Table 2.6).

The EST recognizes not only continuous MWEs – expressions in which it is not possible to add any embedded elements between the constituents ("dope pusher") but also discontinuous MWEs – expressions inside which it is possible to add varying embedded elements ("double up"). To show it, lets consider the template "doubl*_*

Np/P*/R* up_RP" (see table 2.6) would capture both the expression “double up the reward” as well as “double the price up”. As a consequent, the multi-word semantic lexicon covers many more MWEs than is the number of individual entries. Few more examples of English semantic lexicon are presented in Table 2.6, in which the first column shows wild cards and second column depicts semantic tags of the USAS semantic classification scheme.

Table 2.6 Multi word semantic lexicon of the English semantic tagger

Wild cards	USAS Semantic tags**
dope_NN1 pusher*_NN*	F3 S2
dormer_NN1 bungalow*_NN*	H1
doss*_* R* about_RP	K1
doss*_* R* around_RP	K1
dotted_* R*/Np/PP* about_*	M6
dotted_* R*/Np/PP* around_*	M6
dot_NN1 matrix_NN1	Y2
doubl*_* Np/P*/R* up_RP	N5 A2.1 A6.1 E4.1 S1.1.2
double-decker_JJ sandwich*_NN*	F1
double_* breasted_*	B5
double_* check*_*	X2.4 N6

** for more details see Appendix A

The English semantic tagger has been significantly updated and the researchers have expanded its lexical resources (single and multi-words) over multiple years. Now in the present form, they contain 54,953 single words entries, whereas the multi-word lexicon have 18,921 entries [120] (pp. 98). In addition to this, the resources include a small auto-tagging lexicon i.e. around 50 fixed patterns which can have many possible instantiations. Such expressions can be tagged effectively through the use of wild-cards [180]. For instance, the auto-tagging lexicon entry “*km” (kilometre) would tag all combination of numbers and abbreviation “km” as tagged N3.3 which represents the semantic category “Measurement: Distance”.

2.4.1.3 Semantic Tag Disambiguation Methods

The task of semantic tagging can be broadly divided into two phases, (i) tag assignment and (ii) tag disambiguation. In the first tag assignment phase, all potential semantic tags are attached to each lexical unit/word. In second tag disambiguation phase from already assigned potential tags the contextually appropriate semantic tag is selected. If a word in a running text is included in the semantic lexicons and has only one sense as well as not a part of a MWE, tagging it correctly is a straightforward task for an EST. However, the task of semantic tagging becomes difficult, as it has to recognize that if a word is a single word or a part of a multi-word expression further to this, have to identify its appropriate sense in a given context if a word has multiple senses.

The second phase (tag disambiguation) uses different methods to resolve semantic tag ambiguity. The EST utilizes seven different methods for finding the correct semantic tag for the given sense [180], these are, (i) POS tag, (ii) general likelihood ranking for single word and MWE tags, (iii) overlapping idiom resolution, (iv) domain of discourse, (v) text-based disambiguation, (vi) template rules, and (vii) local probabilistic disambiguation.

2.4.1.3.1 POS Tag The POS tagging is a baseline method that can be used to disambiguate different senses of words (see Section 2.2.4). In the EST it is carried out using CLAWS POS tagger, by way of illustration lets assume a word "match" which have two senses a common noun (Lighter consisting of a thin piece of wood) and a verb sense (a game, a contest). These different senses can be defined in English semantic lexicons as: "match NN1 K5.1 O2 A6.1" and "match VV0 A6.1". If EST determines through CLAWS tagger that the tag NN1 representing (VV0: a base form of the lexical verb) is the relevant POS tag, this simplifies the task of the EST by selecting it with A6.1 semantic tag (representing a semantic category: Comparing: Similar/different).

2.4.1.3.2 General Likelihood Ranking The senses in the English semantic lexicon have been arranged in frequency order according to information obtained from past tagging experience, intuition, and frequency-based dictionaries. The most frequent (most likely) semantic tag is placed first, the second most likely semantic tag is placed second, and so on. Therefore, if a further semantic tag disambiguation method is not applied, it is advisable to use the first semantic tag, because it represents the most likely common sense and thus has a high probability to be a correct tag. To show this let's assume an entry from English semantic lexicon for word "multimillion", i.e. "multimillion JJ N1 I1.3". The tag JJ is used to indicates an adjective which have been assigned by CLAWS POS tagger. The very first semantic tag (N1–Numbers) therefore, first and most common/likely sense is that of a number. The second and least likely sense is the semantic tag, I1.3, represents the semantic category "Money: Price", so here it refers to the dollars, pound etc.

2.4.1.3.3 Overlapping Idiom Resolution In EST multi-word expressions take priority over single word tagging. In other words, EST first matches the text against the multi-word expression templates, and if it finds-out words which match a template and thus together form a MWE, it tags these words together as a unit having the same sense. However, if no suitable multi-word expression is found, a word is assumed to be a single word and therefore, tagged individually. But in a few cases, multi-word expression templates can overlap, in that, some multi-word expression templates can produce more than one set of possible tagging for the same set of words. To resolve this, a set of heuristics have been emerged and embedded into the EST. These heuristics help EST to determine which of the multi-word expression templates is the most likely one and should, therefore, be favoured. These heuristics take account of length and the span of the multi-word expressions and of how much of the template is matched in each case.

2.4.1.3.4 Domain of Discourse If the topic or domain of discourse in an input text is known, this information can be used to "weight" semantic tags or to alter the order of semantic tags in the single and multi-word semantic lexicons for a particular domain. By way of illustration, taking the noun word "java" (java NN1 F2 Z2 Y2) if the domain of discourse in the text dealt with computing, rather than geographical (Z2) or drink (F2), it would be sensible to weight least likelihood semantic category i.e. "Information technology and computing" over the other two most likely senses.

2.4.1.3.5 Text-Based Disambiguation As described in [73] *one sense per discourse*, where a polysemous word appears two or more times in a well-written discourse, it is extremely likely that they will all share the same sense thus, well-written discourses tend to avoid multiple senses of polysemous words. One of their tests is a word "sentence", and the same sense repeatedly appeared both in texts which deal with grammar and in texts which deal with the law. If this hypothesis continued to hold in other cases, it would represent an important addition to the methods for determining word senses. This semantic tag disambiguation approach has not yet been implemented in the EST, however, this approach it resembles the above-mentioned method (domain of discourse) with the exception that, while in a domain of discourse method the weighting is adjusted manually, in this approach the weighting would be determined by the program.

2.4.1.3.6 Template Rules The same type of template rules that are written for the identification of multi-word expressions can also be used for detecting certain senses of words. For example, when the noun “account” occurs in a sequence, such as “someone’s account of something”, it is very likely to mean “narrative explanation” and not “bank account”.

2.4.1.3.7 Local Probabilistic Disambiguation It is generally supposed that the local surrounding context determines the correct semantic tag for a given word. Thus, the surrounding context can be identified in terms of (i) words themselves, (ii) grammatical tags, (iii) semantic tags, or (iv) combination of all three. An application of this method named the “Domain Detection System” [117] is developed in the Benedict project, where the most probable sense of a word is calculated by making use of information about the other words in the same sentence.

2.4.1.4 Software Architecture of the EST and Evaluation

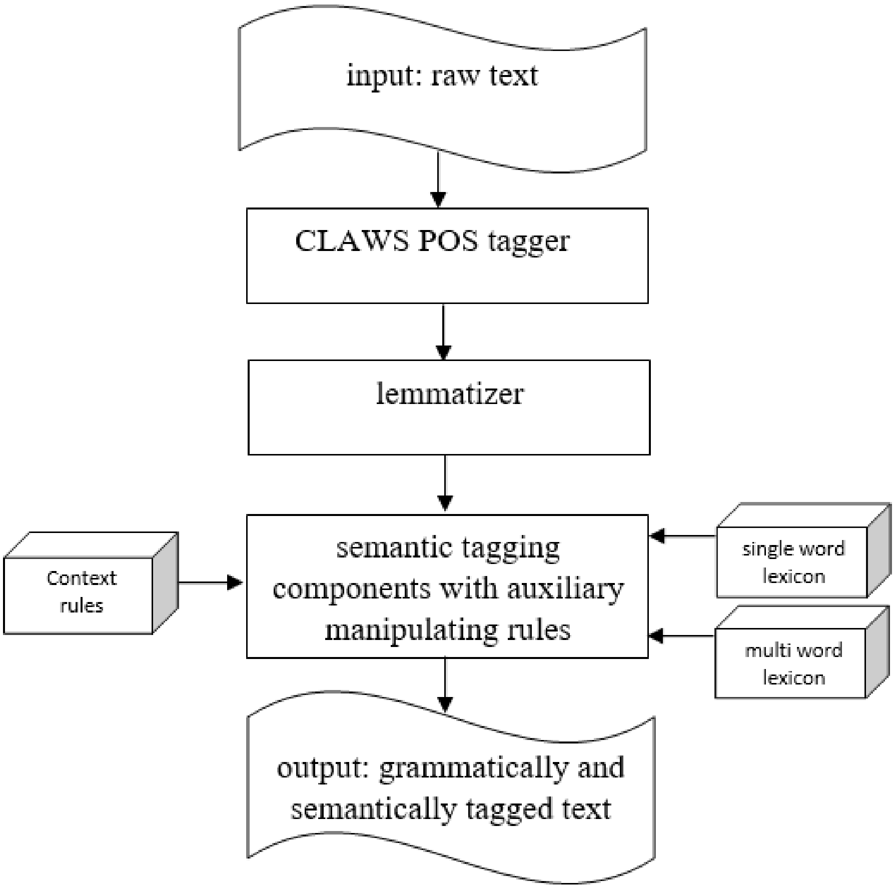
The software architecture of the EST is built on four main components, (i) the CLAWS POS tagger, (ii) the lemmatizer, (iii) semantic tagging component, and (iv) semantic tag disambiguation methods and small auto-tagging lexicon.

The lemmatizer is incorporated into the EST during its original development for the dictionary lookup function. The semantic tagging components are the single word and multi-word semantic lexicons (for more details see Section 2.4.1.2) along with a software module that implements semantic disambiguation methods (see Section 2.4.1.3) which then automatically links words in a text to one or more predefined semantic categories. Figure 2.1 illustrates the multi-level structure of the EST.

The input text is entered into the EST, and the CLAWS POS tagger analyses text grammatically, thus, assigns each lexical-unit (word) with possible CLAWS C7[73] POS tag. In the next preprocessing phase, in order of likelihood, the lemmatizer

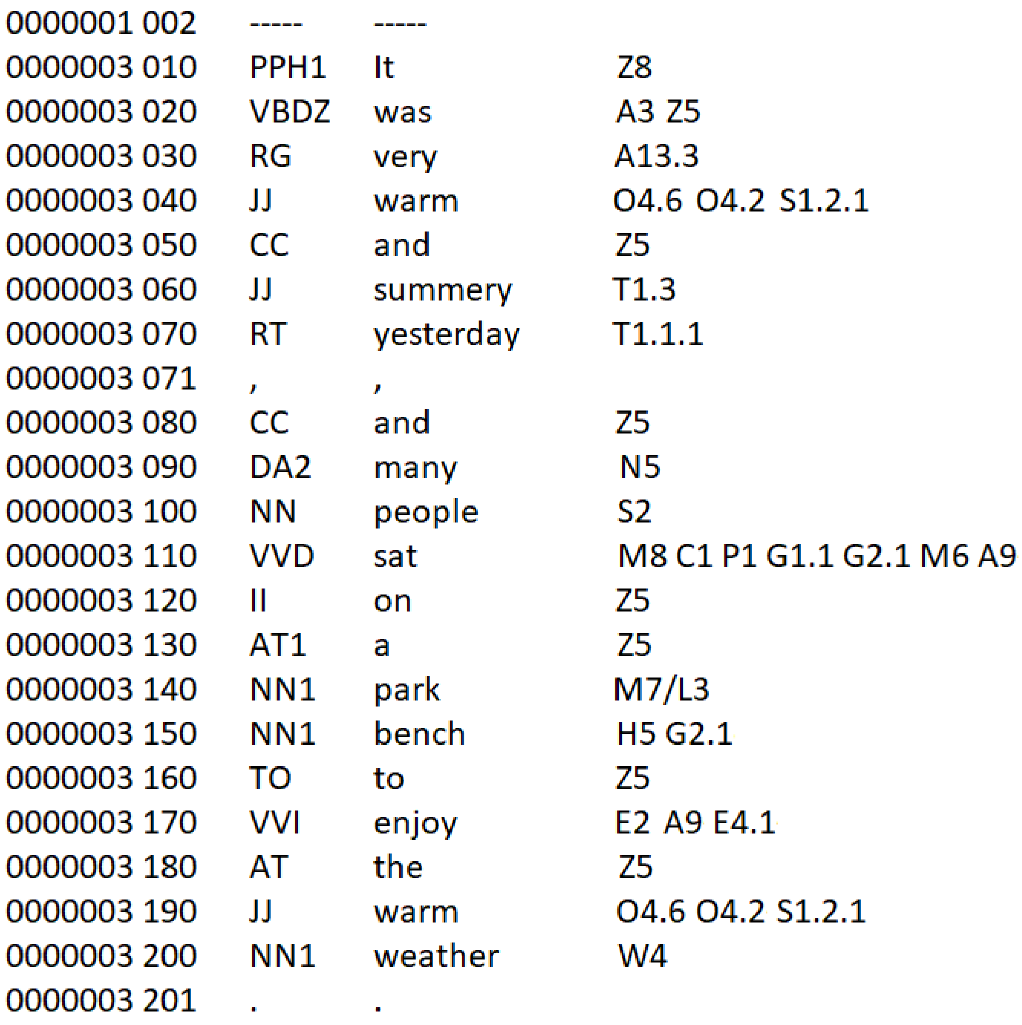
[73] http://ucrel.lancs.ac.uk/claws7tags.html - Last visited: 26-October-2018

Fig. 2.1 Architecture of the English semantic tagger


input: raw text
CLAWS POS tagger
lemmatizer
Context rules
semantic tagging components with auxiliary manipulating rules
single word lexicon
multi word lexicon
output: grammatically and semantically tagged text

finds the basic form of the word. After the lemmatization phase, the semantic tagging components (single and multi-word semantic lexicons along with contextual rules) match the patterns of the output against the patterns in semantic lexicons, utilizing the context rules, and then assigns each word (single or multi-word) with the semantic tag(s) which denotes its meaning. Furthermore, each word or multi-word expression in USAS output may appear with multiple possible semantic field tags to show the different meanings which can be taken in different contexts, and these are left in the output in rough likelihood order if disambiguation methods cannot resolve the correct sense. To illustrate the tagging output of the EST, let's take a sentence "It was very warm and summery yesterday, and many people sat on a park bench to enjoy the warm weather". A tagged output of the EST have been shown in Figure 2.2, where the first part represents the unique IDs for each word, the second column is for CLAWS C7 POS tag, the third column shows the word, and last and fourth column shows semantic tags.

Fig. 2.2 Tagged output of the English semantic tagger



```
0000001 002   -----   -----
0000003 010   PPH1    It           Z8
0000003 020   VBDZ    was          A3 Z5
0000003 030   RG      very         A13.3
0000003 040   JJ      warm         O4.6  O4.2  S1.2.1
0000003 050   CC      and          Z5
0000003 060   JJ      summery      T1.3
0000003 070   RT      yesterday    T1.1.1
0000003 071   ,       ,
0000003 080   CC      and          Z5
0000003 090   DA2     many         N5
0000003 100   NN      people       S2
0000003 110   VVD     sat          M8 C1 P1 G1.1 G2.1 M6 A9
0000003 120   II      on           Z5
0000003 130   AT1     a            Z5
0000003 140   NN1     park         M7/L3
0000003 150   NN1     bench        H5 G2.1
0000003 160   TO      to           Z5
0000003 170   VVI     enjoy        E2  A9 E4.1
0000003 180   AT      the          Z5
0000003 190   JJ      warm         O4.6 O4.2 S1.2.1
0000003 200   NN1     weather      W4
0000003 201   .       .
```

As mentioned earlier, the EST has been tested several times with good results. The latest lexical coverage (see Section 2.5) evaluation of the semantic lexical resources of the EST are carried out in [167]. It shows how many single or multi-words the EST recognizes or how many lexical units (single and multi words) are included in semantic lexical resources which EST can tag. Moreover, tagging results which are reported by Piao et al. (2004) are between 98.49% for the modern English on BNC and for the METER Corpus it shows 95.38%. These results are excellent, which demonstrates that the semantic lexicons (single and multi-word) are able to deal with most words. In addition to this, the EST is also tested on six different historical corpora and its evaluation results ranged between 92.76% to 97.29%. The developers Rayson et al. [180] have evaluated the overall performance of the EST on a corpus of 125K words and reported accuracy (see Section 2.5) as 91.05%, which is outstanding, considering the difficulty of a task to identify different senses. In [168] the authors have used the EST for extracting multi-word expressions on a test corpus, METER, which consists of more than 250K words, and reported accuracy is 90.39%. These all results are comparable to the other existing systems.

2.4.2 Extension of the EST Framework for Other Languages

As with the transformation of the web as a multi-lingual hub and the success of the EST in several research domains (see Section 2.4.1) this encouraged the development of equivalent semantic taggers for other languages. This equivalent semantic tagger enables the development of multi-lingual NLP, HLT, text mining, translation, and other types of information and communication technology systems. In this regard, efforts have been made in the recent past to develop several equivalent semantic taggers for several languages. The first effort to develop non-EST based equivalent semantic tagger is for the Finnish language and carried out by Löfberg (2017) [120], known as a Finnish Semantic Tagger (FST). After the development of the FST, another non-EST semantic tagger has been developed for the Russian language in the Automatic Semantic Assistance for Translators (ASSIST) project [217]. The Russian Semantic Tagger (RST) provides contextual examples of translation equivalents for words from the general lexicon between English and Russian languages [140]. The development of the Finnish and Russian semantic taggers are a relatively similar process, involving the modification of the software framework originally created for English to meet the needs of the analysis of Finnish and Russian languages, respectively. However, these two studies have focused the manual construction of semantic lexical resources (single and multi-word semantic lexicons). These semantic lexicons act as a knowledge base for the semantic tagger, whose creations are indeed a time-consuming, laborious and expensive tasks.

During the last few years, efforts have been reported to create semantic tagger for other languages by using automatic methods to develop semantic lexicons much more rapidly. These methods involve bootstrapping new lexical resources via automatically translating the English semantic lexicons into other languages [165]. This method has proved to be a successful way to create equivalents semantic lexicons for several languages (for more details see Section 2.3.3). Currently, there are twelve non-English semantic taggers available for Arabic, Chinese, Czech, Dutch, French, Italian, Malaya, Portuguese, Spanish, Urdu (described in this thesis) and Welsh languages. The lexical coverage (see Section 2.5) for twelve languages are recently evaluated in [213]. Moreover, there are further plans to extend the EST framework for Turkish, Norwegian, and Swedish languages.

2.5 Evaluation Measures

2.5.1 Evaluation Measures for the US Tagger

To evaluate the results of the US Tagger (see Chapter 5), two main evaluation measures consistent with previous best practice have been used, i.e. lexical coverage, and annotation precision.

Lexical coverage can be defined as the proportion of tokens in the running text that are recognised by semantic annotation system and can be defined as, the total number of words (N) minus the total number of tagged words (W_{tagged}) divided the total number of tagged and untagged words ($W_{untagged}$) (see Equation 2.1).

$$Lexical\ coverage = \frac{N - W_{tagged}}{W_{tagged} + W_{untagged}} \tag{2.1}$$

Precision is defined as the proportion of the correctly tagged words ($W_{correctly\ tagged}$) divided by the total number of tagged words ($N_{tagged\ words}$) (see Equation 2.2).

$$Precision = \frac{W_{correctly\ tagged}}{N_{tagged\ words}} \tag{2.2}$$

2.5.2 Evaluation Measures for Multi-Target Classifiers

The performance of a multi-target classifier (see Chapter 4) can be measured using, (i) Exact Match, (ii) Hamming Loss, and (iii) Accuracy.

Exact match computes the percentage of instances whose predicted set of labels ($\hat{y}$) are exactly the same as their corresponding true set of labels (y), this measure is also known as 0/1 subset or classification accuracy (see Equation 2.3). Where, $\mathbb{I}$ is the indicator function.

$$Exact\ match = \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}(\hat{y}^{(j)} \neq y^{(j)}) \tag{2.3}$$

Hamming loss is used to evaluate how many times, on average, an example-label pair is misclassified (see Equation 2.4). This is a loss function, therefore, lower the value means higher the performance of the classifier.

$$Hamming\ loss = \frac{1}{NL} \sum_{i=1}^{N} \sum_{j=1}^{L} \mathbb{I}[\hat{y}_j^{(i)} \neq y_j^{(i)}] \tag{2.4}$$

Accuracy is the proportion of label values correctly classified of the total number of labels for that instance averaged over all instances (predicted ($\hat{y}$) and true (y)), for a set of N test examples (see Equation 2.5).

$$Accuracy = \frac{1}{N}\sum_{j=1}^{N}\frac{\left|\hat{y}^{(j)} \wedge y^{(j)}\right|}{\left|\hat{y}^{(j)} \vee y^{(j)}\right|} \tag{2.5}$$

2.5.3 Evaluation Measures for the Urdu Natural Language Tools

The approaches applied for Urdu word tokenization (see Section 3.2 of Chapter 3), sentence tokenization (see Section 3.3 of Chapter 3) and POS tagging (see Section 3.4 of Chapter 3) tasks are evaluated using Accuracy, Precision, Recall, F_1 measures, Error Rate, Variance and Standard Deviation.

Accuracy is defined as the proportion of the total number of predictions that are correct (see Equation 2.6).

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{2.6}$$

Where tp[74], tn[75], fp[76] and fn[77] represent true positive, true negative, false positive and false negative respectively.

Precision is defined as the proportion of the predicted positive cases that are correct (see Equation 2.7) .

$$Precision = \frac{tp}{tp+fp} \tag{2.7}$$

Recall is defined as the proportion of positive cases that are correctly identified (see Equation 2.8).

[74] A true positive test result is one that detects the condition when the condition is present.

[75] A true negative test result is one that does not detect the condition when the condition is absent.

[76] A false positive test result is one that detects the condition when the condition is absent.

[77] A false negative test result is one that does not detect the condition when the condition is present.

$$Recall = \frac{tp}{tp + fn} \tag{2.8}$$

The F_1 measure is the harmonic mean of *precision* (P) and *recall* (R), and it is calculated by using the following equation.

$$F_1 = \frac{2 * R * P}{R + P} \tag{2.9}$$

The *error rate* is defined as the ratio between the predicted and actual values (see Equation 2.10).

$$Error\ rate = \frac{fp + fn}{tp + tn + fp + fn} \tag{2.10}$$

In addition, the *standard deviation* (σ) is a measure of variability that represents how far members of a group are spread out from their average value (see Equation 2.11).

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N - 1}} \tag{2.11}$$

Where x_i represent result of the i-th measurement, μ is arithmetic mean of the N results, and N is Number of samples.

2.6 Chapter Summary

In this chapter of related work, it first established the background for this Ph.D. thesis by defining the most important related concepts starting with computational linguistics and then moving on to successively more specialized concepts, natural language processing, text annotation, POS tagging, semantic tagging, semantic fields, and word and sentence tokenization. Semantic tagging is one method of carrying out text annotation, and the necessary pre-processing for semantic tagging is provided by word and sentence tokenization and POS tagging. Thereafter, this chapter briefly discussed corpora and methods for the WSD task as well as some other less related systems for semantic tagging task. Furthermore, this chapter described state-of-the-art lexical resources. After that, an overview of the existing NLP toolkits, Urdu word and sentence tokenization along with POS tagging is presented. Benchmark corpora that have been developed for various Urdu NLP tasks are presented.

Following that, this chapter presented the UCREL semantic analysis system, which this thesis focuses on. The most important undertaking has been the development of the English semantic tagger and its applications to various fields and purposes; they represent the state-of-the-art in the field. In addition to this, key components and the software framework of EST has also been introduced. After this, other extensions to the USAS framework which has now evolved into a multilingual semantic annotation system have been presented. Thereafter, this chapter concluded with a brief account of the measures commonly used to evaluate the performance of semantic tagger, multi-target classifiers and Urdu natural language processing tools, accuracy, precision, recall, F_1, lexical coverage, exact match, hamming loss, and error rate are described.

Chapter 3

Urdu Natural Language Tools

3.1 Introduction

This chapter describes development of Urdu natural language tools. When the problem of semantic tagging is viewed, the primary unit inside Urdu single or multi-word semantic lexicons (see Chapter 5) are words. These single or multi-words are matched and assigned semantic tags (to show different meaning which can be taken in different context) from the Urdu semantic lexicons. However, to select one potential semantic tag from several assigned tags it uses several semantic tag disambiguation methods, for instance, POS tag, where the final selected tag denotes a true or closely related word sense. Therefore, to match words in the Urdu semantic lexicons, the text must be split into sentences, words/tokens, and to resolve semantic tag ambiguity a POS tagger is required. Therefore, the aim in this chapter is to develop Urdu processing tools which are incorporated into the US Tagger (see Chapter 5).

The rest of this chapter has been divided into four parts as follows. The first part (see Section 3.2), second part (Section 3.3), and third part i.e. Section 3.4 explain Urdu word, sentence tokenizers, and POS Taggers respectively along with supporting resources and evaluation results. Finally, the final fourth part (see Section 3.5) presents a chapter summary.

3.2 Urdu Word Tokenizer

This part presents the challenges of Urdu word tokenization, the proposed Urdu word tokenizer, training and testing dataset which is developed to train and evaluate the proposed Urdu word tokenizer, experimental set-up (evaluation measures, results and their analysis).

3.2.1 Challenges of Urdu Word Tokenization

Word tokenization is a challenging and complex task for the Urdu language due to three main problems [66]: (i) the space omission problem - Urdu uses *Nastalique* writing style and *cursive script*, in which Urdu text does not often contain spaces between words, (ii) the space insertion problem - *irregular* use of spaces within two or more words and (iii) ambiguity in defining Urdu words - in some cases Urdu words lead to an ambiguity problem because there is no clear agreement to classify them as a single word or multiple words.

The first two problems stated above, mostly arise due to the nature of Urdu characters, which are divided into: (i) *joiner* (non-separators), and (ii) *non-joiner* (separators). Non-joiner characters,[1] only merge themselves with their preceding character(s). Therefore, there is no need to insert space or Zero Width Non Joiner (ZWNJ; an Urdu character which is used to keep the word separate from their following) if a word ends with such characters. These can form isolated shapes

[1] آ، ا، د، ڈ، ذ، ر، ز، ڑ، ژ، و، ے (Transliteration: alif_mad, alif, daal, ddaal, Zaal, ray, zay, rray, jay, wao, bari_ye). All Urdu characters and word are transliterated as given in [8].

besides final shape, whereas, joiner characters[2] can form all shapes (isolated, initial, medial and final) with respect to their neighbouring letter(s). For instance, the Urdu character خ (khay) is a joiner and has four shapes: (i) isolated خ (khay) e.g. خوخ (KHOKH 'peach') i.e. it can be seen that at the end of a word, if the character is a joiner and its preceding character is non-joiner, it will form an isolated shape, (ii) final ـخ (khay) e.g. مخْ (MKH 'brain'), it can be observed that at the end of a word, if the character is a joiner, it acquires the final shape when leading a joiner, (iii) medial ـخـ (khay) e.g. بخَار (BKHAR 'fever'), in other words, it shows that in the middle of a word, if the character is a joiner, it will form the medial shape when the preceding character is a joiner, (iv) initial خـ (khey) e.g. خوف (KHOF 'fear'), it shows that at the start of a word, if the character is a joiner, it acquires the initial shape when following a non-joiner. Furthermore, the Urdu character ذ (zaal) is a non-joiner, thus has only two shapes: (i) isolated ذ (zaal) e.g. ذاکر ('Zakir'), it can be noticed that at the beginning of a word, if the character is a non-joiner, it acquires isolated shape when following a joiner, (ii) final ـذ (zaal) e.g.لذْیـذْ (LZYZ 'delicious'), it can be examined that at the end of a word, if the character is non-joiner, it acquires final shape when preceding a joiner character. The shapes that these characters (joiner or non-joiners) acquire totally depend upon the context.

A reader can understand a text if a word which ends on a joiner character is separated by a space وہ شہر (OH SHHR, 'that city') or ZWNJ character[3] نئی سائکل ہي (NYY SAYYKL HE, 'is new bicycle'). Likewise, the dropping of either of them (space or ZWNJ) will result in a visual incorrect[4] text, وہشہر (OH SHHR, 'that city') and نئسائکلہي (NYY SAYYKL HE, 'is new bicycle'), thus being perceived as a single word

[2] ب، پ، ت، ٹ، ث، ج، چ، ح، خ، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ک، گ، ل، م، ن، ۂ، ء، ه، ی" (Transliteration: bay, pay, tay, ttay, say, jeem, chay, hay, khay, seen, sheen, suad, zuaad, tuay, zuay, ain, ghain, fay, qaaf, kaaf, laam, meem, noon, hay_gol, hamza, hey_dochasmi, chooti-ye) for such characters, it is needed to insert a space between words or ZWNJ at the end of the first word, otherwise it will join itself with the following word.

[3] Non-printing character (U+200C) is used for computer writing systems.

[4] Human readable but words that are merged into a single token.

even though they are two and three different words, respectively. On the other hand, a word which ends on a non-joiner character does not merge itself with other words, for instance, کمپوٹر انٹرنٹ (KMPYOTR ANTRNYT, 'computer internet') and مدد کرو (MDDKRO, 'help him'), even if we remove space or ZWNJ character. Note that the کمپوٹرانٹرنٹ (KMPYOTR ANTRNYT, 'computer internet') and مددکرو (MDDKRO, 'do help') are also incorrect text, each of them is a combination of two words. As, مددکرو (MDDKRO, 'do help') is مدد (MDD, 'help') and کرو (KRO, 'do'), whereas, کمپوٹرانٹرنٹ (KMPYOTR ANTRNYT, 'computer internet') have کمپوٹر (KMPYOTR, 'computer') and انٹرنٹ (ANTRNYT, 'internet') words. However, omitted space(s) between all ambiguous text results in a space omission problem, which can be overcome by inserting a space at the end of the first word so that two or three distinct words can be detected. For example, نئسائکلهي (NYY SAYYKL HE, 'is new bicycle) are three distinct words, written without spaces, in order to tokenize them properly we need to insert spaces at the end of نئی (NYY, 'new') , and سائکل (SAYYKL 'bicycle') so that three different tokens can be generated: (i) نئی (NYY, 'new'), (ii) سائکل (SAYYKL, 'bicycle'), and (iii) ہي (HY, 'is'). As can be noted from the above discussion, space omission problems are complex thus making the Urdu word tokenization task particularly challenging.

In the space insertion problem, if the first word ends either on a joiner or non-joiner, a space at the end of the first word (see Table 3.1, Correct column– incorrect multiple tokens with space (-), but correct shape) can be inserted for several reasons: (i) affixes can be separated from their root, (ii) to keep separate Urdu abbreviations when transliterated, (iii) increase readability for Urdu proper nouns and English/foreign words are transliterated, (iv) compound words and reduplication morphemes do not visually merge and form a correct shape and (v) to avoid making words written incorrectly or from combining (see Table 3.1, incorrect column– single token but incorrect shape). For example, خوش اخلاق (KHOSH AKHLAK, 'polite') is a compound word of type affixation, however, space was inserted between خوش (KHOSH, 'happy') i.e. a prefix (literally 'happy') and اخلاق (AKHLAK, 'ethical') i.e. root to increase the readability and understandability. To identify خوش اخلاق (KHOSH AKHLAK, 'polite') as a single word/token the tokenizer will need to ignore the space between them. This also serves to emphasise the fact that the space insertion problem is also a very challenging and complex task in Urdu word tokenization.

Table 3.1 Example text for various types of space omission problems

Type	**Correct**	**Incorrect**	**Translation**
Affixation	خوش اخلاق KHOSH AKHLAK	خوشاخلاق KHOSHAKHLAK	Polite
Abbreviations	این ایل ای AYN AYL AY	اینایلای AYNAYLAY	NLE
Compound word	تغیر پذیر TGHYR PZYR	تغیرپذیر TGHYR PZYR	Variable
English word	نیٹ ورک NYT ORK	نیٹورک NYTORK	Network
Proper noun	ویسٹ انڈیز OYST ANDYZ	ویسٹانڈیز OYSTANDYZ	West Indies
Reduplication	آنن فانن AANN FANN	آننفانن AANNFANN	Quickly

As discussed earlier, in some cases Urdu words are harder to disambiguate. There is no clear agreement on word boundaries in a few cases (sometimes they are considered as a single word even by a native speaker). For example the compound word, وزیر اعلی (OZYR AALY, 'chief minister'), بہن بھائی (BHN BHAYY, 'sibling', literally 'brother sister'). The same is the case for reduplications, فر فر (FR FR, 'fluent') and affixation, بد اخلاق (BD AKLAK, 'depravedly'). Certain function words (normally case markers, postpositions, and auxiliaries) can be written jointly e.g.

اسمیں (ASMYN, 'herein'), یہوقت (YHOKT, 'this time') or ہوگی (Ho GEE). Alternatively, the same function words can be written separately such as اس میں (AS MYN, 'herein'), یہ وقت (YH OKT, 'this time') and ہو گی (HO GEE) (i.e two auxiliaries) respectively. These distinct forms of the same word(s) are visually correct and may be perceived as single or multiple words. These types of cases are ambiguous i.e. can be written with or without spaces and can be treated as a single unit or two different words. Consequently, this changes the perception of where the word boundary should sit. A possible solution to handle such words is to use a knowledge base.

To conclude, the space insertion problem, space omission problem and ambiguity in tokenizing multi-words makes the Urdu word boundary detection a complex and challenging task. This may be a possible explanation for the fact that no standard efficient Urdu word tokenizer is publicly available. An efficient Urdu word tokenization system would be needed to deal with these issues and to properly tokenize Urdu text.

3.2.2 Pilot Study to Find Out Word Tokenization Issues of Urdu Text:

To analyse and understand the significance of space (omission and insertion) related problems in Urdu word tokenization, a pilot study is further conducted. Whose primary aim is to explore the challenges that arise in Urdu word tokenization due to the irregular use of spaces in writing Urdu text [87].

For this analysis, a subset (6K tokens) of UMC dataset (see Section 2.3.8) is used. Note that this analysis is carried out manually to identify space related problems in Urdu text. The UMC dataset contains Urdu text, which is collected from various sources (BBC Urdu News[5], Express News[6], Urdu Library[7], Minhaj Library[8], Awaz-e-Dost[9], Wikipedia[10]).

Table 3.2 presents the statistics of space related problems in Urdu text. It can be observed from the table that a high percentage of space omission (378) and space insertion errors (1,303) (see Section 3.2.1) are found in Urdu text (total errors are 1,681 which is 33.62% of text). It has been observed that the joiner characters (256) (see Section 5.1) are the most significant causes for space omission errors, whereas in space insertion, the most common error causes are affixation (176), MWE (277), English words (220), and proper noun (386). As far as the reduplication is concerned, it is the less observed phenomenon (101) in the subset of UMC dataset used in this study.

Table 3.2 Statistics of space related problems in the subset of UMC dataset

Problem	**BBC**	**Express**	**Library**	**Minhaj**	**Awaz**	**Wiki**	**Total**
			Space Omission				
Joiner	102	45	23	42	21	23	256
Non-Joiner	22	28	15	23	19	15	122
			Space Insertion				
Affixation	46	34	29	28	20	19	176
Abbreviations	29	38	17	14	15	30	143
MWE	59	68	39	52	31	28	277
English word	36	24	36	15	07	102	220
Proper noun	70	92	53	67	49	55	386
Reduplication	20	24	19	21	11	06	101
Total	384	353	231	262	173	278	1,681

[5] http://www.bbc.com/urdu - Last visited: 14-November-2018

[6] http://www.express.pk/ - Last visited: 14-November-2018

[7] http://www.urdulibrary.org/ - Last visited: 14-November-2018

[8] http://www.minhajbooks.com/urdu/control/ - Last visited: 14-November-2018

[9] http://awaz-e-dost.blogspot.co.uk/ - Last visited: 14-November-2018

[10] https://ur.wikipedia.org/wiki/ - Last visited: 14-November-2018

To summarize, both space omission and space insertion are serious and common problems in Urdu text. An efficient Urdu word tokenization system would be needed to deal with space related issues and to properly tokenize Urdu text.

3.2.3 Generating Supporting Resources for the Urdu Word Tokenizer

For the proposed Urdu word tokenizer, two dictionaries are developed: (i) a complex words dictionary - to address space insertion problem and (ii) a morpheme dictionary - to address the problem of space omission.

3.2.3.1 Complex Words Dictionary

To address the space insertion problem, a large complex words dictionary was created using the UMC Urdu dataset [91], which contains data from various domains including Sports, Politics, Blogs, Education, Literature, Entertainment, Science, Technology, Commerce, Health, Law, Business, Showbiz, Fiction and Weather. From each domain, at least 1,000 sentences were randomly selected and pre-processed to remove noise (see Section 3.2.6). After noise removal, to speed up the dictionary creation process a basic space-based tokenization approach was implemented in Java to split sentences into words. Space based tokenization resulted in some incorrect word generation, e.g., complex words such as the prefix ان گنت (AN GNT'countless') is incorrectly split into a morpheme, ان (AN, literally 'this') and a stem, گنت (GNT, literally 'count'), postfix حملہ آور (HMLH AAOR, 'assailant') is incorrectly split as حملہ (HMLH, 'attack') i.e. a root and آور (AAOR, literally 'hour') i.e. a morpheme.

Complex words which can be categorised into three types with respect to their formation: (i) *AB formation*– two roots and stems join together, (ii) *A-o-B formation*– two stems or roots are linked to each other with the help of و (wao) (a linking morpheme), and (iii) *A-e-B formation*– 'e' is the linking morpheme which shows relation between A and B. (for more detailed discussion see [191]). In this research all three types have been used without any classification e.g. A-o-B formation type of compound word غور و فکر (GHOR O FKR, 'contemplation') is incorrectly split as غور (GHOR, literally 'ponder') a root, و (O) a linking morpheme, and a stem فکر (FKR, literally 'worry'). Reduplication which have two types: (i) *full reduplicated word*– two duplicate words are used to form a word and (ii) *echo reduplication*– the onset of the content word is replaced with another consonant (detailed information can be found in [33]). Echo reduplication word, دن بدن (DN BDN, 'day by day') is incorrectly split as دن (DN, literally 'day') i.e. content word and بدن (BDN, literally 'body'), a consonant. One million space-based tokenized words list (henceforth UMC-Words) has been used to form a large complex words dictionary containing: (i) affixes, (ii) reduplications, (iii) proper nouns, (iv) English words, and (v) compound words.

To collect affixes (prefixes and postfixes) complex words from the UMC-Words list [91], a two-step approach is used. In the first step, a list of prefixes and postfixes are manually generated.

In the second step, an automatic routine is used to extract words containing affixes from the large UMC-Words list. Using prefixes and postfixes, the previous and next words are extracted respectively from the UMC-Words list.

Reduplications complex words are collected using two methods: (i) full extraction and (ii) partial extraction. The full extraction method is used to extract the full reduplicated words such as جسي جسي (JYSY JYSY, 'as'). To extract such full reduplicated words, we compared each word in the UMC-Words list to the next word, if both are the same then concatenate both to form a full reduplicated compound word. The partial extraction method is used to collect the words of echo reduplication i.e.

in which a consonant word is a single edit distance away from the first content word. The echo reduplication words can be further collected using two methods: (i) one insertion extraction and (ii) single substitution extraction.

One insertion extraction method extracts the one insertion reduplicated words, in which the consonant word has one insertion in its content word e.g. دن بدن (DN BDN, 'day by day'). It can be noted that the consonant word بدن (BDN, literally meaning 'body') has one more character (three) as compared to the content word دن (DN, literally 'day') (which have two characters). Furthermore, the last two characters of the consonant word are identical to the content word. To extract one insertion reduplicated words, we used the UMC-Words list. The extraction process works as follows: after excluding the first character, if the remaining characters of consonant word are identical as well as having the same character count to the content word, they are one insertion reduplicated word(s) we concatenated them to form a single word.

The single substitution extraction method extracts the single substituted reduplicated word(s) - here the consonant word has single substitution in its content word e.g. خلط ملط (KHLT MLT, 'intermixed'). It is worth noting that both words content خلط (KHLT, literally 'bad') and consonant ملط (transliteration: MLT) has three characters and the final two characters are overlapping. To extract one substituted reduplicated word(s) we used automatic routine and applied the following process over the UMC-Words list as: if the length of the content word is matched with the length of the consonant word and the length of content word is greater than two[11] characters, and if one character is dissimilar after comparing character by character, then it will form a single substitution reduplicated complex word.

[11] To make sure the two character words or auxiliaries could not be erroneously identified as reduplication such as کر کے (KR KE, literally 'by doing')

To automatically extract abbreviations (91) and proper nouns (2K), regular expressions are used and further supplemented by manual checking to increase the size of the proper nouns (3K) and abbreviations lists (187). The remaining 65K proper noun list was generated in another NLP project and are used in this study for Urdu word tokenization. In addition to this, manual work[12] was also carried out to remove noisy affix entries. Moreover, compound words (of formation AB and A-e-B) and English words are added to increase the size of the complex words dictionary. However, to collect words of A-o-B formation automatically, a linking morpheme (و, O) has been used. While using a linking morpheme both previous and next words are extracted from the UMC-Words list to form a A-o-B compound words.

The complete statistics of the compound word dictionary are shown in Table 3.3. There are in total 80,278 compound words (7,820 are affixes, 278 are abbreviations, 10,000 are MWEs, 1,480 are English words, 60,000 are proper nouns and 700 are reduplication words).

Table 3.3 Statistics of compound words dictionary

Class	**Tokens**
Affixation	7,820
Abbreviations	278
MWE's	10,000
English words	1,480
Proper nouns	60,000
Reduplication	700
Total	80,278

[12] Five undergraduate NLP students have been employed to carry out manual tasks, all are native Urdu speakers and have an interest in Urdu NLP and literature. Furthermore, each student undertook a practical training session on annotation tasks. Each student was given an annotation assignment of 80 random sentences from the UPC dataset and requested to extract affixes, compound words, abbreviations and English words. These assignments were marked and each student was awarded with a score. Students having scored 85% or above were thus selected for annotation tasks.

3.2.3.2 Morphemes Segmentation Process

To address the space omission problem (see Section 3.2.4), a large-scale morpheme dictionary is automatically compiled from the HC dataset [50]. Before we proceed further towards the approach used to generate the morphemes dictionary, it is worth describing the morpheme types. Urdu language morphemes can be categorized into: (i) free and (ii) bound morphemes. Proposed word tokenizer has to rely on both categories. The bound or functional morphemes such as affixes include prefixes, e.g., "کو ، لا ، گا" (GA, LA, KO), linking morphemes, for e.g., و ، ا (A, O) or suffixes, e.g., زدہ ، شدہ (transliteration: SHDH, ZDH), can only expose their meanings if they are attached to other words, i.e. they cannot stand alone. Whereas, free or lexical morphemes can stand alone, for example, مقبول ، چست ،علم ، غم (MKBOL, CHST, ALM, GHM, 'grief, knowledge, clever, famous').

There are two further categories of free morphemes: (i) true free morphemes and (ii) pseudo-free morphemes. True free morphemes can be either standalone (for e.g., دل (DL, 'heart')) or form part of other words (e.g. درد دل (DRD DL, 'angina pectoris')). Pseudo-free morphemes can be a character, affix or word.

The preceding discussion summarizes the various types of morphemes. However, from a computational linguistics view, free and bound morphemes play a vital role in Urdu word formations [102], hence, they will be used without any further classification in our proposed UNLT word tokenizer module.

In order to generate the morpheme dictionary, the 1,000 most frequent words of the HC dataset are used [50]; the selected words were split to form a morpheme dictionary. The whole chopping process is completed in two steps: (i) Crude-Morphemes (CM) chopping and (ii) Ultra-Crude-Morphemes (UCM) chopping.

In the first step, the first *n* character(s) of each word are kept while the rest are discarded. For example, in case of $n = 1$, we kept only the first character and discarded all others, thus words such as واقفت (OAKFYT, 'awareness') will return و (wao). Such single character morphemes are helpful to formulate A-o-B formation type of complex words, for instance خوش و خرم (KHSH O KHRM, 'canty'). Furthermore, we keep chopping all the words repeatedly with the following values of $n = 2, 3.4, 5, 6$[13]. This process returns وا، واق، واقف، واقفی، واقفت (transliterations are: OA, OAK, OAKF, OAKFY, OAKFYT) morphemes for the word واقفت (OAKFYT, 'awareness'). There may be a situation where we may lose several valuable morpheme(s), if the length of $n > 6$. Nevertheless, this is a rare case. Henceforth, we will call this method Crude-Morpheme chopping.

To generate entirely different morphemes from the same word, we further applied a modified version of CM chopping, i.e. UCM. In which, we skipped the first character and then applied the CM chopping with length $n = 2, 3, 4$. Thus, UCM chopping resulted with these morphemes, اق، اقف، اقفی، اقفت (transliterations are: AK, AKF, AKFY, AKFYT) for the word واقفت (OAKFYT, 'awareness'). Furthermore, we iterate the UCM chopping method by skipping the first two characters (as well as three, four etc.), until we meet the last two characters. Thus, the following morphemes are returned by UCM, in the third قف، قفی، قفت (transliterations are: KF, KFY, KFYT), in the fourth فی، فت (transliterations are: FY, FYT) and in the last یت (transliteration, YT) iterations.

Repeating CM and UCM chopping on the entire list of words will return all possible morphemes. The two chopping methods used in this study will result in erroneous morphemes. However, we manually examined the morpheme dictionary and removed these. The number of morphemes generated by the CM and UCM chopping methods were 5,089 and 7,376 respectively.

[13] An assumption made by us after analysing Urdu text that a word is formed of a maximum of six morphemes

It can be observed from the above discussion that two different large-scale dictionaries i.e. the complex words dictionary and the morphemes dictionary are generated with distinct approaches and with various statistics. These dictionaries will be used to solve the space omission and space insertion problems with the word tokenizer module. To the best of our knowledge, no such large complex words (a study [85] just proposed a scheme to extract location and person name) and morpheme dictionaries have been previously compiled semi-automatically for Urdu, to perform Urdu word tokenization task.

3.2.4 Proposed Urdu Word Tokenizer

To investigate an effective approach for Urdu Word Tokenization (henceforth UNLTool-WT approach), the proposed Algorithm 1 is a combination of state-of-the-art approaches: rule-based maximum matching, dictionary lookup, statistical *tri*-gram Maximum Likelihood Estimation (MLE) with backed-off to *bi*-gram MLE. Furthermore, smoothing is applied to avoid data sparseness. A step by step working example of the proposed algorithm can be seen in Section 3.2.4.1.

3.2.4.1 Processing Steps with an Example

This section will present a step by step worked example of the proposed Algorithm 1.

——— First Iteration ———

1. Initialize flag_bit=false, row=1, column=1, word_counter=0;

2. Create array words_list[row][column], array morphemes_list, array compound_words_list, array input_text;

3. Remove all white spaces and ZWNJ, to form a space free input text.

Algorithm 1 UNLTool-WT approach

```
Step 1: Initialize flag_bit=false, row=1, column=1, word_counter=0;
Step 2: Create array words_list[row][column], array morphemes_list, array compound_words_list;
Step 3: Remove all white spaces and ZWNJ, to form a space free input text.
Step 4: Read bi-gram of input text.
Step 5: Match this bi-gram with each word of morphemes_list
Step 6: Extract all those morphemes from morphemes_list, which matched with bi-gram.
Step 7: Store each extracted morpheme on a separate row/column of words_list
Step 7.1: For each row, copy the flag_bit, word_counter++
Step 8: If no match is found in morphemes_list, split bi-gram into uni-gram.
Step 8.1: Store the first uni-gram with previous morpheme (column) except و (character O) and ا (character A) (use in compound words) and turn the flag_bit=true. For و and ا, store it on separate column of arraywords_list[row][column] and increment word_counter.
Step 9: Repeat the steps 4 to 8, until sentence ending marker, and for each row of words_list.
Step 10: Select the row having minimum word_counter value and flag_ bit=false.
Step 11: If multiple rows are qualified in step 10 then
Step 11.1: Calculate tri-gram MLE for each row.
Step 11.1.1: Select the one having highest value of tri-gram MLE.
Step 11.2: If in step 11.1.1, any row having tri-gram MLE value equal to Zero, then calculate bi-gram MLE for each row.
Step 11.2.1: Select the one having highest value of bi-gram MLE.
Step 11.3: If in step 11.2.1, any row having bi-gram MLE value equal to Zero then, calculate bi-gram smoothing for each row.
Step 11.3.1: Select the one having highest value of smoothing.
Step 12: For final selected row, read each column and match in the compound word dictionary.
Step 12.1: If a match is found then read the next column of selected row in step 12 and repeat step 12 for the remaining part of selected row.
Step 12.1.1: If complete match is found then concatenate with the columns in step 12.1.
Step 12.1.2: Move each element of final selected row in step 12, decrease the array index.
Step 13: Finally, the list of tokenized words will be produced.
```

- input_text: وہ سعودی عرب گیا ۔ (OH SAODY ARB GYA, 'He went to Saudi Arabia').

- input_text: after removing white spaces and ZWNJ: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia').

4. Read first *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: وہ (OH)

5. Match this (وہ) *bi*-gram with each word of *morphemes_list*.

   - morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

6. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. وہ.

7. Store each of the extracted morpheme on a separate row/column of *words_list*. i.e. words_list[1][1] (see Table 3.4, Row (i) and C1)

   7.1. For each row, copy the flag_bit, word_counter++, i.e. flag_bit=false and word_counter=1;

8. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Second Iteration ———

1. Read next *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: سع (SA)

2. Match this (سع) *bi*-gram with each word of *morphemes_list.*

   - morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA))

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. سع.

4. Store each of the extracted morphemes (سع، سعود، سعودی (transliteration: SA, SAOD, SAODY)) on a separate row/column of *words_list* and with previous columns of *words_list*[1][1].

   - store (سع) on words_list[1][2] (see Table 3.4, Row (i) and C2)
   - store (سعود) on words_list[2][2] (see Table 3.4, Row (ii) and C2)
   - store (سعودی) on words_list[3][2] (see Table 3.4, Row (iii) and C2)

   4.1. For each row, copy the flag_bit, word_counter++, i.e. flag_bit= false and word_counter=2;

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Third Iteration ———

1. Read next *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: ود (OD)

2. Match this (ود) *bi*-gram with each word of *morphemes_list.*

- morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. ود.

4. If no match is found in *morphemes_list*, split *bi*-gram into *uni*-gram.

   4.1. Store the first *uni*-gram with previous morpheme (column) except و (O) and turn the flag_bit= true. Otherwise, store it on separate column of *words_list* and increment word_counter.

   - store first *uni*-gram i.e. و on a separate column of *words_list*[1][3] (see table 3.4, Row(i) and C3).
   - word_counter++ i.e. word_counter= 3;
   - concatenate remaining د (D) with the next *uni*-gram of input_text i.e. ی (Y)

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Fourth Iteration ———

1. Read next *uni*-gram of input text i.e. ی (Y) and concatenate with د (D)

   - input_text: وہسعودیعربگا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: دی (DY)

2. Match this (دی) *bi*-gram with each word of *morphemes_list.*

- morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. دی.

4. Store extracted morpheme on a separate column of *words_list*. i.e. words_list[1][4] (see Table 3.4, Row(i) and C4)

   4.1. word_counter++, i.e. word_counter= 4;

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker, and for each row of *words_list* i.e. the condition is *false*.

——— Fifth Iteration ———

1. Read next *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: عر (AR)

2. Match this (عر) *bi*-gram with each word of *morphemes_list.*

   - morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. عر.

4. Store each of the extracted morphemes (عر، عرب (AR, ARB)) on a separate row/column of *words_list* and with previous columns of *words_list*[1].

- store (عر) on words_list[1][5] (see Table 3.4, Row (i) and C5)
- store (عرب) on words_list[4][5] (see Table 3.4, Row (ii) and C2)

4.1. For each new row, copy the flag_bit, word_counter++ of *words_list*[1], i.e. flag_bit= false and word_counter=5;

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Sixth Iteration ———

1. Read next *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, ‘HewenttoSaudiArabia’) →*bi*-gram: بگ (BG)

2. Match this (بگ) *bi*-gram with each word of *morphemes_list.*

   - morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. بگ.

4. If no match is found in *morphemes_list*, split *bi*-gram into *uni*-gram.

4.1. Store the first *uni*-gram (ب (B)) with previous column of *words_list*[1][5] (see table 3.4, Row(i) and C5).

   - concatenate remaining گ (G) with the next *uni*-gram of input_text.
   - set flag_bit= true;

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Seventh Iteration ———

1. Read next *uni*-gram of input text i.e. ی (Y) and concatenate with گ (G)

    - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, 'HewenttoSaudiArabia') →*bi*-gram: گی (GY)

2. Match this (گی) *bi*-gram with each word of *morphemes_list.*

    - morphemes_list: وہ، سع، سعود، سعودی، دی، عر، عرب، رب، گی، گیا (transliteration: OH, SA, SAOD, SAODY, OY, AR, ARB, RB, GY, GYA)

3. Extract all those morphemes from *morphemes_list*, which matched with the *bi*-gram. i.e. گی (GY).

4. Store each of the extracted morphemes (گی، گیا (GY, GYA)) on a separate row/column of *words_list* and with previous columns of *words_list*[1].

    - store (گی) on words_list[1][6] (see Table 3.4, Row (i) and C6)
    - store (گیا) on words_list[5][6] (see Table 3.4, Row (v) and C6)

    4.1. For each new row, copy the flag_bit, word_counter++ of *words_list*[1], i.e. flag_bit= true and word_counter=6;

5. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker i.e. the condition is *false*.

——— Eighth Iteration ———

Table 3.4 Tokenized words using the UNLTool-WT approach

Index	**Description**						**Flag**	**Tokens**
	C1	C2	C3	C4	C5	C6		
Row (i)	وہ	سع	و	دی	**عر ب	*گی ا	1	6
Transliteration	OH	SA	O	DY	AR B	GY A		
Translation	‘He’	–	–	–	‘Arab’	‘Went’		
Row (ii)	وہ	#سعود ی	**عر ب	*گی ا			1	4
	OH	SAOD Y	AR B	GY A				
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (iii)	وہ	سعودی	**عر ب	*گی ا			1	4
	OH	SAODY	AR B	GY A				
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (iv)	وہ	سع	و	دی	عرب	*گی ا	1	6
	OH	SA	O	DY	ARB	GY A		
	‘He’	‘Sa’	‘Wao’	‘De’	‘Arab’	‘Went’		
Row (v)	وہ	سع	و	دی	عرب	گیا	1	6
	OH	SA	O	DY	ARB	GYA		
	‘He’	‘Sa’	‘Wao’	‘De’	‘Arab’	‘Went’		
Row (vi)	وہ	سعودی	عرب	*گی ا			1	4
	OH	SAODY	ARB	GY A				
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (vii)	وہ	سعودی	عرب	گیا			1	4
	OH	SAODY	ARB	GYA				
	’He’	‘Saudi’	‘Arab’	‘Went’				
Row (viii)	وہ	سعودی	عرب	*گی ا			1	4
	OH	SAODY	ARB	GY A				
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (ix)	وہ	سعودی	عرب	گیا			1	4
	OH	SAODY	ARB	GYA				
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (x)	وہ	سع	و	دی	عرب	گیا	0	6
	OH	SA	O	DY	AR B	GY A		
	‘He’	‘Sa’	‘Wao’	‘De’	‘Arab’	‘Went’		
Row (xi)	وہ	سعودی	عرب	گیا			1	4
	‘He’	‘Saudi’	‘Arab’	‘Went’				
Row (xii)	وہ	سعودی	عرب	گیا			0	4
	‘He’	‘Saudi’	‘Arab’	‘Went’				
(iv)	وہ	سعودی	عرب	گیا			0	4
	‘He’	‘Saudi	Arab’	‘Went’				
(v)	وہ	سعودی عرب	گیا					
	‘He’	‘Saudi Arab’	‘Went’					

∗Visibility in the column: گیا, ∗∗Visibility in the column: عرب, #Visibility in the column: سعودی

1. Read next *bi*-gram of input text.

   - input_text: وہسعودیعربگیا ۔ (OHSAODYARBGYA, ‘HewenttoSaudiArabia’) →*bi*-gram: ا ۔ (A.)

2. As sentence boundary marker is detected.

   - store (ا (A)) on words_list[1][6] (see Table 3.4, Row (i) and C6)

3. Repeat the steps 4 to 8 (see Algorithm 1), until sentence ending marker, and for each row of *words_list*.

4. Select the row having minimum *word_counter* value (4) and flag_bit= false i.e. (see Table 3.4, Row (xii)).

5. For selected row, read each column and match in the compound word dictionary.

   - selected row: وہ سعودی عرب گیا ۔ (OH SAODY ARB GYA, ‘He went to Saudi Arabia’) (see Table 3.4, Row (xii)).

   - compound_word_dictionary: سعودی عرب (SAODY ARB, ‘Saudi Arab’).

   5.1. If match is found then read the next column of selected row in step 5 and repeat step 5 for the remaining part of selected row.

   - match is found in compound_word_dictionary for (سعودی (SAODY, ‘Saudi’)) i.e. table 3.4, index (iv), C2.

   - repeat step 5 for the remaining part of selected row i.e. (عرب (ARB, ‘Arab’) i.e. table 3.4, index (iv), C3.

   5.1.1. If complete match is found then concatenate with the column in step 5.

- concatenate with the column in step 5 i.e. (سعودی عرب SAODY ARB, 'Saudi Arab') (see Table 3.4, index (iv), C2).

6. Move each element of final selected row in step 5, decrease the array index.

    - move each element to the left: table 3.4, index (v), C3.
    - decrease the array index i.e. column=3;

7. Finally, the list of tokenized words will be produced i.e. Table 3.4, index (v).

3.2.4.2 Maximum Likelihood and Smoothing Estimation

In the proposed UNLTool-WT approach (see Algorithm 1) at step 11.1, a *tri*-gram MLE and smoothing estimations are used, because there can be multiple tokenized sequences for which *flag_bit=false* and *word_count* are equal. For instance, there are two given texts, (i) اسي باہر جا کي پڑھني دو (transliteration: ASE BAHR *JA* KE PRHNE DO, 'let him *go* abroad for higher studies'), and (ii) اسي باہر جی کي پڑھني دو (transliteration: ASE BAHR *JY* KE PRHNE DO, literally meaning 'let him *yes* abroad for higher studies'). Both have six tokens with flag_bit= false, but only the first text is semantically correct and meaningful. For such ambiguous cases, an *N*-gram language model is calculated with MLE for parameter and Laplace for smoothing estimation. The goal of these estimations is to find an optimized segmented sequence with the highest probability. This can be shown by a given mathematical expression, a general statistical model of the proposed UNLTool-WT approach.

$$P(t_1^n) \approx \prod_{j=1}^{n} P(t_k|t_{i-1}) \tag{3.1}$$

Where, $\prod_{j=1}^{n}$ denotes the probability of a complete word sequence of an input string i.e. $j_1 j_2 ... j_n$ with t tokens. Theoretically, it is assumed that the n-gram model outperforms with a high value of N. However, practically the data sparseness restricts better performance with high order N. Therefore, in the UNLTool-WT approach, the chosen value is *tri*-gram ($N = 3$) or *bi*-gram ($N = 2$) MLE. These have proved to be successful in several tasks for resolving ambiguity (e.g. POS tagging [38], automatic speech recognition [3] and word tokenization [70]).

The task of resolving similar sequence ambiguities for the above two texts is accomplished by using *tri*-gram MLE [97] as:

$$P(t_j|t_{j-2},t_{j-1}) = \frac{C(t_{j-2},t_{j-1},t)}{C(t_{j-2},t_{j-1})} \tag{3.2}$$

Where t represents the individual token, C is a count of three $(t_{j-2}t_{j-1}t)$ and two $(t_{j-2}t_{j-1})$ consecutive words in the dataset and P is the *tri*-gram contestant MLE value of each of the possible segmented sequences. The calculated probability for the first sequence is 3.2e-08 while for the second it is 0.

As *tri*-grams take account of more context, if this specific context is not found in the training data (see Section 3.2.5), we back-off to a narrower contextual *bi*-gram language model. *Bi*-gram cumulative probability values have been calculated as given by Jurafsky and Martin [97]:

$$P(t_j|t_{j-1}) = \frac{C(t_{j-1}t)}{C(t_{j-1})} \tag{3.3}$$

Where t represents the individual token, C is a count of two $(t_{j-1}t)$ and one (t_{j-1}) consecutive word(s) in the dataset and P is the *bi*-gram contestant MLE value of each of the possible segmented sequences. The calculated probability for the first sequence is 2.7e-6 for former sequence and 0 for later one.

These *zero* probabilities are again an underestimation of the input string, ultimately a cause for the data sparseness. Even if a statistical language model is trained on a very large dataset, it will remain sparse in some cases. However, there is always a possibility that the input text occurs in the test dataset [48], thus assigning them to zero made this an unstable, frail and specific estimator. Therefore, to overcome this, different *smoothing* techniques have been proposed in previous literature [97] with different characteristics (such as smoothing the probability etc.). Hence, it is primarily aimed at making a robust and generalize language model by re-evaluating lower or zero probability upwards and vice-versa for high probabilities.

In this research work, Laplace (a.k.a add-one) smoothing [94] is use, as one of the oldest, simplest and baseline estimations. This estimation adds one to all frequency counts, i.e. that all *bi*-gram probability counts have been seen one more time than actually exists in the training data as:

$$P_{add:1}\left(t_j|t_{j-1}\right) = \frac{1+C\left(t_{j-1},t\right)}{V+C\left(t_{j-1}\right)} \tag{3.4}$$

Where v represents the unique words (types), added to the total number of words $C(t_{j-1})$ in order to keep the probability normalized [97]. We have used Laplace smoothing to estimate the parameters required for data sparseness in order to increase the *bi*-gram MLE value for اسي باہر جى کي پڑھني دو (transliteration: ASE BAHR *JY* KE PRHNE DO, 'let him *go* abroad for higher studies'), from 2.7e-6 to 3.8e-7 and decreased value for اسي باہر جا کي پڑھني دو (transliteration: ASE BAHR *JA* KE PRHNE DO, literally meaning 'let him *yes* abroad for higher studies'), from 0 to 1.9e-14. The latter tokenized sequence has the highest smoothing MLE. Therefore, it will be selected by UNLTool-WT as the best tokenized sequence, which is correct.

3.2.5 Proposed Datasets for Urdu Word Tokenization Task

3.2.5.1 Testing Data

Another key element of this research is to develop a large benchmark dataset, for the evaluation of the proposed UNLTool-WT approach (see Section 3.2.4). The process of developing a benchmark test dataset is divided into three steps: (i) raw text collection, (ii) cleaning and (iii) annotation.

In the first phase, raw data is collected from various online sources (BBC Urdu[14], Express news[15], Urdu Library[16], Urdu Point[17], Minhaj Library[18], Awaz-e-Dost[19] and Wikipedia[20]) by using a Web crawler[21]. The collected raw data is free and publicly available for research purposes, and belongs to following genres: Commerce, Entertainment, Health, Weather, Science and Technology, Sports, Politics and Religion. This collected text contains 61,152 tokens.

In the next phase of the test dataset creation process, the collected raw text is pre-processed (see Section 3.2.6), which resulted in the removal of 2,152 tokens. The remaining cleaned data is composed of 59,000 tokens (3,583 sentences).

The quality of evaluation of an Urdu word tokenization approach depends on the annotation quality of the test dataset because inconsistent and noisy annotations deteriorate the model's performance. Thus, the annotations are performed by three different annotators (D, E and F). All the annotators are native speakers of Urdu. The annotation process is further divided into three phases: (i) training, (ii) annotation, and (iii) inter-rater agreement calculation and conflict resolution.

In the training phase, two annotators (D and E) annotated a subset of 58 sentences. After that, the inter-annotator agreement is computed for these sentences and conflicting tokens are discussed to further improve the annotation quality. In the annotation phase, the remaining test dataset comprising of 3,525 sentences is annotated

[14] http://www.bbc.com/urdu, terms of use: https://www.bbc.com/urdu/institutional-37588278 - Last visited: 05-April-2019

[15] http://www.express.pk/ - Last visited: 14-November-2018

[16] http://www.urdulibrary.org/ - Last visited: 14-November-2018

[17] http://www.urduweb.org/planet/ - Last visited: 14-November-2018

[18] http://www.minhajbooks.com/urdu/control/ - Last visited: 14-November-2018

[19] http://awaz-e-dost.blogspot.co.uk/ - Last visited: 14-November-2018

[20] https://ur.wikipedia.org/wiki/ - Last visited: 14-November-2018

[21] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 14-November-2018

by annotators D and E. After the annotation phase, the inter-rater reliability score is computed for the entire test dataset of 59,000 tokens. The inter-annotator agreement of 86.3% is obtained as the annotators had agreement on 50,917 pairs. The Kappa Coefficient is computed to be 78.09%, which is considered as good, considering the levels of difficulty for classifying the merge (space omission) and compound words (space insertion) into single or multiple tokens (see Section 3.2.1). Furthermore, the conflicting tokens are annotated, and decisions resolved by the third annotator F, which resulted in a gold standard UNLTool-Word Tokenizer-Test (UNLTool-WT-Test) dataset.

The Table 3.5 shows the type-token ratio of the UNLTool-WT-Test dataset, that have a total of 59,000 tokens and 5,849 types. The UNLTool-WT-Test dataset is stored in the standard "txt" format and is free and publicly available for research purposes (under the terms of the Creative Commons Attribution 4.0 International License[22].).

Table 3.5 Domain statistics of the UNLTool-WT-Test dataset

Domains	**Tokens**	**Types**
Commerce	7,254	663
Entertainment	8,578	937
Health	6,765	651
Weather	6,606	756
Science and Technology	7,749	823
Sports	6,868	691
Politics	9,627	777
Religion	5,553	556
Total	59,000	5,849

3.2.5.2 Training Data

The training dataset for a proposed Urdu word tokenizer is created by using a subset of the HC Corpus [50] (see Section 2.3.8). To develop a gold standard training

[22] https://creativecommons.org/licenses/by/4.0/ - Last visited: 11-November-2018

dataset, two million tokens are randomly selected from the following domains: Politics, Culture, Crime & Law, Fashion, Religion, Business & Economy, Science & Technology, Sports, Weather, Education, Health, Entertainment.

After pre-processing (see Section 3.2.6) the collected raw data, the resulting dataset contained 1.65 million tokens. The pre-processed text is used to create the gold standard training dataset. In the first step, the text is tokenized on the basis of space. After that, a human annotator manually corrected the improperly tokenized words generated in the first step. The final benchmark training dataset (hereafter called UNLTool-WT-Train dataset) is comprised of 1.65 million tokens.

The UNLTool-WT-Train dataset is used to generate N-grams using the approach (see Algorithm 2) described in [97]. Furthermore, the occurrences of each unique N-gram type is counted, resulting in a total 1,335,263 N-gram pairs with the following statistics: *tri*-grams: 636,765, *bi*-grams: 494,988 and *uni*-grams: 203,510.

Algorithm 2 N-gram model generation algorithm

```
procedure GENERATENGRAMS(int $s$)
    Initialize int $N$ (size of $n-$gram) $= s$;
    Initialize List $ngramList$ (to store generated n-grams);
    Initialize String[] $tokens = UNLTool\text{-}WT\text{-}Train$ dataset;
    Initialize int $k = 0$;
    for for each k<tokens.length-N+1 do
        String $st =$ “”;
        int $start = k$;
        int $end = k + N$;
        $j = start$;
        for for each j<end do
            $s = s +$ “” $+ tokens[j]$;
        end for
        ngramList.add(s);
    end for
    return $ngramList$
end procedure
```

3.2.6 Pre-processing

In this study, various datasets have been used, all these datasets (see Sections 3.2.5, 3.4.5, and 3.3.3) are pre-processed as follows. Text in a dataset is cleaned by removing multiple spaces, duplicated text, diacritics as they are optional (only used for altering pronunciation [143]) and HTML tags. Moreover, noise from the data is removed by discarding ASCII and invalid UTF-8 characters, emoticons, asterisks, bullets, right and left arrows [91]. Further, only sentences with three or more words are kept[23]. A language detection tool[24] is used to discard foreign words and a text normalization tool[25].

3.2.7 Experimental set-up

3.2.7.1 Datasets

For the set of experiments presented in this study, the UNLTool-WT-Train (containing 1.65 million tokens) and UNLTool-WT-Test (containing 59K tokens) datasets are used for training and testing of the proposed UNLTool-WT approach respectively.

3.2.7.2 Approaches

For this study, four approaches for word tokenization are applied: (i) word tokenization on the basis of space (henceforth UNLTool-WT-SP approach), (ii) a hybrid

[23] This is calculated by dividing the total words in dataset by the total number of sentence disambiguation markers.

[24] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 14-November-2018

[25] Text normalization tool can be downloaded from http://www.cle.org.pk/software/langproc/urdunormalization.htm-Lastvisited:14-November-2018 is used to keep the Unicode of the characters consistent.

approach of tokenization [66] (hereafter Durani's), (iii) CLE's word tokenization[26], and (iv) word tokenization using a proposed UNLTool-WT Algorithm 1.

3.2.7.3 Evaluation Measures

The evaluation of the proposed Urdu word tokenizer is carried out using precision, recall, F_1 measure, accuracy, and standard deviation (see Section 2.5.3).

3.2.8 Results and Analysis

Table 3.6 presents precision, recall, F_1 and accuracy results when training on UNLTool-WT-Train dataset, and testing on the UNLTool-WT-Test for Urdu word tokenization task by using various approaches (rule-based maximum matching, dictionary lookup, statistical *tri*-gram MLE with backed-off to *bi*-gram MLE along with smoothing). The standard deviations (σ) associated with the computed results have also been presented. UNLTool-WT-SP refers to results obtained using space-based tokenization approach. UNLTool-WT refers to results obtained using the proposed approach for Urdu word tokenization. Durani's refers to a hybrid method (see Section 2.3.5 of Chapter 2). Whereas, CLE's word tokenizer refer to an online tokenizer (the online link refers three papers but does not describe which one of them is used for the creation of CLE Urdu word tokenizer).

Overall, the best results are obtained by using a proposed UNLTool-WT approach (precision = 0.96, recall = 0.92, F_1 = 0.94, and accuracy = 0.97). These results show that UNLTool-WT is the most appropriate method for Urdu word tokenization on the UNLTool-WT-Test dataset. Furthermore, this also shows that combining maximum matching, dictionary lookup and statistical N-gram MLE along with smoothing estimation are helpful in getting good performance on UNLTool-WT-Test dataset

[26] tokenize up-to 100 words at one time and implementation details are not available http://www.cle.org.pk/clestore/segmentation.htm - Last visited: 18-Dec-2019

for Urdu word tokenization task. However, the highest F_1 score of 0.94 for the word tokenization task indicates that Urdu word tokenization is a challenging task leaving a room of improvement.

Table 3.6 Results obtained on UNLTool-WT-Test dataset using various techniques

Technique	**Precision**$\pm\sigma$	**Recall**$\pm\sigma$	**F$_1$-measure**$\pm\sigma$	**Accuracy**$\pm\sigma$
UNLTool-WT-SP	0.55 ± 0.27	0.52 ± 0.25	0.54 ± 0.17	0.61 ± 0.21
UNLTool-WT	0.96 ± 0.08	0.92 ± 0.11	0.94 ± 0.09	0.97 ± 0.06
Durani's	0.18 ± 0.39	0.20 ± 0.36	0.19 ± 0.29	0.49 ± 0.40
CLE's	0.58 ± 0.29	0.56 ± 0.30	0.57 ± 0.18	0.73 ± 0.28

As expected, the overall results for UNLT-Tool-WT approach are higher as compared to all other baseline approaches (see Figure 3.1): space-based tokenization UNLTool-WT-SP approach report precision = 0.55, recall = 0.52, $F_1 = 0.54$, and accuracy = 0.61, on UNLTool-WT-Test dataset. Durani's word tokenizer report an accuracy of 0.49, precision of 0.18, recall of 0.20, and $F_1 = 0.19$. Furthermore, the CLE's Urdu word tokenizer has show precision = 0.58, recall = 0.56, $F_1 = 0.57$, and accuracy = 0.73. This highlights the fact that the UNLTool-WT-SP, Durrani's and CLE's approaches are not suitable for Urdu word tokenization tasks.

While analysing the errors of the proposed UNLTool-WT approach, its is observed that it does not explicitly handle unknown words for space omission, and this resulted in splitting an unknown Urdu morpheme into smaller morphemes. For instance, the word کثیراللسان (KSYR ALLSAN, 'multilingual') erroneously split into کثی (KSY), را, (RA) للس (LLS), and ان (AN). Likewise, it might be less appropriate when a word is a combination of known and unknown morphemes, for instance, شہبازکوجانیدو (SHBAZ KO JANE DO, 'let the Shahbaz go'). For space insertion, some compound words are not found in the compound words dictionary, another major cause of incorrect word tokenization.

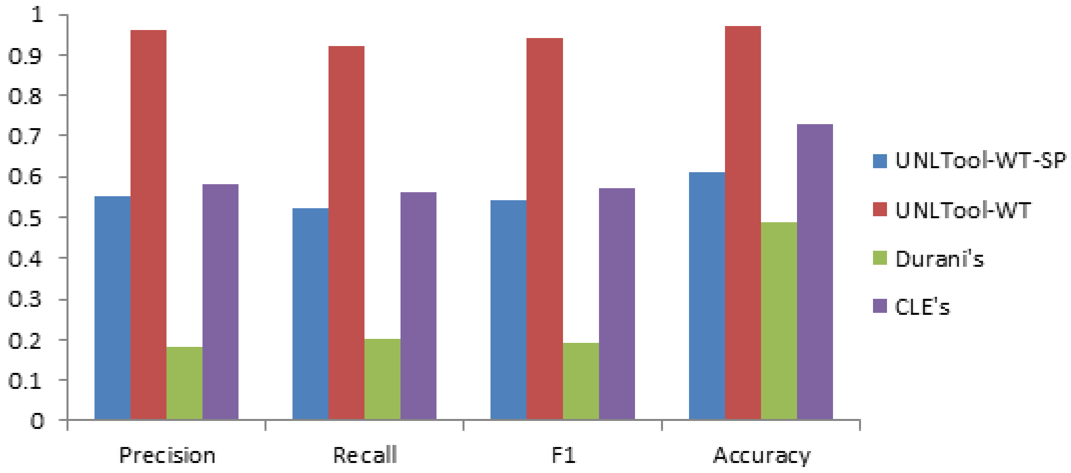


Fig. 3.1 Performance comparison of various Urdu word tokenizers on UNLTool-WT-Test dataset

3.3 Urdu Sentence Tokenizer

This part presents challenges faced in the Urdu sentence tokenization task, the proposed rule based Urdu sentence tokenizer, the test dataset which has been developed to evaluate the proposed sentence tokenization approach, experimental set-up, and results along with their analysis.

3.3.1 Challenges Of Urdu Sentence Tokenization

Sentence boundary detection is a non-trivial task for Urdu text because: (i) it does not use any special distinguishing characters between upper and lower case, (ii) punctuation markers are not always used as sentence separators and (iii) there is a lack of standard evaluation and supporting resources. For English and other languages, the difference in upper and lower case is helpful in identifying sentence boundaries. Furthermore, in English language there is a convention that if a period is followed by a word starting with a capital letter then it is more likely to be a sentence marker, whereas, in Urdu, there are no upper and lower-case distinctions.

Punctuation characters such as “-”, “.”, “؟” and “!”′ are used as sentence terminators and these can also be used inside the sentence.

Table 3.7 shows example Sentence Boundary Markers (SBM) (such as sentences at index i, ii, iii, and iv, in all these sentences question, period, exclamation, and double quotes marker are used at the end of sentences to represent a sentence boundary) and Non-Sentence Boundary Markers (NSBM) for Urdu text. It can be observed from these examples that the NSBM are also frequent because they are being used between dates (such as sentence at index vii, in this sentence a period mark is used with in a sentence which is actually not a sentence boundary), abbreviations (index v, this sentence is composed of several period markers, however first two are not indicating a sentence boundary marker), emphatic declaration (index vi, here exclamation marker is used with in a sentence i.e. not a sentence boundary mark), names and range (index viii i.e. a first period and double quote marker is used within a sentence but both are not a sentence ending marker). Consequently, these kind of examples makes the sentence tokenization of Urdu text a challenging task.

3.3.2 Proposed Urdu Sentence Tokenization Approach

Two existing broad approaches for sentence tokenization are: (i) rule-based and (ii) machine learning-based (see Section 2.3.6). To develop supervised machine learning-based approaches, a large amount of training data is required. Urdu is an under-resourced language and there is a lack of large annotated datasets, therefore, a rule-based approach is used for proposed sentence tokenizer. It can be observed that all existing Urdu sentence tokenizers (see Section 2.3.6) are based on statistical approaches. Here, previously unexploited rules for Urdu sentence tokenization are adopted.

Table 3.7 Examples showing Sentence Boundary Markers (SBM) and Non-Sentence Boundary Markers (NSBM) for Urdu text

Index	**Marker**∗	**Text**
i	QM-SBM	مشرف کو باہر کیوں جاني دیا گیا ؟ ؟ GYA DYA JANE KYON BAHR KO MSHRF ‘Why was Musharraf let to go abroad?’
ii	PM-SBM	انڈیا مس آئی سی سی ورلڈ ٹی ٠٢ کا آغاذ ۔ ۔AAGHAZ KA 20 TY ORLD SY SY AAYY MY ANDYA ‘Inauguration ceremony of ICC world T 20 held in India.’
iii	EM-SBM	اس پر بہی عوام نہ سمجھي تو ! ! TO SMJHY NH AOAM BHY PR IS ‘Even then if public do not understand then!’
iv	DQ-SBM	« مري خيال مس ان کي وزير خارجہ ٢١ اگست کو آرہي ہيں » "HYN RHY AA KO AGST 21 KHARJH OZYR KE AN MY KHYAL MYRE" ‘In my opinion the foreign minister is visiting on August 21st’
v	PM-NSBM	يو۔ اي ۔ اى میں کافی پاکستانی بستي ہیں ۔ HYN BSTE PAKSTANY KAFY MY AY- AE -YO ‘Many Pakistanis are living in U.A.E.’
vi	EM-NSBM	حضور والا ! آپ پوري ملک کي بادشاہ ہیں ۔ - HYN BADSHH KE MLK PORE AAP ! OALA HZOR ‘My lord! You are the king of this country.’
vii	PM-NSBM	آج ٥١٠٢ - ٦ - ٣ ہي - - HYN 3-6-2016 AAJ ‘Today is 3rd of May 2015.’
viii	PM-NSBM DQ-NSBM	”پاکستان“ ٢ - ٤ سي جيت رہا ہي - - HE RHA JET SE 3 - 2 PAKSTAN ‘ "Pakistan" is winning by 2-4.’

∗ M: Question Mark, PM: Period Mark, EM: Exclamation Mark, DQ: Double Quotes

For the proposed rule-based approach, to manually extract rules for the sentence tokenization task, initially, a subset of the UMC dataset [91] comprised of 13K sentences are selected, which contains Urdu text from various domains or genres including News, Religion, Blogs, Literature, Science and Education. After the pre-processing (see Section 3.2.6) 10K sentences are retained, which have been used to extract rules to develop proposed Urdu sentence tokenizer.

The rules are devised to include sentence termination markers (۔, ؟, ” and !), regular expressions and supplementary dictionary lookup[27] (henceforth UNLTool-ST-RB approach). These heuristics are applied as follows:

1. If the current character is a period marker (۔) *AND* the same mark appears after two or three characters, then consider it as an abbreviation and match it in the abbreviation list.

2. If within the next 9 characters (from any previous SBM marker), an exclamation mark (!) is found, then this is not a sentence boundary marker.

3. If the character before a double quote (”) is a period (۔) or question (؟) mark, then it is a sentence boundary marker.

4. Apply regular expressions for detecting the date and hyphenated numeric values.

5. In addition to this all the above rules from 1 to 4, split sentences based on the question (؟), period (۔) and exclamation (!) markers.

Table 3.8 shows an example of Urdu text tokenized into sentences using a proposed UNLTool-ST-RB sentence tokenizer. As can be noted, the raw Urdu text is split into sentences on the basis of SBMs (see index 1), whereas for NSBMs the raw text has not been split into sentences (see index 2).

[27] Same dictionary compiled for the word tokenization task (see Section 3.2.3) is used.

Table 3.8 Examples of Urdu text split into sentences using a proposed UNLTool-ST-RB sentence tokenizer

Sentence Tokenized Text
Index 1
گلگت بلتستان مىں جاری شدىد بارشوں سي اب تک ٥٣ افراد ہلاک ہوگئي ہىں ۔ 35 people have been killed due to continuous heavy rain in Gilgit-Balitstan.
کىا وىسٹ انڈىز سي اس شکست کي بعد کىا وہ مزىد بہی کھىلنا جاری رکھىں گي؟ Will he continue to play as a captain after this defeat by West Indies?
ان مىں اىک تحرىک جس مىں بہت ذىادہ لوگ شامل ہوئے جنہوں ني کہا سىاست نہىں رىاست بچاؤ ! Among them, the one movement in which many people participated was that whose moto was to save the state not politics!
« غىر ملکی سىاح چترال کو پىراگلائىڈنگ کی جنت بہی کہتي ہىں » “Foreign tourists also call Chitral as a paradise of paragliding”
Index 2
فرنس آئل کی کھپت مىں جولائی ٥١٠٢ کي مقابلي مىں ٨٢ ۔ ٩٢ اضافہ ۔ An increase of Furnace oil consumption is recorded by 28-29 percent, compared to July 2015.
آئی ۔ سی ۔ اي ۔ سی کو اس عرصي کي دوران بالواسطہ ٹىکسوں مىں ٤١ فىصد کی کمی رہی ۔ During this period I.C.A.C faced a reduction of 14% indirectly paid taxes.
اس تحرىک کي مرکزی دفتر کا سنگ بنىاد ٧-٩-٥٨٩١ کو رکھا گىا ۔ Its head office was founded on 7-9-1985.
ىہاں آئي ! کچھ مستی کرتي ہىں ۔ Come here and let’s have some fun.
جنرل ۔ ر ۔ شرىف ني زىر علاج زخمىوں کی شام چھي ۔ سات بجي عىادت کی ۔ General-R-Shareef visited the hospital between six and seven p.m. to inquire the patients after their health.

3.3.3 Test Dataset for Urdu Sentence Tokenization

For the evaluation of the proposed Urdu sentence tokenizer, a benchmark dataset (hereafter called UNLTool-ST-Test dataset) is created by following three steps: (i) raw Urdu text collection, (ii) pre-processing of raw data and (iii) annotation.

To construct the UNLTool-ST-Test dataset, in the first step, a Web crawler[28] is used to extract raw Urdu text of 10K sentences from online sources (see Section 3.2.5)

[28] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 14-November-2018

including: BBC Urdu, Express news, Urdu Library, Urdu Point, Minhaj Library, Awaz-e-Dost and Wikipedia. These sources allow their text (content) to be freely used for research purposes. To make the dataset more realistic, we extracted the raw text of different domains and genres including Sports, Politics, Blogs, Education, Literature, Entertainment, Science, Religion, Fashion, Weather, Entertainment, Fiction, Health, Law and Business. BBC Urdu is the largest source of the text collection, which contains 3,000 sentences, while the Urdu Point is the smallest one, containing 800 sentences. Statistics of sentences collected from other sources are: Awaze-e-Dost: 1,100, Express news: 1,200, Minhaj library: 1,300, Urdu library: 1,000, and Wikipedia: 1,600 sentences.

In the second step, the raw data has been pre-processed (see Section 3.2.6), which resulted in the removal of 2,000 sentences. The remainder of the 8,000 clean sentences are distributed as follows: Awaz-e-Dost: 915, BBC Urdu: 2,316, and Express News: 1,012, Minhaj Library: 1,018, Urdu Library: 834, Urdu Point: 663 and Wikipedia: 1,242 sentences.

In the third step, the pre-processed text containing 8,000 cleaned sentences are manually tokenized by three annotators (G, H and I). All the annotators are native speakers of Urdu and have good knowledge about the Urdu sentence tokenization task. Furthermore, the annotation process was split into three phases: (i) training, (ii) annotation and (iii) inter-rater agreement and conflict resolution.

During the training phase, two annotators (G and H) annotated 200 sentences. Subsequently, the inter-annotator agreement has been computed for these sentences and conflicting sentences are discussed to further improve the annotation quality. Further, during the annotation phase, the remaining 7,800 sentences are manually annotated by annotators (G and H). In the third phase, the inter-rater agreement score is computed for all 8,000 sentences. We achieved an inter-rater agreement of

92%, as the annotators agreed upon 7,350 sentences. Moreover, the Kappa Coefficient has been computed to be 83.69% [53]. The conflicting 650 sentences are annotated by the third annotator (I) for conflict resolution and this judgement is considered as decisive, resulting in the gold standard UNLTool-ST-Test dataset.

The UNLTool-ST-Test dataset consists of 8,000 sentences (see Table 3.9). In proposed test dataset, 6,469 period markers are SBM, while 536 are NSBM; 531 exclamation marks are SBM and 198 are NSBM; 421 question marks are SBM and 17 are NSBM; 203 double quotes, 194 double quotes are SBM and 9 are NSBM; the remaining 382 SBM markers are #, @, $, * etc. As can be noted from these statistics, the proposed UNLTool-ST-Test dataset contains both SBM and NSBM for different characters, which makes the dataset much more realistic and challenging. The UNLTool-ST-Test dataset is saved in standard "txt" format.

Table 3.9 Statistics of UNLTool-SD-Test dataset

Sources	**Sentence Count**
Awaz-e-Dost	915
BBC Urdu	2,316
Express News	1,012
Minhaj Library	1,018
Urdu Library	834
Urdu Point	663
Wikipedia	1,242
Total	8,000

3.3.4 Experimental Set-up

3.3.4.1 Dataset

For the set of experiments presented in this section, the entire UNLTool-ST-Test dataset is used which contains 8,000 sentences (see Section 3.3.3).

3.3.4.2 Approaches

For this study, we applied five different approaches for sentence tokenization: (i) baseline approach– sentence tokenization on the basis of "period", "question mark", "exclamation mark", and "double quotes" characters (henceforth UNLT-ST-PQEQM approach), and (ii) rule base proposed approach– sentence tokenization by using a proposed UNLTool-ST-RB approach.

3.3.4.3 Evaluation Measures

The evaluation of sentence tokenization techniques is carried out using precision, recall, F_1, error rate and standard deviation measures (see Section 2.5.3).

3.3.5 Results and Analysis

Table 3.10 presents precision, recall, F_1 and error rate results on the UNLTool-ST-Test dataset for various Urdu sentence tokenization approaches. The standard deviations associated with the computed results are also presented.

Table 3.10 Results obtained by using various sentence tokenization approaches on UNLTool-ST-Test dataset

Technique	**Precision**$\pm\sigma$	**Recall**$\pm\sigma$	$\mathbf{F_1}$**-measure**$\pm\sigma$	**Error rate**$\pm\sigma$
UNLT-ST-PQEQM	0.94 ± 0.10	0.24 ± 0.21	0.27 ± 0.17	0.79 ± 0.12
UNLTool-ST-RB	0.91 ± 0.12	0.94 ± 0.07	0.93 ± 0.09	0.07 ± 0.03

Overall, the best results are obtained using the proposed UNLTool-ST-RB approach (precision = 0.91, recall = 0.94, F_1 = 0.93, error rate = 0.07). This shows that combining various heuristics, regular expressions and dictionary lookup is helpful in producing a good performance on the UNLTool-ST-Test dataset. The highest F_1 score of 0.93 for sentence tokenization task indicates that Urdu sentence tokenization is a challenging task and there is still room for further improvement.

Other approach (UNLT-ST-PQEQM) which use different characters as a sentence boundary indicator, shows precision of 0.94 (see Figure 3.2). The likely reason for this is that the majority of sentences in Urdu text are terminated using several characters (see Section 3.3.3 for statistics on UNLT-ST-Test dataset). However, other evaluations measures shows very low results (recall = 0.24, F_1= 0.27, error rate = 0.79). This highlights the fact that these characters alone are not suitable for Urdu sentence tokenization task.

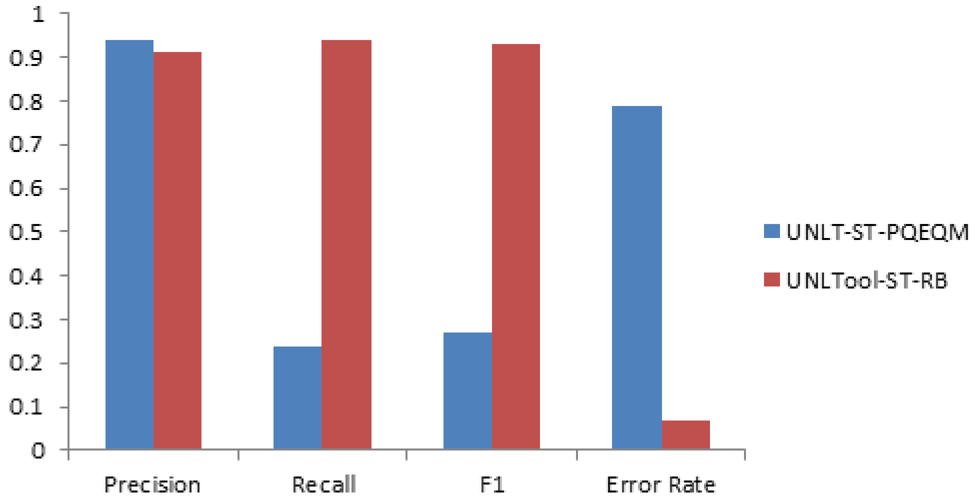


Fig. 3.2 Performance comparison of Urdu sentence tokenizers on UNLTool-ST-Test dataset

While manually analysing the errors of the proposed UNLTool-ST-RB approach, some scenarios have been observed where the proposed approach failed to accurately tokenize sentences. It is found that NSBM including: ':', '||', '\$', '$*$', '@' and '#' are the major reasons for incorrect tokenization of sentences. Moreover, the period used between different abbreviations also caused misclassification.

3.4 Urdu Part of Speech Tagging

This part presents the challenges faced in the Urdu Part-of-Speech (POS) tagging task, proposed statistical based Urdu POS taggers, the training and test datasets that are developed to train and evaluate proposed POS tagging approaches, experimental set-up and finally, results and analysis.

3.4.1 Challenges of Urdu POS Tagging

POS tagging for the Urdu language is a challenging and difficult task due to four main problems [143]: (i) free word order (general word order is SOV), (ii) polysemous words, (iii) Urdu is highly inflected and morphologically rich, and (iv) the unavailability of gold-standard training/testing dataset(s). We briefly discuss these issues here.

Firstly, Urdu sentences have a relatively complex syntactic structure compared to English. Table 3.11 shows examples of the free word order and its semantic meaningfulness in the Urdu language. Secondly, as with other languages, Urdu also has many polysemous words, where a word changes its meaning according to its context. For example, the word باسی (BASY) means 'stale' if it is an adjective and 'resident' when it is a noun. Thirdly, Urdu is also a highly inflected and a morphologically rich language because gender, case, number and forms of verbs are expressed by the morphology [83, 205]. Moreover, Urdu language represents case with a separate character after the head noun of the noun phrase [205]. They are sometimes considered as postpositions in Urdu due to their place of occurrence and separate occurrence. If we will consider them the case markers, then Urdu has accusative, dative, instrumental, genitive, locative, nominative, and ergative cases ([95]: Pg 10). Usually, a verb phrase contains, a main verb, a light verb (which use to describe the aspect) and a tense verb (describes the tense of the phrase) [83, 205].

Finally, there is a lack of benchmark training/testing datasets that can be used for the development and evaluation of Urdu POS taggers.

Table 3.11 Free word order example text for Urdu language

Sentence	Meaningful	Translation
شر گوشت کو کھا رہا ہي HE RHA KHA KO GOSHT SHAYR	Y∗	Lion is eating the meat
گوشت کو شر کھا رہا ہي HE RHA KHA SHAYR KO GOSHT	Y	Meat is eaten by the lion
شر کھا رہا ہي گوشت کو KO GOSHT HE RHA KHA SHAYR	Y	Lion eating is meat
کھا رہا ہي شر گوشت کو KO GOSHT SHAYR HE RHA KHA	Y	Eating is lion meat
رہا ہي کھا شر گوشت کو KO GOSHT SHAYR KHA HE RHA	Y	Eating lion meat is
شر ہي رہا کھا گوشت کو KO GOSHT KHA RHA HE SHAYR	Y	Lion meat eating is

∗: Y: Yes

3.4.2 Existing Urdu POS Tagset

The tagging accuracy of a POS tagger is not only dependent on the quality and amount of training dataset but also on the POS tagset used for annotation. In the prior literature, we found three commonly used POS tagsets for the Urdu language: (i) Hardie's POS tagset [84], (ii) Sajjad's POS tagset [205] and (iii) Centre for Language Engineering (CLE) Urdu POS tagset [225].

Hardie's POS [84] tagset was an early attempt to resolve the grammatical tag disambiguation problem for the Urdu language. This tagset follows the EAGLES[29] guidelines and consists of 350 morphosyntatic tags, which are divided into 13 main categories. Some grammarians [169] propose only three main categories whereas

[29] http://www.ilc.cnr.it/EAGLES96/home.html - Last visited: 07-December-2016

[212] used 10 main categories for Urdu text. There were a number of shortcomings observed in Hardie's POS tagset [84]. For example, the possessive pronouns like میرا (MYRA 'my'), تمہارا (TMHARA 'your') and ہمارا (HMARA 'our') are assigned to the category of *possessive adjective*, which is incorrect. Many grammarians marked them as *pronouns* [169, 90]. Moreover,the Urdu language has no *articles* but this tagset defined articles. Another issue with the tagset is the use of *locative* and *temporal adverbs* such as یہاں (YHAN 'here'), وہاں (OHAN 'there'), and اب (AB 'now'), which are treated as *pronouns*. The *locative* and *temporal* nouns such as صبح (SBH 'morning'), شام (SHAM 'evening'), and گھر (GHR 'home') appear in a very similar syntactic context. To conclude, these grammatical misclassifications as well as the large number of POS tags with relatively small training data will affect the accuracy of POS taggers developed for the Urdu language.

Another POS tagset (henceforth Sajjad's POS tagset) [205], consists of 42 POS tags with finer grained categories for pronouns and demonstratives. However, it is lacking in terms of Urdu verb, tense and aspect.

A recently released CLE Urdu POS tagset [225] contains 35 tags and addresses most of the issues reported above. It is based on the critical analysis of several previous iterations of Urdu POS tagsets. Furthermore, it is built on the guidelines of the Penn Treebank[30] and a POS tagset for common Indian languages[31]. In the CLE Urdu POS tagset, a verb category has multiple tags based on the morphology of the verbs. Furthermore, it has shown promising results on Urdu text (see Section2.3.7).

For this study, the CLE Urdu POS tagset [225] is selected for following reasons: (i) it provides correct grammatical classifications, (ii) it provides purely syntactic

[30] https://www.cis.upenn.edu/treebank/ - Last visited: 05-April-2019

[31] https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/I08-7013.pdf - Last visited: 11-November-2018

categories for major word classes and (iii) provides reasonable performance on a small size test dataset.

3.4.3 Proposed Urdu POS Tagging Approaches

For this study, we applied two stochastic approaches for Urdu POS tagging: (i) *tri*-gram Hidden Markov Model and (ii) Maximum Entropy-based model. The reason for selecting these two methods for Urdu POS tagging is many fold, (a) they have proven to be effective for POS tagging not just for English [250] but also for other languages which are closely related to Urdu such as Hindi [96, 58], (b) both are well established stochastic models for automatic POS tagging task [242], (c) these methods have been primarily investigated for under resourced or when dealing with languages with limited resources [20, 67], and (d) these models have not been previously compared for the Urdu language.

3.4.3.1 Hidden Markov Model (HMM) for POS Tagging

In general, the Urdu POS tagging task can be formulated as: given a sequence of words $w_1,...,w_n$, find the sequence of POS tags $t_1,...,t_n$ from a POS tagset T[32] using some statistical model. In this section a HMM stochastic learning model has been used as described by [174], while [229] redefined it for the POS disambiguation task. This model is implemented in [76, 31] for POS tagging. For experiments, a third order HMM learning model is used, also referred to as *tri*-gram POS tagging. This model is composed of transitional (contextual) and lexical (emission) probabilities and using Bayes' theorem, the HMM 3^{rd} order model can be written as:

[32] 35 tags as in CLE Urdu POS tagset: http://www.cle.org.pk/software/langproc/POStagset.htm - Last visited: 11-November-2018

$$\hat{t}_1^n = \underset{t_1^n}{argmax} \prod_{j=1}^{n} P \underbrace{(t_j|t_{j-1},t_{j-2})}_{Transition} P\underbrace{(w_i|t_i)}_{Lexical} \tag{3.5}$$

During the training process, the above *tri*-gram HMM language model (see Equation 3.5) computes two probability factors for the sequences: (i) lexical probabilities, aimed at determining the probability of a particular tag conditioned on particular word, and (ii) transitional probabilities, used to find the probability of a particular tag on the basis of given preceding tag(s). Given a sentence, the aim of the HMM language model is to search the tagging sequence and choose the most likely sequence that maximises the dot product of lexical and transition probabilities. That can be computed by using a Viterbi algorithm [236].

3.4.3.1.1 Parameters Estimation The HMM parameters can be estimated by applying the simplest *tri*-gram MLE (see Section 3.2.4), used for computing relative frequencies. A training dataset (see Section 3.4.5) has been used to find tag frequency counts (C) for two or three consecutive tag pairs (t_{j-2},t_{j-1},t_j), (t_{j-2},t_{j-1}). Where, t_j is the j_{th} tag of annotated dataset used during training process. The following equation requires frequency counts of w_it_i, where w_i is the word and t_i is the tag assigned to i_{th} word. The *tri*-gram language model (see Section 3.2.4) and the following equation is used with these parameter settings, $1 \leq (i,j) \leq n$.

$$P(w_i|t_i) = \frac{C(w_it_i)}{C(t_i)} \tag{3.6}$$

3.4.3.1.2 Smoothing The MLE has been used for parameter estimation (see Section 3.4.3), consequently, such models may come across a situation where unseen events do not occur or have quite low frequencies in the trained model. Therefore, the zero probability of such occurrences produces problems in the multiplication of probabilities, eventually, leading to a data sparseness.

To avoid data sparseness, there is a need of some estimators that automatically assign a part of the probability mass to unknown words and tag sequences, thus yielding an improvement for unseen events and overall accuracy improvement for the POS tagger. For this, different smoothing techniques have been cited in the literature with an objective to decrease the probability of seen events and assigning appropriate non-zero probability mass to unseen events. In this study, three different smoothing techniques are adopted including: (i) linear interpolation, (ii) Laplace and (iii) Lidstone's estimations. Adopting them with an HMM model thus alleviates sparse data issues.

3.4.3.1.3 Linear Interpolation: A well-practised smoothing technique consists of linearly combined estimation for different order *n*-grams as:

$$P(t_i|t_{i-1},t_{i-2}) = \lambda_1\rho(t_i) + \lambda_2\rho(t_i|t_{i-1}) + \lambda_3\rho(t_i|t_{i-1},t_{i-2}) \tag{3.7}$$

Where P is a valid probability distribution, ρ are maximum likelihood estimates of the probabilities and $\lambda_1 + \lambda_2 + \lambda_3 = 1$ to normalise the probability. Although, there are different ways to estimate λs, but for the experiments conducted here, a *deleted linear interpolation* is adopted as cited in [38].

The deleted linear interpolation successively removes each *tri*-gram from the training dataset. Moreover, this technique estimates the best value for the λs from all other *n*-grams in the dataset, making sure that the value of λ does not depend upon the particular *n*-gram. Further, it computes the weights depending on the counts of each *i*-gram, involved in the interpolation. Thus, the first HMM based proposed model is a combination of linear interpolation smoothing technique along with *tri*-gram HMM model (henceforth T-HMM-LI).

3.4.3.1.4 Laplace and Lidstone's Estimation: Laplace estimation (one of the oldest and simplest smoothing techniques) updates the count by one of each *bi*-gram occurrences compared to the actual frequency in training data [97] (see Section 3.2.4). Lidstone's smoothing estimation [125] generalizes Laplace, by adding an arbitrary value to all (seen or unseen) events. Although the values for λ can be calculated using different methods, for experiments presented here, the same value cited in the research article [125] has been used, i.e. a well-known Expected Likelihood Estimation (ELE). Thus, Lidstone's estimation [125] can be calculated as:

$$P_{lidstone(x,\lambda)} = \frac{\lambda + C(X)}{V\lambda + N} \quad \lambda = 0.5 \tag{3.8}$$

Where V represents the unique words (vocabulary) against the total number of words N to keep probabilities normalized [97]. The generalized formulation of Lidstone's and Laplace estimation in an HMM-based Urdu tagger is as follow:

$$\pi_i = \frac{C(s_i(t=0)) + \lambda}{C(tokens) + V_{tag}\lambda} \tag{3.9}$$

$$a_{ij} = \frac{C(s_i \rightarrow s_j) + \lambda}{C(tokens) + V_{tag}\lambda} \tag{3.10}$$

$$P(s_j) = \frac{C(s_j) + \lambda}{C(tokens) + V_{tag}\lambda} \tag{3.11}$$

$$P(w_k) = \frac{C(w_k) + \lambda}{C(tokens) + V_w\lambda} \tag{3.12}$$

Here, V_{tag} is the number of possible tags and V_w is the size of the approximated vocabulary.

The proposed second POS tagging model is a combination of Laplace and *tri*-gram HMM model (henceforth T-HMM-LaE). The third POS tagger makes use of Lidstone's estimation and supervised *tri*-gram HMM model parameters (we shall call this T-HMM-LiE).

3.4.3.2 Maximum Entropy (MaEn) Markov Model for POS Tagging:

The other adopted stochastic learning model is MaEn, and aimed to compare this to the above described *tri*-gram HMM based models, to find the most optimal POS tagger for Urdu. The MaEn statistical assumption is a simplistic model, it assigns a probability distribution for every tag, given a word and its context as:

$$\hat{T} = \underset{T}{argmax} \prod_{j=1}^{n} P(t_j|c_j, t_{j-1}) \tag{3.13}$$

Where, t is the individual tag in the set T of all possible tags i.e. $t_1, ..., t_n$ for a given a sentence, c is defined as the context, usually defined as the sequence of words $w_1, ..., w_n$ and the tag preceding the word. The maximum likelihood tag sequence is used for assigning probabilities to a string of input words.

The principle of estimating probabilities in MaEn model is to make as few assumptions as possible, other than the constraint imposed. Furthermore, these constraints are learned from the training data, which express some relation between features extracted and outcome. The probability distribution which satisfies the above property has the highest entropy, thus, it agrees with the maximum likely-hood distribution, and has a general form as cited in [178]:

$$P(t|c) = \frac{1}{N} exp \sum_{j=1}^{k} \alpha_j f_j(c, t) \tag{3.14}$$

Where N is the total number of training samples (normalization constant), f_j is feature function on the event (c,t). Feature functions used by MaEn model are binary valued and defined to capture relevant aspects of language. The α_j is a model parameter with k features, which is determined through the Generalize Iterative Scaling (GIS) algorithm [57]. However, these model values and features, are primary ingredients of MaEn learning model.

3.4.3.3 Features Selection in MaEn Model

As described previously the MaEn is feature based probabilistic model, to obtain high accuracy two binary valued features are used that might be helpful for predicting POS tag, these are determined empirically for Urdu POS tagging along with MaEn model as: (i) context window, and (ii) word number.

The best context window with five words has been identified, which is comprised of n-gram (W_{i-2}, W_{i-1}, W_i, W_{i+1}, W_{i+2}) and n-POS (t_{i-2}, t_{i-1}, and t_i) information.

If the current word is a number such as "٢٥٠٣١", another feature can be created:

$$f_j(c,t) = \left\{ \begin{array}{c} 1\ if\ WordReadIsNumber\ (w_j) = true\ and\ t_j = CD \\ else\ 0 \end{array} \right\} \tag{3.15}$$

Using the above mentioned features with MaEn another Urdu POS tagging model (henceforth MEn) is formulated. However, these suitable binary valued features are the same for other languages. This research examines some other important feature sets for the Urdu language below.

3.4.4 Morphological Information for HMM and MaEn Models:

To improve the tagging accuracy of the above models, an exclusive feature set is formulated after deep analysis of UNLTool-POS training dataset (see Section 3.4.5). This feature set is intended to have the capability to capture lexical and morphological characteristics (features) of the Urdu language. The captured morphological features are based on information retrieved from a stemmer[33] and dictionary[34], assuming that information is complete[35]. Thus, the lexical probability of assigning restricted lexical (POS) tag to a word is boosted. Consequently, the integrated models are expected to perform better with such artificial weights (reduced set of possibilities) for a given word. All the above models (T-HMM-LI, T-HMM-LaE, T-HMM-LiE, and MEn) are incorporated with such restricted POS tags features, henceforth, T-HMM-LI-MA, T-HMM-LaE-MA, T-HMM-LIE-MA and MEn-MA.

The above mentioned MA information is helpful to restrict the possible choice of POS tags for a given word, on the other hand, suffix[36] information can also help us to further improve the POS models. For HMM based POS models, suffix information has been used during the smoothing of emission probabilities. For the MEn model the suffix and prefix information are used as another type of feature. It is extended using a prefix and suffixes up to a length of four. It is also important to note, using prefix and suffixes of length $<=4$ for all words in MEn gives better results instead of using only rare words as described by [178]. The primary reason for much improved results based on prefix and suffix is that, a significant number of instances are not found for most of the word of the language vocabulary, with a small amount of annotated data. HMM based (T-HMM-LI, T-HMM-LaE, and T-HMM-LiE) and MEn models are incorporated with suffix information, shall be call them T-HMM-LI-Suf, T-HMM-LaE-Suf, T-HMM-LiE-Suf, and MEn-Suf POS taggers.

The last four POS models represent combinations of various statistical, smoothing and features as described above. The T-HMM-LI-Suf-MA is a combination of *tri-*

[33] http://www.cle.org.pk/software/langproc/UrduStemmer.htm - Last visited: 11-November-2018

[34] http://182.180.102.251:8081/oud/default.aspx - Last visited: 11-November-2018

[35] If a word is unknown then it belongs to one of the open class lexical categories, i.e. all classes of Noun, Adjective, Verb, Adverb, and Interjection.

[36] The sequence of the last few characters of a word.

gram HMM along with Linear interpolation, restricted POS tags feature and suffix information. T-HMM-LaE-Suf-MA is based on the *tri*-gram HMM model with further incorporation of Laplace smoothing, suffix and restricted POS tags. In T-HMM-LiE-Suf-MA, a *tri*-gram HMM has been used along with Lidstone's estimation, with suffix and restricted POS tags. MEn-Suf-MA POS tagging model is a collection of, MaEn, contextual window, suffix and restricted POS tags.

Table 3.12 shows an example of the Urdu text annotated with POS tags using the proposed T-HMM-Suf-MA POS tagger. As can be noted, the raw text is correctly annotated with POS tags.

Table 3.12 Example of Urdu text annotated using proposed T-HMM-Suf-MA POS tagger

Tagged Data
PU/! AUXT/ہي AUXP/رہا VBF/کھیل NNP/کرکٹ PSP/سي NN/برس Q/کافی PRP/وہ He is playing cricket for many years.
PU/۔ AUXT/ہي AUXP/رہی VBF/کر NN/فراہم NN/مواقع JJ/اچھي PSP/کي NN/نوکری NN/گورنمنٹ Government is providing good opportunities for investment.
PU/؟ VBF/کی RB/کیوں NN/حرکت PDM/یہ PSP/ني PRP/اس SC/کہ VBF/آئی NEG/نہیں NN/سمجھ I am unable to understand why he did so?
PU/۔ VBF/دیا NN/زور PSP/بھی PSP/پر NN/بات PDM/اس PSP/ني NN/اخبار Newspaper insisted on this point.
PU/۔ VBF/گزرا PSP/سي JJ/اوپر PSP/کي NN/عمارت NN/جہاز Aeroplane pass over the building.

3.4.5 Training/Testing Dataset for Urdu POS Tagging

This section describes the creation of a large dataset (hereafter called UNLTool-POS dataset) for the training and testing of the Urdu POS taggers. The dataset creation process is accomplished in three steps: (i) raw text collection, (ii) cleaning process and (iii) annotation process.

To construct a gold-standard Urdu POS tagging dataset, in the first step, a Web crawler (see Section 3.3.3) is used to extract Urdu text of 239,834 words (14,137 sentences) from various online sources (see Section 3.2.5) including BBC Urdu, Express news, Urdu library, Urdu point, Minhaj library, Awaz-e-Dost and Wikipedia. To make the dataset more realistic the raw data is from various domains: Sports (23,153), Politics (33,944), Blogs (10,976), Education (12,845), Literature (9,045), Entertainment (13,946), Science and Technology (17,683), Fashion (10,463), Weather (9,459), Business (17,328) and Commerce (10,496), Showbiz (19,503), Fictions (8,678), Health (12,783), Law (8,185), and Religion (21,347).

The raw data is pre-processed (see Section 3.2.6), which resulted in 200,000 words. The domain and genre distribution of these words is: Sports (20,128), Politics (26,145), Blogs (9,428), Education (10,742), Literature (8,756), Entertainment (10,560), Science and Technology (13,143), Fashion (9,758), Weather (8,996), Business (14,418) and Commerce (9,710), Showbiz (16,228), Fictions (8,084), Health (11,584), Law (6,952), and Religion (15,368).

The UNLTool-POS dataset was created using a manual approach. In the first step, a total of 2,000 tokens were POS tagged using the CLE online POS tagger[37] to train annotators. Manual inspection of the tagged data showed that a reasonable number of words are incorrectly tagged, particularly proper nouns, common nouns, verbs, auxiliaries, pronouns, adjectives, cardinal nominal modifiers, adverbs, conjunctions, participles, interjections and foreign fragment. In the second training step, three annotators (A, B and C) manually annotated[38] the tagged data i.e annotators A and B initially annotated same automatically annotated 2,000 tokens. An inter-annotator agreement was calculated for these tokens and conflicting tagged tokens

[37] http://182.180.102.251:8080/tag/ - Last visited: 06-August-2016

[38] In the training annotation process, the tag assigned by the CLE online POS tagger is retained if the annotator determines that it is correct, otherwise the annotator replaces it with the correct POS tag.

were discussed to further improve the annotation quality. After the training phase, the 200,000 words was manually annotated by annotators A and B and the inter-annotator agreement was computed on the entire dataset. An inter-annotator agreement of 85.7% was obtained. The Kappa Coefficient was computed to be 77.41% [53]. The conflicting tokens were annotated by the third annotator, resulting in a gold-standard UNLT-POS training/testing dataset saved in "txt" format. As far as we are aware, our UNLT-POS training/testing dataset is the largest manually POS tagged Urdu dataset, free and publicly available for research purposes.

For experiments presented in this study, the UNLTool-POST gold-standard dataset is randomly divided into two different datasets: (i) consisting of 60K training and 20K of test data (henceforth UNLTool-POS-Small training/testing dataset respectively), (ii) consisting of 120K training and 20K for testing (henceforth UNLTool-POS-Moderate training/testing dataset respectively).

The detailed statistics of different train/test datasets are shown in Table 3.13. The rows "Unknown Tokens" and "Unknown Types" of the Table 3.13 represent the count of total tokens and types (unique tokens) respectively, not seen in the different UNLTool-POS training/testing datasets. It has been observed that each test dataset holds 9% to 11% words that are unknown with respect to the training data. These figures are a little higher as compared to the several European languages [64]. However, Table 3.14 shows the detailed statistics of most frequent POS tags of the UNLTool-POS testing dataset.

3.4.6 Experimental Set-up

3.4.6.1 Datasets

For the set of experiments presented in this study, three datasets are used: UNLTool-POS-Small, UNLTool-POS-Moderate, and UNLTool-POS datasets. The purpose of

Table 3.13 Statistics of three different training/testing datasets for evaluating the performance of Urdu POS taggers

Dataset		Training set	Testing set
UNLTool-POS	Tokens	180,000	20,000
	Types	16,742	2,124
	Unknown Tokens	–	1,948
	Unknown Types	–	246
UNLTool-POS-Moderate	Tokens	120,000	20,000
	Types	14,843	2,457
	Unknown Tokens	–	2,078
	Unknown Types	–	273
UNLTool-POS-Small	Tokens	60,000	20,000
	Types	9,538	2,801
	Unknown Tokens	–	3,024
	Unknown Types	–	311

Table 3.14 Statistics of most frequent POS tags of UNLTool-POS testing dataset

POS Tag	Tokens count	Unknown tokens
NN: Common Noun	1,764	123
PSP: Postposition	1,572	0
VBF: Main Verb Finite	1,129	192
JJ: Adjective	1,315	91
AUXA: Aspectual Auxiliary	1,023	0
NNP: Proper Noun	1,243	398
RB: Common Adverb	826	63
AUXT: Tense Auxiliary	639	3

conducting experiments with three different sizes (60K, 120K, and 180K words) of the training data is to understand the relative performance of several Urdu POS tagging models as the size of training data increases.

3.4.6.2 Models

For this study, a total of 18 models are applied for Urdu POS tagging (two baseline models and sixteen other models (as described in Section 3.4.3)) as: (i) a *baseline* POS tagging model, in it each word in the test data will be assigned the POS tag based on the most frequent POS tag in the training data, (henceforth BL-MFT model), (ii) another *baseline* POS tagging model[39] [225], which uses Decision Trees along with a smoothing technique of Class Equivalence [225] (henceforth BL-CLE model), The reason for using the BL-CLE model as a baseline approach is that, currently this is the only POS tagger is available for Urdu which uses CLE Urdu POS tagset (see Section 3.4.2). Therefore, the results of CLE Urdu POS tagger can compare with the proposed UNLTool-POS tagger, (iii) T-HMM-LI model, (iv) T-HMM-LI-Suf model, (v) T-HMM-LI-MA model, (vi) T-HMM-LI-Suf-MA model, (vii) T-HMM-LaE model, (viii) T-HMM-LaE-Suf model, (ix) T-HMM-LaE-MA model, (x) T-HMM-LaE-Suf-MA model, (xi) T-HMM-LiE model, (xii) T-HMM-LiE-Suf model, (xiii) T-HMM-LiE-MA model, (xiv) T-HMM-LiE-Suf-MA model, (xv) MEn model, (xvi) MEn-Suf model, (xvii) MEn-MA model, and (xviii) MEn-Suf-MA model.

3.4.6.3 Evaluation Measures

Evaluation of the proposed Urdu POS taggers are carried out using the accuracy and standard deviation measures as before (see Section 2.5.3).

3.4.7 Results and Analysis

Table 3.15 presents the accuracy results when trained and tested on the UNLTool-POS-Small (D1), UNLTool-POS-Moderate (D2), UNLTool-POS (D3) test datasets for the Urdu POS tagging tasks by using different models (see Section 3.4.3). The standard deviations associated with the computed average accuracy has been also presented.

[39] http://182.180.102.251:8080/tag/ - Last visited: 12-November-2018

Table 3.15 Results obtained using various POS tagging models based on several approaches on different POS test datasets

Approaches∗	**Model**		**Accuracy**±σ	
		D1#	**D2#**	**D3#**
Most frequent tag	BL-MFT	-	-	84.72±0.18
Decision Tree	BL-CLE	-	-	88.45±0.16
tri-gram HMM, LI	T-HMM-LI	67.14±0.39	80.34±0.19	87.34±0.18
tri-gram HMM, LI, suffix	T-HMM-LI-Suf	83.23±0.18	87.91±0.17	91.53±0.11
tri-gram HMM, LI, MI	T-HMM-LI-MA	88.37±0.19	90.39±0.12	92.27±0.08
tri-gram HMM, LI, suffix, MI	T-HMM-LI-Suf-MA	90.87±0.12	93.76±0.07	95.14±0.05
tri-gram HMM, LaE	T-HMM-LaE	65.97±0.35	79.14±0.21	85.92±0.16
tri-gram HMM, LaE, suffix	T-HMM-LaE-Suf	80.42±0.19	86.39±0.18	89.98±0.15
tri-gram HMM, LaE, MI	T-HMM-LaE-MA	87.88±0.18	89.74±0.16	90.19±0.12
tri-gram HMM, LaE, suffix, MI	T-HMM-LaE-Suf-MA	89.04±0.16	91.64±0.13	93.74±0.08
tri-gram HMM, LiE	T-HMM-LiE	66.98±0.38	80.02±0.19	86.89±0.18
tri-gram HMM, LiE, suffix	T-HMM-LiE-Suf	82.78±0.18	87.13±0.16	90.93±0.13
tri-gram HMM, LiE, MI	T-HMM-LiE-MA	88.13±0.18	90.02±0.13	91.69±0.11
tri-gram HMM, LiE, suffix, MI	T-HMM-LiE-Suf-MA	90.23±0.13	92.59±0.11	93.97±0.07
MaEn, CW, WN	MEn	80.59±0.18	84.92±0.18	88.31±0.19
MaEn, CW, WN, suffix	MEn-Suf	84.43±0.19	88.06±0.19	92.56±0.11
MaEn, CW, WN, MI	MEn-MA	88.32±0.18	89.49±0.18	93.11±0.07
MaEn, CW, WN, suffix, MI	MEn-Suf-MA	90.26±0.12	93.31±0.07	94.20±0.06

∗: LI: Linear Interpolation, MI: Morphological Information, LaE: Laplace Estimation, LiE: Lidstone's Estimation, MaEn: Maximum Entropy, CW: Context Window, WN: Word Number, # D1: UNLTool-POS-Small training/testing dataset, D2: UNLTool-POS-Moderate training/testing dataset, D3: UNLTool-POS training/testing dataset

It can be observed that overall best results are obtained using T-HMM-LI-Suf-MA followed by MEn-Suf-MA POS tagging models 95.14% and 94.20% respectively. This shows that combining various stochastic and smoothing techniques with language dependent features are helpful in producing a very good performance on the UNLTool-POS test dataset. The highest accuracy score of 95.14% indicates that the Urdu POS tagging task is challenging and there is still room for improvement. It can also be noted from these results that a proposed POS tagging approach (T-HMM-LI-Suf-MA) outperforms both baseline approaches BL-MFT (accuracy = 84.72%) and BL-CLE (accuracy = 88.45%) on UNLTool-POS test dataset (see Figure 3.3).

It can be further observed that the *tri*-gram HMM based models can produce good results if incorporated with linear interpolation, suffix as well as Morphological Information (MI). Certainly, using MI along with linear interpolation gives better results as compared to suffix, but what is significant to note, using all information together improved the accuracy of the models, T-HMM-LI-Suf-MA: 95.14, T-HMM-LaE-Suf-MA: 93.74, and T-HMM-LiE-Suf-MA: 93.97 and MEn-Suf-MA: 94.20. Furthermore, it can be observed, T-HMM-LI, T-HMM-LaE, and T-HMM-LiE produce accuracies of 87.34%, 85.92%, and 86.89 respectively, on the UNLTool-POS dataset. For the case of MEn, the reported accuracy is 88.31%. One important observation here is that by using smoothing and language dependent features, the proposed Urdu POS tagging accuracies can be improved as compared to BL-MFT and BL-CLE models.

It can be observed from the Table 3.15, that T-HMM-LI performs better than other two models T-HMM-LaE and T-HMM-LiE, on UNLTool-POS, UNLTool-POS-Small, and UNLTool-POS-Moderate test datasets. Moreover, the accuracy of T-HMM-LaE model is slightly poorer than the other HMM based models (T-HMM-LI and T-HMM-LiE), with UNLTool-POS-Small data due to model overfitting. However,

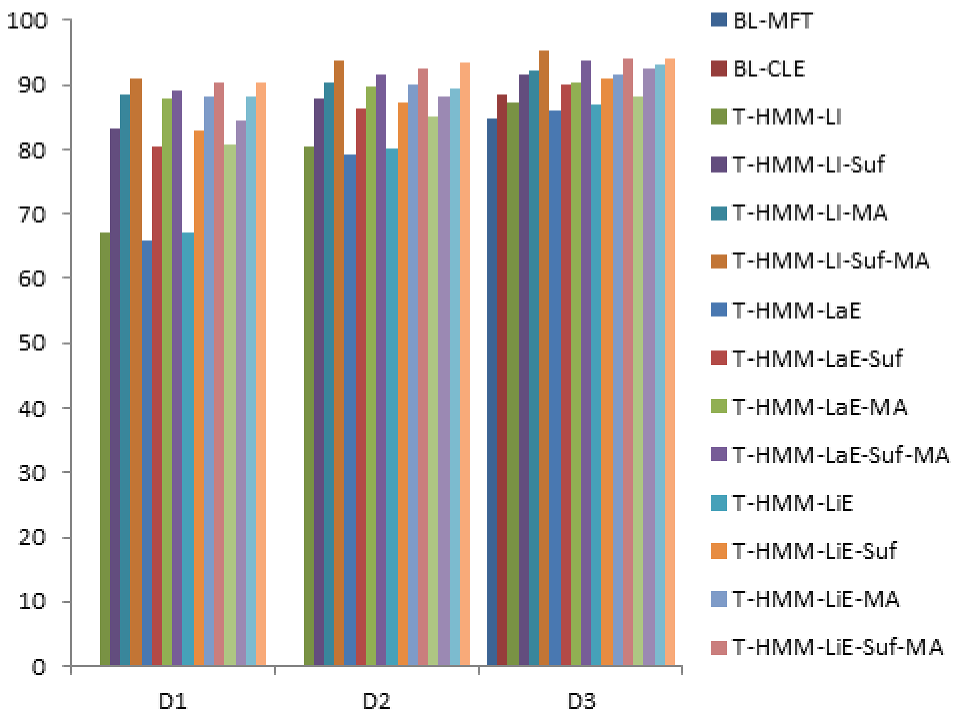


Fig. 3.3 Performance comparison of several Urdu part-of-speech taggers on different datasets

such discrepancies are alleviated with the increase of training data (UNLTool-POS-Moderate and UNLTool-POS training datasets).

It has been further observed that language dependent features increased the accuracy of the models to a certain extent, even if trained on a UNLTool-POS-Moderate training dataset. However, with different features along with smoothing, the increase in the model accuracy is higher when training data is smaller. For instance, T-HMM-LI-MA and T-HMM-LI-Suf models improved around 16%, 7% and 4%, and 21%, 10% and 5% respectively over the T-HMM-LI models, for UNLTool-POS, UNLTool-POS-Small, and UNLTool-POS-Moderate test datasets.

From the above observations, it can be concluded that using MI and suffix, increases in the model accuracy are higher for UNLTool-POS-Small and UNLTool-POS-Moderate training datasets. It is also important to note, the T-HMM-LI-MA models give an approximate improvement of around 7%, 5% and 4% over the T-HMM-LI-

Suf model for UNLTool-POS-Small, UNLTool-POS-Moderate and UNLTool-POS training dataset respectively. However, integrating all of them, an improvement has been observed in T-HMM-LI-Suf-MA models which are 5%, 3%, and 1% improved with respect to T-HMM-LI-Suf model in case of UNLTool-POS-Small, UNLTool-POS-Moderate and UNLTool-POS training dataset. It can also be noticed that similar results have been observed for the other two (T-HMM-LaE and T-HMM-LiE) HMM based models. However, T-HMM-LiE performed better than the T-HMM-LaE model, but with the higher training data, the performance of these models are somewhat comparable.

MEn models outperform all others with smaller training data but contrasting results have been observed with large training data (see Figure 3.4). It is worth noting that MEn along with suffix and morphological information has positive effects with poor resources. Results show the T-HMM-LI-Suf-MA and MEn-Suf-MA are more accurate than others, providing support for further analysis based on such models.

Table 3.16 shows cases where the MEn-Suf-MA model performs better than T-HMM-LI-Suf-MA, by comparing the accuracies of open class tags for known and unknown words on the UNLTool-POS testing dataset. Result indicate that the T-HMM-LI-Suf-MA model shows poor accuracy while predicting proper nouns (NNP) over the MEn-Suf-MA model. Mostly the proper nouns (NNP) in T-HMM-LI-Suf-MA model are erroneously classified as an adjective (JJ). Furthermore, it is worth noting again that in Urdu, there is no discrimination between upper and lower-case characters, also using an adjective as a proper noun is frequent in Urdu e.g. کبیر (KBYR, 'big') and صغیر (SGHYR, 'small'). Another reason for misclassification in tagging of the proper nouns is that many of them end with negation marker or pronoun e.g. the ناگینہ ('Nagyna') end with the نہ (NH, 'no') or the NNP ناذیہ

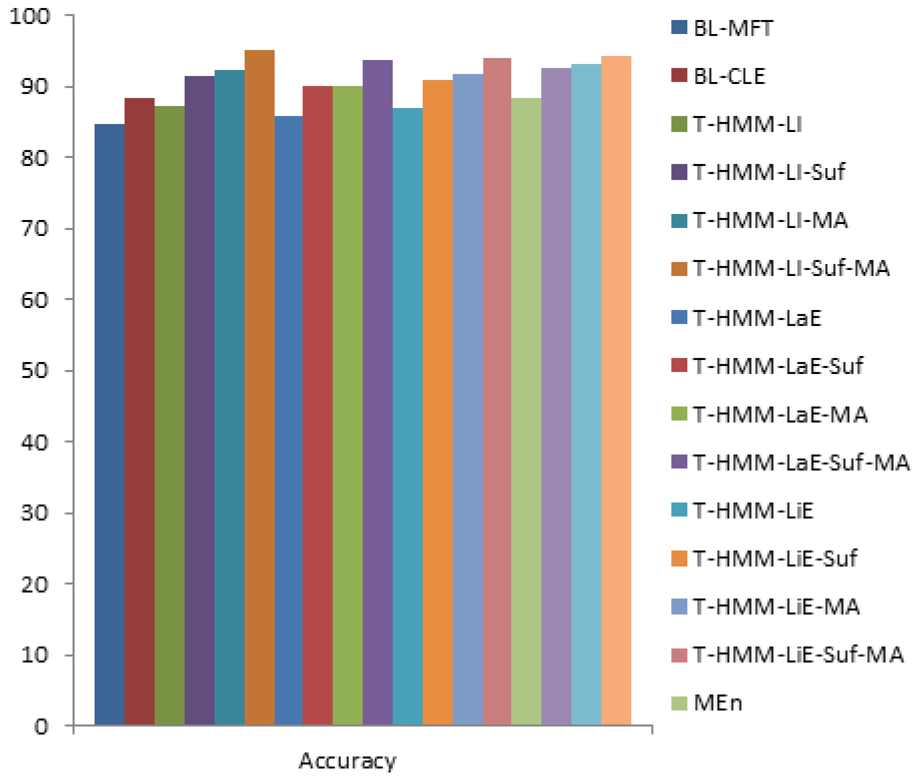


Fig. 3.4 Accuracy of various Urdu part-of-speech taggers on UNLT-POS test dataset

('Nazyh') which end with the یہ (YH, 'this'), a pronoun. These errors needs further investigation.

Table 3.16 Accuracies of open class tags on UNLTool-POS-Large testing dataset using T-HMM-LI-Suf-MA and MEn-Suf-MA

Tag	**T-HMM-LI-Suf-MA**		**MEn-Suf-MA**	
	Known	**Unknown**	**Known**	**Unknown**
NN	94.17	80.07	92.32	78.23
NNP	74.87	56.18	76.56	70.74
JJ	91.42	63.38	89.54	61.97
RB	81.78	57.71	84.45	64.33
VBF	92.93	72.67	92.47	72.03

Hence, MEn-Suf-MA is disregarded due to the lack of fine-grained POS analysis. Finally, Table 3.17 shows the confusion matrix of the T-HMM-LI-Suf-MA model by finding the most frequent confused open as well as closed class tag pairs for known and unknown words on the UNLTool-POS testing dataset. The columns and rows of the matrix represent the POS instances in the actual tags and predicted tags respectively. Only those tag pairs are considered for discussion which had more than 12 occurrences. Urdu does not use capitalised letters for upper and lower-case discrimination, which causes almost half of the NNPs erroneously tagged as common nouns (NN). In many cases, NNPs are confused with adjectives (JJ) and quantifiers (Q), particularly when NNPs are used to refer some property, quantity, feature or state in the context.

Table 3.17 Confusion matrix for most confused tags pairs on UNLTool-POS-Large testing dataset using T-HMM-LI-Suf-MA POS tagger

	Predicted Tags										
Actual tags	NN	NNP	VBI	VBF	JJ	Q	AUXA	AUXT	RB	SC	Total
NN	-	88	23	22	96	26	-	-	24	-	239
NNP	153	-	-	-	35	28	-	-	-	-	256
JJ	47	13	-	-	-	38	-	-	21	-	119
VBF	29	-	26	-	-	-	29	17	-	-	101
RB	31	-	-	-	28	-	-	-	-	23	82
Total	260	101	49	22	159	92	29	17	45	23	797

The most prominent causes for Urdu POS tag misclassification are its free word order which is difficult to classify with a coarse-grained POS tagset, and the highly inflected nature of Urdu where the grammatical categories of inflections are very closely related.

3.5 Chapter Summary

This chapter described the design, development, and evaluation of several Urdu natural language processing tools (word, sentence tokenizers and POS tagger), these tools are crucial pre-requisites for the Urdu semantic tagger. The Urdu language has a highly complex and morphological rich structure, yet it is under-resourced, and thus less advanced in terms of NLP research activities than other major world languages. It has been shown that it is possible to develop highly accurate Urdu NLP processing tools, if formulated using rule-based, dictionary look-up, n-gram language models and stochastic methods. Results have shown good performance and thus provided an evidence that these tools can be used for the Urdu semantic tagging task. In addition, a second set of contributions are the design, collection, as well as manual annotation of large Urdu datasets, and developing supporting resources.

Results showed that the proposed Urdu word tokenizer obtained precision of 96.10%, recall of 92.11%, F_1 of 94.01%, and accuracy of 97.21%. The proposed Urdu sentence tokenizer has obtained promising results (precision = 91.08%, recall = 94.14%, F_1 = 92.59%, and error rate = 6.85%). Finally, for the Urdu POS tagging task, the best accuracy (95.14%) is achieved by a tagger which is a combination of *tri*-gram HMM, linear interpolation, suffix, and morphological information.

NLP preprocessing resources (for instance, word/sentence tokenizers and POS taggers) are important for those working on computational methods to analyse and study natural languages. These resources are very much needed to help advancing the research in NLP, AI, information retrieval and for general text analysis. Here in this chapter several useful resources have been proposed and developed, that is more cheaper but of high quality. For languages which are currently under-supplied in terms of NLP resources,[40] our research study will provide a case study for the creation of useful new resources.

[40] As can be seen from the META-NET whitepaper series (http://www.meta-net.eu/whitepapers/key-results-and-cross-language-comparison - Last visited: 08-January-2020) some European languages also suffer either from weak or no support.

Chapter 4

Semantically Annotated Corpus and Multi-Target Classification Methods

4.1 Introduction

In Chapter 3 various Urdu NLP tools are proposed, and these are incorporated in the Urdu semantic tagger. However to test the performance of the semantic tagger, this chapter describes research on developing and evaluating a benchmark Urdu semantically annotated corpus. The proposed corpus contains 8,000 semi-automatically annotated tokens (2,000 each for news, social media, Wikipedia, and historical text). Each word in the corpus is assigned with one to nine semantic tags. To demonstrate how the proposed corpus can be used for the development and evaluation of supervised multi-target classification methods, a feature extraction approach is used to extract features from the proposed corpus and apply seven multi-target classifiers on them.

The remainder of the chapter is divided into four parts as follows: the first part (see Section 4.2) presents the corpus generation process. The second part (Section 4.3) explains the experimental set-up, dataset as well as semantic annotation methods (applied to the proposed corpus), evaluation measures and evaluation methodology. The third part (Section 4.4) discusses results and their analysis. Finally, the last part (Section 4.5) concludes the chapter.

4.2 Corpus Creation

In USAS (see Section 2.4) not all words fall into one predefined semantic category, rather, some words can belong to two or more semantic categories. For instance, a word "officer" can be tagged with G3/S7.1/S2, since it can be considered to belong to the semantic category "Warfare, defence and the army; Weapons" (G3), as well as to the category "Power, organizing" (S7.1), and to the category "People" (S2). These multiple memberships of categories have been indicated with "slash tag (/)" separating tags in USAS. Furthermore, USAS is a concept-driven tagging tool rather than content driven, in that it provides a general conceptual structure of the world, instead of trying to offer a semantic taxonomy for specific domains [165]. Therefore, our proposed multi-target Urdu Semantically Annotated Corpus (USA-19 Corpus) has been annotated with multiple potential semantic tags (up to nine, if required). This section describes the creation of our proposed gold standard USA-19 Corpus, including raw data collection, development of an annotation tool, annotation process, corpus statistics and standardization of the corpus.

4.2.1 Data Collection

To train and test supervised multi-target machine learning algorithms, an Urdu annotated corpus is required based on the USAS semantic taxonomy. Therefore, to develop a corpus with realistic examples, we have collected data from different domains. For example, social media texts are short and informal, whereas, newspaper articles are formally written and of moderate length. To develop the USA-19 Corpus, raw data is collected from the following domains: (i) news articles, (ii) social media (Twitter[1], Facebook[2], and Blogs), (iii) literary magazines, and (iv) Wikipedia[3] articles.

The reasons for collecting data from these domains are, firstly, they contain data which are significantly different from one another. Secondly, variation in data poses different types of challenges for the semantic annotation task, which makes our proposed corpus more realistic and challenging. Thirdly, data from these sources are free and readily available in digital format for research purposes. Fourthly, to evaluate semantic annotation tools (or methods) on a variety of writing styles and publication times. Fifthly, to make sure that our vocabulary inventory is of sufficient coverage. Finally, to produce a more robust semantic field annotated corpus.

Raw text of news articles is collected from various sources including BBC Urdu[4], Express news[5], Urdu Library[6], and Minhaj Library[7] using a Web crawler[8]. The newspaper text is useful as it is written in continuous prose and purports to be a mainly factual report of events which have taken place. The news articles collected are from different genres including Sports, Politics, Showbiz, Science and Technology, Business, Health and Religion. There are in total 2,100 word tokens in the collected text (for each genre there are 250-300 tokens). We call this sub-corpus the USA-19-raw-news corpus.

[1] https://twitter.com/ - Last visited: 11-January-2019

[2] https://facebook.com/ - Last visited: 11-January-2019

[3] https://ur.wikipedia.org/wiki/ - Last visited: 11-January-2019

[4] BBC terms of use is available at this link: https://www.bbc.com/urdu/institutional-37588278 - Last visited: 27-January-2019

[5] https://www.express.pk/ - Last visited: 11-January-2019

[6] http://www.urdulibrary.org/ - Last visited: 11-January-2019

[7] http://www.minhajbooks.com/urdu/control/ - Last visited: 11-January-2019

[8] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 11-January-2019

To form a sub-corpus from social media, raw data is collected from the following four sources: Twitter[9], Facebook[10], Blogs, and Reviews. These sources monthly serve around 2,375 million active users[11]. We manually collected publicly available data (user generated content) on different topics to make sure that the collected data is genuine, realistic, diverse and of high quality. From each source, we collected Urdu texts of 600 tokens (a total of 2,400 tokens). We call this sub-corpus the USA-19-raw-smedia corpus. It has been shown [63] that social media text poses additional challenges to automatic NLP methods, as text from these sources tends to be less grammatical. Thus, forming a corpus from social media sources provides challenging text for the Urdu semantic annotation task.

To form a third sub-corpus, Urdu text is collected from the following Wikipedia[12] articles: Culture, History, Geography and Areas, Personalities, Science and Technology. A passage of size 300-350 words is excerpted from each of these Wikipedia articles (giving a total of around 2,300 words). The sub-corpus is called USA-19-raw-wiki corpus. The reason for using Wikipedia as a text collection source is that it is large, reliable, freely available, contains texts on a variety of topics and articles written by different authors exhibiting language variation.

The last and fourth type of collected Urdu text consists of words from old Urdu literature (fiction and non-fiction short stories). Raw text of Urdu literature of early 1940s is collected from HamariWeb[13]. We collected Urdu text of approximately 2,200 words. This sub-corpus is called the USA-19-raw-historic corpus and contains Urdu text with a variety of writing styles and time periods.

[9] To address privacy issues, we asked users for their permission to use the tweets, https://twitter.com/en/privacy - Last visited, 27-January-2019

[10] Under its privacy policy we can ask Facebook users to share their data, https://www.facebook.com/about/privacy/ - Last visited: 27-January-2019.

[11] https://www.statista.com - Last visited: 11-January-2019

[12] Its terms of use are available via this link: https://foundation.wikimedia.org/wiki/Terms_of_Use/en - Last visited: 27-January-2019

[13] http://www.hamariweb.com/ - Last visited: 11-January-2019

4.2.2 Pre-processing

In this study, four different raw sub-corpora (USA-19-raw-news, USA-19-raw-smedia, USA-19-raw-wiki, and USA-19-raw-historic) have been used to form the gold standard USA-19 Corpus. All the four sub-corpora are pre-processed as follows. Text in a sub-corpus is cleaned by removing multiple spaces, duplicated text, diacritics as they are optional (only used for altering pronunciation), HTML tags, hashtags, and emoticons. Only sentences with five or more words are kept (as our empirical analysis shows that sentences with a length less than five words are typically incorrectly tagged). A language detection tool (see Section 3.2.6) has been used to discard foreign words, which resulted in the removal of 957 tokens. After pre-processing, the four cleaned sub-corpora contain raw text of 8,000 tokens (2,000 tokens in each sub-corpus).

In the next step of pre-processing, the raw text of 8,000 tokens is tokenized, lemmatized and POS tagged. The tokenization and POS tagging are carried out by using the UNLTools (see Chapter 3). UNLTools uses an Urdu POS tagset consisting of 35 tags [225]. This POS tagset is simple but based on the critical analysis of several previous iterations of Urdu POS tagset[14] (see Section 3.4.2). Furthermore, simplification of POS tagsets generally does not affect USAS semantic annotation system accuracy [165]. Lemmatization is carried out using an online Urdu tool[15]. Finally, the 8,000 tokens with automatically assigned POS tags, and lemmas are stored in txt files (called USA-19-pp-news, USA-19-pp-smedia, USA-19-pp-wiki, and USA-19-pp-historic).

[14] http://www.cle.org.pk/Downloads/langproc/UrduPOStagger/UrduPOStagset.pdf - Last visited: 11-January-2019

[15] http://lemmatization.herokuapp.com/ - Last visited: 11-January-2019

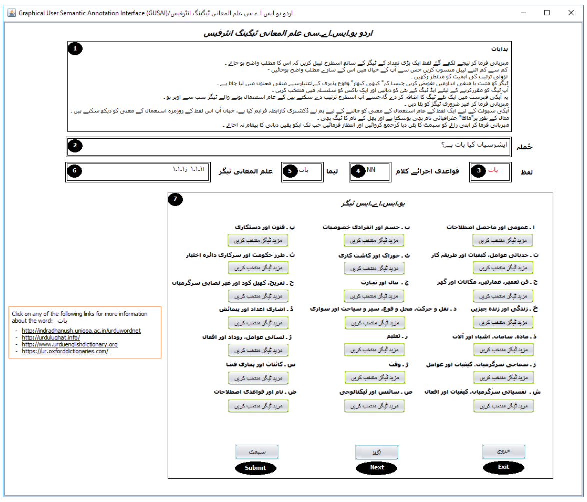
4.2.3 Annotation Tool for Urdu Semantic Annotations

To facilitate annotation of Urdu text with semantic field tags, we developed a user-friendly Java based Graphical User Semantic Annotation Interface (henceforth called GUSAI). Figure 4.1 shows the GUSAI for a sample word بات (BAT, 'Talk') (see Label 3) for the sentence اشرسیاں کیا بات ہي؟ (AYSHR SYAN KYA BAT HE?, 'Easher what's the matter?') (see Label 2) along with other information (this information has been loaded from a file, see Section 4.2.2) including POS tag (see Label 4), lemma (see Label 5), and semantic field tags[16] (see Label 6). Annotators are asked to attach as many (up to nine and at least one) USAS semantic field tag(s), as they deem appropriate for all senses of a word and place them in descending order of importance. We asked annotators to edit the POS tag, lemma, and semantic field tags(s), if the pre-assigned information is incorrect, inappropriate, or incomplete. For words whose information is missing, they must add POS tag, lemma and semantic field tag(s) information using GUSAI.

To assign semantic field tag(s) (if the assigned tag(s) is/are incomplete), an annotator needs to click on the مزید ٹگز منتخب کریں (MZYD TYGZ MNTKHB KRYN, 'add more tags') button (see Figure 5.1). Furthermore, to understand appropriate and common senses of a word, بات (BAT, 'Talk') in our case (see Figure 5.1, Label 3), the references (of dictionaries, and thesauri) are displayed alongside the GUSAI. However, annotators are free to use any other resources as they wished.

By clicking مزید ٹگز منتخب کریں (MZYD TYGZ MNTKHB KRYN, 'add more tags') button (see Figure 5.1), an annotator is directed to sub-GUSAI (see Figure 5.2) in order to attach more semantic field tag(s) (see Section 2.1) or to remove irrelevant, incorrect or inappropriate ones by selecting or deselecting the check-boxes

[16] For the process of semantic field tags assignment, a word along its POS tag information is looked up in the Urdu semantic lexicons (see Chapter 5), resulting in 7,461 semantically annotated tokens. The remaining 539 tokens which are not found in the Urdu semantic lexicons are manually annotated.



Graphical User Semantic Annotation Interface (GUSAI)/اردو یو.ایس.اے.سی علم المعانی ٹیگنگ انٹرفیس

اردو یو.ایس.اے.سی علم المعانی ٹیگنگ انٹرفیس

1

ہدایات

مہربانی فرما کر نیچے لکھے گئے لفظ ایک بڑی تعداد کے ٹیگز کے ساتھ اسطرح لیبل کریں کہ اس کا مطلب واضح ہو جائے ۔
کم سے کم اتنے لیبل منسوب کریں جس سے آپ کے خیال میں اس کے سارے مطلب واضح ہوجائیں ۔
نزولی ترتیب کی اہمیت کو مدنظر رکھیں ۔
ٹیگز کو مثبت یا منفی اندازمیں تفویض کریں جیسا کہ" کبھی کبھار" وقوع پذیری کےاعتبارسے منفی معنوں میں لیا جاتا ہے ۔
آپ ٹیگ کو مقررکرنے کے لیئے ایڈ ٹیگ کے بٹن کو دبائیں اور ایک باکس کو سلسلہ میں منتخب کریں ۔
یہ آپکی فہرست میں ایک نئے ٹیگ کا اضافہ کر دے گا،جسے آپ اسطرح ترتیب دے سکتے ہیں کے عام استعمال ہونے والے ٹیگز سب سے اوپر ہو ۔
مہربانی فرما کر غیر ضروری ٹیگز کو ہٹا دیں ۔
آپکی سہولت کے لیے ایک لفظ کے عام استعمال کے معنی کو جاننے کے لیے ہم نے ڈکشنری کارابطہ فراہم کیا ہے، جہاں آپ اس لفظ کے روزمرہ استعمال کے معنی کو دیکھ سکتے ہیں ۔
مثال کے طور پر"مالٹا" جغرافیائی نام بھی ہوسکتا ہے اور پھل کے نام کا ٹیگ بھی ۔
مہربانی فرما کر اپنی رائے کو سبمٹ کا بٹن دبا کرجمع کروائیں اور انتظار فرمائیں جب تک اپکو یقین دہانی کا پیغام نہ اجائے ۔

2 جُملہ: ایشرسیاں کیا بات ہے؟

3 لفظ: بات

4 قواعدی اجزائے کلام: NN

5 لیما: بات

6 علم المعانی ٹیگز: ا١.١.١ ز١.١.١

Click on any of the following links for more information about the word: بات

- http://indradhanush.unigoa.ac.in/urduwordnet
- http://urdulughat.info/
- http://www.urduenglishdictionary.org
- https://ur.oxforddictionaries.com/

7
یو.ایس.اے.ایس ٹیگز
ا . عمومی اور ماحصل اصطلاحات
ب . جسم اور انفرادی خصوصیات
پ . فنون اور دستکاری
ت . جذباتی عوامل، کیفیات اور طریقہ کار
ٹ . خوراک اور کاشت کاری
ث . طرز حکومت اور سرکاری دائرہ اختیار
ج . فن تعمیر، عمارتیں، مکانات اور گھر
چ . مال اور تجارت
ح . تفریح، کھیل کود اور غیر نصابی سرگرمیاں
خ . زندگی اور زندہ چیزیں
د . نقل و حرکت، محل و قوع، سیر و سیاحت اور سواری
ڈ . اشاری اعداد اور پیمائش
ذ . مادہ، سامان، اشیاء اور آلات
ر . تعلیم
ڑ . لسانی عوامل، روداد اور افعال
ز . سماجی سرگرمیاں، کیفیات اور عوامل
ژ . وقت
س . کائنات اور ہماری فضا
ش . نفسیاتی سرگرمیاں، کیفیات اور افعال
ص . سائنس اور ٹیکنالوجی
ض . نام اور قواعدی اصطلاحات
مزید ٹیگز منتخب کریں
سبمٹ
Submit
اگلا
Next
خروج
Exit

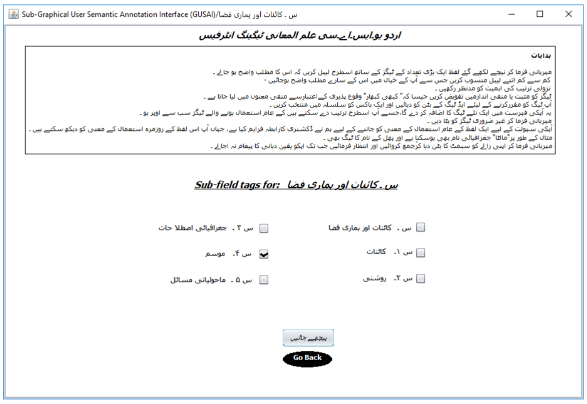
Fig. 4.1 Graphical User Semantic Annotation Interface (GUSAI) developed for the semantic annotations of our proposed USA-19 Corpus

respectively. Furthermore, by clicking *go back*, it redirects to the main-GUSAI (see Figure 5.1), where the annotator may complete the remaining (add/remove relevant/irrelevant tag(s)) annotation process. However, by clicking the *submit* button it finalizes the annotation process for a word and then stores annotated information i.e., word, POS tag, lemma, and semantic field tag(s), in persistent storage. *Next* button will load the following word along with its POS tag, lemma, and semantic field tag(s). When annotations are completed for the entire corpus, an annotator is prompted with an "annotation completion message" and (s)he can use the *Exit* button to close the annotation tool.

4.2.4 Annotation Process

Our proposed USA-19 Corpus (containing 8,000 tokens) has been semi-automatically annotated by three annotators (A, B and C). All three annotators are Urdu native speakers and had a very good understanding of the USAS semantic tagset (see Section 2.1). All the annotators are graduates, experienced in text annotations, and had a high level of proficiency in Urdu. The USA-19 Corpus has been annotated at the

Fig. 4.2 Sub-GUSAI to add/remove semantic field tag(s).



Sub-Graphical User Semantic Annotation Interface (GUSAI)/س . کائنات اور ہماری فضا

اردو یو.ایس.اے.سی علم المعانی ٹیگنگ انٹرفیس

ہدایات

مہربانی فرما کر نیچے لکھے گئے لفظ ایک بڑی تعداد کے ٹیگز کے ساتھ اسطرح لیبل کریں کہ اس کا مطلب واضح ہو جائے .
کم سے کم اتنے لیبل منسوب کریں جس سے آپ کے خیال میں اس کے سارے مطلب واضح ہوجائیں -
نزولی ترتیب کی اہمیت کو مدنظر رکھیں .
ٹیگز کو مثبت یا منفی اندازمیں تفویض کریں جیسا کہ" کبھی کبھار" وقوع پذیری کےاعتبارسے منفی معنوں میں لیا جاتا ہے .
آپ ٹیگ کو مقررکرنے کے لیئے ایڈ ٹیگ کے بٹن کو دبائیں اور ایک باکس کو سلسلہ میں منتخب کریں .
یہ آپکی فہرست میں ایک نئے ٹیگ کا اضافہ کر دے گا،جسے آپ اسطرح ترتیب دے سکتے ہیں کے عام استعمال ہونے والے ٹیگز سب سے اوپر ہو .
مہربانی فرما کر غیر ضروری ٹیگز کو ہٹا دیں .
آپکی سہولت کے لیے ایک لفظ کے عام استعمال کے معنی کو جاننے کے لیے ہم نے ڈکشنری کارابطہ فراہم کیا ہے، جہاں آپ اس لفظ کے روزمرہ استعمال کے معنی کو دیکھ سکتے ہیں .
مثال کے طور پر"مالٹا" جغرافیائی نام بھی ہوسکتا ہے اور پھل کے نام کا ٹیگ بھی .
مہربانی فرما کر اپنی رائے کو سبمٹ کا بٹن دبا کرجمع کروائیں اور انتظار فرمائیں جب تک اپکو یقین دہانی کا پیغام نہ اجائے .

Sub-field tags for: س . کائنات اور ہماری فضا

☐ س . کائنات اور ہماری فضا	☐ س ۳ . جغرافیائی اصطلا حات
☐ س ۱. کائنات	☑ س ۴. موسم
☐ س ۲. روشنی	☐ س ۵ . ماحولیاتی مسائل

word level with 21 major semantic fields and 232 sub domains of the USAS semantic tagset. The complete annotations are carried out in three phases: (i) training phase, (ii) annotations, and (iii) conflict resolving.

In the training phase, two annotators (A and B) manually annotated a subset of 62 sentences from the USA-19 Corpus using GUSAI (see Section 4.2.3). Annotators A and B discussed the annotations (both those agreed and conflicting) on the initial subset of 62 sentences to further improve the quality of annotations. After that, the remaining corpus comprising of 461 sentences are manually annotated by annotators A and B. After the annotation process, the Inter-Annotator Agreement (IAA) is computed for the entire corpus. In the third and last phase, the conflicting tokens are annotated by a third annotator (C), which resulted into a gold-standard semantically annotated corpus for Urdu language.

The Inter-Annotator Agreement (IAA) on the entire USA-19 Corpus is calculated by using three approaches: (i) first correct – check whether the first semantic field tag selected by the annotator A matches with the first semantic field tag of annotator B, (ii) fuzzy-order – check whether semantic field tags selected by an annotator A are contained within the tags annotated by B in any order, (iii) strict-order – check whether annotator A semantic field tag(s) is/are identical to B in terms of semantic field tag(s) selection and order.

On the entire USA-19 Corpus, IAA of 79.88% (first-correct) is obtained, 81.61% (fuzzy-order), and 26.56% (strict-order) (see Table 4.1). It is important to note that annotators had agreement on 6,390, 6,529, 2,125 words for first-correct, fuzzy-order, and strict-order approaches, respectively. The IAA scores of first-order and fuzzy-order are considered as good, considering the difficulty of the Urdu semantic annotation task. However, strict-order shows low IAA results (26.56%). The Kappa Coefficient [131] computed for the entire USA-19 Corpus is 77.01%, 74.96%, and 21.07% using first-correct, fuzzy-order, and strict order semantic tagging approaches, respectively.

The details of IAA for the four domain wise sub-corpora (USA-19-News, USA-19-SMedia, USA-19-Wiki, and USA-19-Historic) are also shown in Table 4.1. It shows that the highest IAA score is obtained on the USA-19-News sub-corpus using first-correct semantic tagging approach (84.65%). IAA scores of 83.76% and 81.05% are obtained for USA-19-SMedia and USA-19-Wiki sub-corpora respectively. The lowest IAA score of 70.07% is obtained for the USA-19-Historic sub-corpus. The possible reason for a low IAA score on the USA-19-Historic sub-corpus is that text in this sub-corpus is from older Urdu literature and annotators would have faced difficulty in correctly understanding the meanings of words from old Urdu. For fuzzy-order semantic tagging approach, the USA-19-News sub-corpus has obtained the highest IAA score (86.06%), followed by USA-19-Wiki (82.42%), and USA-19-SMedia (81.97%) sub-corpora. The lowest score is 75.98% for USA-19-Historic sub-corpus. Finally, for the strict-order semantic tagging approach, the highest IAA score is obtained by USA-19-News sub-corpus i.e. 31.86%. The USA-19-SMedia, USA-19-

Wiki, and USA-19-Historic sub-corpora have obtained IAA of 28.78%, 25.95%, and 19.63%, respectively.

The above discussion highlights the fact that in the case of first-order and fuzzy-order, the annotators are consistent, however for the strict-order annotators have huge variability. It also shows that the nature of text has an impact on the quality of semantic annotations as the USA-19-Historic sub-corpus obtained the lowest IAA compared to the other three sub-corpora on all three semantic tagging approaches i.e. first-order, fuzzy-order and strict-order. Finally, it is worth noting here that in the majority of cases, annotators have annotated the first tag correctly, it shows that on the most important or core tags, annotators have good IAA scores.

Table 4.1 Inter-Annotator Agreement scores for USA-19 corpus and domain wise sub-corpora.

IAA approach Corpus/Sub-corpus	First-correct	Fuzzy-order	Strict-order
USA-19	79.88%	81.61%	26.56%
USA-19-News	84.65%	86.06%	31.86%
USA-19-SMedia	83.76%	81.97%	28.78%
USA-19-Wiki	81.05%	82.42%	25.95%
USA-19-Historic	70.07%	75.98%	19.63%

4.2.5 Corpus Statistics

Table 4.2 shows the detailed statistics of the USA-19 Corpus. The gold standard USA-19 Corpus consists of 8,000 words (tokens), 2,213 unique tokens and 523 sentences. The average number of words per sentence is approximately 15. In the USA-19 Corpus, there are 2,442 nouns, 1529 verbs, 814 adjective, 636 pronouns, and 161 adverbs.

To characterize the properties of any multi-targeted Corpus (USA-19 in our case), several useful multi-label indicators have been used in the recent past [253]. The primary and natural way to measure the multi-labeledness of the entire USA-19 Corpus is label cardinality. *Label cardinality* is a standard measure to calculate the average number of tags or labels per example present in the USA-19 Corpus. For a given multi-target corpus (USA-19), the label cardinality can be computed using the following equation.

$$Label\ cardinality = \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{L}(USA-19)_k^j \tag{4.1}$$

Where, N means number of examples, and L is number of labels. If the label cardinality score is greater than 1 then it means the corpus is a multi-targeted corpus (note that when L=1 the corpus is a single-label corpus). On the other hand, a label cardinality score less than 2 means it is low multi-targeted. On proposed USA-19 Corpus, label cardinality score of 2.09 is obtained. This high number shows that proposed corpus has a good label frequency.

The USA-19 Corpus contains 16 words with typos (spelling errors), which are annotated with Foreign Fragment “FF” POS tag and “Z99” (unmatched token) semantic field tag. Note that these typos are carried inherently from sources mentioned in Section 4.2.1. Typos are not replaced with correct words because it would be interesting to see the behaviour of semantic annotation methods (see Section 2.3.1.2) on such typographical words.

4.2.6 Corpus Encoding

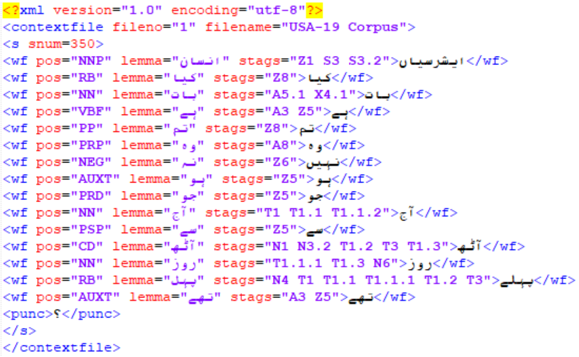
Our proposed USA-19 Corpus is encoded in XML format. Figure 4.3 shows an example of a semantically annotated sentence from the USA-19 Corpus in standard XML format. In this sentence, *<contextfile fileno=“1” filename=“USA-19”>*, indicates the beginning of a context file. The *fileno* and *filename* attributes show file number and file name, respectively. The attribute *<s snum=350>* indicates the

Table 4.2 Detailed statistics of USA-19 Corpus

Complete Urdu semantically annotated corpus	
Sentence count	434
Word count	8,000
Unique words	2,213
Words with Z99	16
Tagged words	7,477
Punctuations (untagged)	523
Semantic tags	15,624
Named entities	590
Average no of words per sentence	15
Label cardinality	2.09

beginning of a sentence, with unique IDs, i.e. *snum*. The tag <*wf pos="POS_tag" lemma="Lemma_of_Word" stags="USAS_Semantic_Tags"*>, indicates the beginning of a word in a particular sentence. The *pos* attribute shows the POS tag for a word, and *lemma* represents the lemma of a word (i.e. the dictionary head word), and *stags* shows USAS based semantic field tag(s) for a target word.

Fig. 4.3 A semantically annotated sentence in standard XML format from our proposed USA-19 Corpus.



```
<?xml version="1.0" encoding="utf-8"?>
<contextfile fileno="1" filename="USA-19 Corpus">
<s snum=350>
<wf pos="NNP" lemma="انسان" stags="Z1 S3 S3.2">ایشرسیاں</wf>
<wf pos="RB" lemma="کیا" stags="Z8">کیا</wf>
<wf pos="NN" lemma="بات" stags="A5.1 X4.1">بات</wf>
<wf pos="VBF" lemma="ہے" stags="A3 Z5">ہے</wf>
<wf pos="PP" lemma="تم" stags="Z8">تم</wf>
<wf pos="PRP" lemma="وہ" stags="A8">وہ</wf>
<wf pos="NEG" lemma="نہ" stags="Z6">نہیں</wf>
<wf pos="AUXT" lemma="ہو" stags="Z5">ہو</wf>
<wf pos="PRD" lemma="جو" stags="Z5">جو</wf>
<wf pos="NN" lemma="آج" stags="T1 T1.1 T1.1.2">آج</wf>
<wf pos="PSP" lemma="سے" stags="Z5">سے</wf>
<wf pos="CD" lemma="آٹھ" stags="N1 N3.2 T1.2 T3 T1.3">آٹھ</wf>
<wf pos="NN" lemma="روز" stags="T1.1.1 T1.3 N6">روز</wf>
<wf pos="RB" lemma="پہل" stags="N4 T1 T1.1 T1.1.1 T1.2 T3">پہلے</wf>
<wf pos="AUXT" lemma="تھے" stags="A3 Z5">تھے</wf>
<punc>؟</punc>
</s>
</contextfile>
```

4.3 Semantic Annotation Methods

In the proposed multi-target USA-19 Corpus, a tagged word can have one to nine Urdu semantic field tags associated with it. These tags have been used to indicate multiple membership categories from the USAS semantic taxonomy i.e. different components of one sense (see Section 4.2). Therefore, the Urdu semantic tagging problem is treated as a multi-target classification problem. The following sections will describe the baseline and machine learning based approaches used for the Urdu semantic tagging task, corpus, evaluation methodology and evaluation measures.

4.3.1 Approaches

4.3.1.1 Most Frequent Sense Approach

The Most Frequent Sense (MFS) heuristic is a simple but primary and the strongest baseline for any supervised semantic annotation task [202]. To handle multi-target classification, the most frequent sense has been adapted in a way that it always predicts the most frequent *set* of senses (semantic tags - up to nine tags, if available) in the entire USA-19 Corpus.

4.3.1.2 Machine Learning Approach

For this purpose, three different types of features were extracted from each input word, (i) local, (ii) topical, and (iii) semantic features.

4.3.1.2.1 Local features These are comprised of word form, POS tags– POS tags of a word itself “w_p”, for two previous words “w_{p-1}, w_{p-2}” and the next two words “w_{p+1}, w_{p+2}”. However, if there are fewer words (before or after) in the same sentence I_u, then the corresponding feature is denote as NIL, and lemmas– the lemma of a target word.

4.3.1.2.2 Topical features This consists of a bag-of-words. For each training/testing word, the vocabulary of the surrounding words can be used as feature(s). All surrounding words of a target word in the USA-19 Corpus has been used, within the same sentence. However, it has been shown [146] that this feature is position insensitive thus we use an unordered set of words based on vocabulary of the corpus and ignore the position of words. We also use a number of positional features i.e. collocations. We adopted the same 11 collocations features as cited in [42] i.e. $C_{-1,-1}$, $C_{1,1}$, $C_{-2,-2}$, $C_{2,2}$, $C_{-2,-1}$, $C_{-1,1}$, $C_{1,2}$, $C_{-3,-1}$, $C_{-2,1}$, $C_{-1,2}$, and $C_{1,3}$. Collocation $C_{j,k}$ means the ordered sequence of words and punctuation characters surrounding the target word. Furthermore, j and k refers to the starting and ending position of the sequence, respectively, a negative value refers to the word position prior to target word.

4.3.1.2.3 Semantic feature This type of feature consists of a domain indicator (cluster of texts regarding similar topics/subjects). In our case, four main domains have been used i.e. News, Social Media, Wikipedia, and Literature (see Section 4.2.1). For instance, a word ميچ (MYCH, 'match') belongs to *News* domain.

All the above mentioned extracted features (word form, POS tags, lemma, bag-of-words, collocation, and semantic) are used to train different multi-target classifiers. After extracting the local, topical and semantic set of features from the entire USA-19 Corpus, we applied seven different multi-target classifiers on them. The next section discusses these multi-target classifiers in more detail.

4.3.1.3 Multi-Target Classifiers

In contrast to single-label ML algorithms (see Section 2.3.1.2), in supervised multi-target settings, each target variable can take multiple class values. This type of classification is performed using two main approaches: (i) Problem Transformation, and (ii) Algorithm Adaptation [253, 233].

Problem Transformation is primarily used for multi-target classifiers – a multi-target problem is transformed into one or more single-label problems. Doing so, single-label ML algorithms are employed in such a way, that their single-label predictions are transformed into multi-label predictions. On the other hand, Algorithm Adaptation is an alternative to problem transformation, where internal modification is required in existing classifiers to handle multi-target data directly (off-the-shelf approaches include Decision Tree [235], MLRF (Multi-Label Random Forest) [115]). However, Algorithm Adaptation approaches are usually discipline specific, for instance, decision tree is popular in bioinformatics [187]. Consequently, problem transformation provides flexibility and scalability: any state-of-the-art single-label ML algorithms (K-Nearest Neighbour [220], etc.) can be used to suit requirements. Problem transformation can be primarily sub-classified into two categories: (i) Binary Relevance [233], and (ii) Label Combination [186] classifiers.

Binary Relevance (BR) is the most common and baseline multi-target problem transformation classifier [233]. It transforms a multi-target problem into multiple independent binary classification problems, where each binary classifier is trained to predict the relevance of one of the labels, i.e. it derives a binary training set D_j from the original multi-target training set D in the following manner:

$$D_j = \left\{ (x^i, y^i_j) | 1 \leq i \leq m \right\} \tag{4.2}$$

Each binary classification problem corresponds to one class label in the label space $Y = \{\lambda 1, \lambda 2, ..., \lambda q\}$ which contains q class labels. More precisely, each multi-target training instance (x^i, y^i) is transformed into a binary training example based on its relevance to λ_j. Where, for each j, a state-of-the-art single-label ML algorithm is employed to map a data instance to the relevance of the j_{th} label to induce a binary classifier.

There are several families of Binary Relevance classifiers in the literature each with its own pros and cons. However, an in depth study and comparison of all these classifiers is beyond the scope of this chapter. Therefore, for the Urdu semantic tagging task, the four most common and popular classifiers will be used: (i) Bayesian Classifier Chains, (ii) Classifiers Chains, (iii) Classifiers Probabilities Chains, and (iv) Class Relevance [185, 46].

Another well-known Problem Transformation approach to handle the supervised multi-target classification task is Label Combination (LC). It also transforms a multi-label problem into a multi-class problem by treating all label sets as atomic labels, that is, each label set is treated as a single label in a single-label multi-class problem. Label probability in LC can be expressed by:

$$\hat{y} = \underset{y \in Y}{argmax}\ p(y|x), \quad |Y| \ll 2^L \tag{4.3}$$

For this study, three Label Combination algorithms have been selected: (i) Nearest Set Replacement, (ii) Random -labEL Disjoint Pruned Sets (RAkELd), and (iii) Super Class Classifiers [187, 185], as these have proven to be effective in literature [231].

These multi-target classifiers have been applied in a number of research studies; text classification [60], bio-informatics [51], scene classification [37], shape detection in ultrasound images [252] etc. However, to the best of our knowledge, multi-target classifiers have never been explored for a semantic tagging task in general and particularly in the context of the Urdu language. Therefore, another contribution of this chapter is extraction of various features (see Section 4.3.1.2) from the USA-19 Corpus and the application of seven different multi-target classifiers on them.

4.3.2 Evaluation Measures

The performance of a multi-target classifier can be measured using two approaches: (i) *label-based* – evaluated on a per-label basis, and (ii) *instance-based* – used to carry out evaluation on label sets [46]. In this chapter, three evaluation measures are used to evaluate the performance of our Machine Learning based approaches: (i) Exact Match (an instance-based evaluation measure), (ii) Hamming Loss (an instance-based evaluation measure), (iii) Accuracy (a label-based evaluation measure), and standard deviation (see Section 2.5.2).

4.3.3 Corpus

For the set of experiments presented in this chapter, the entire USA-19 Corpus and its sub corpora are used (see Section 4.2.5). There are total 8,000 tokens in the USA-19 Corpus (2,000 for each of the sub corpora i.e. USA-19-News, USA-19-SMedia, USA-19-Wiki, and USA-19-Historic).

4.3.4 Evaluation Methodology

The task of Urdu semantic tagging is treated as a multi-target classification task, as one word can have one or more semantic field tags. Features extracted using local, topical and semantic approaches (see Section 4.3.1.2) are used as input to multi-target classifiers. Seven different multi-target classifiers have been applied (Bayesian Classifiers Chain, Classifier Chain, Classifier Chain Probabilities, Class Relevance, Nearest Set Replacement, Random -labEL Disjoint Pruned Sets (RAkELd), and Super Class Classifiers). To better evaluate the performance of Machine Learning based Urdu semantic tagging methods, 10-fold cross validation[17] has been applied.

[17] The *MEKA* http://waikato.github.io/meka/ [188] implementation of the multi-target classifiers, with its default parameter settings (except RAkELd – where the following parameters are selected empirically: subset size is varied from 2 to 5, number of models selected 1 to 100, and

4.4 Results and Analysis

Table 4.3 presents the Exact Match (EM), Hamming Loss (HL) and Accuracy scores obtained for Urdu semantic annotation tasks using Most Frequent Sense (MFS) and Machine Learning (ML) based approaches applied on our proposed USA-19 Corpus. The standard deviations associated with the computed multi-target evaluation masseurs have been also presented. "Classifiers" in the table refers to the Problem Transformation (PT) based Multi-target classifiers which produced the highest results among all the three single-label algorithms used in this research. "NB", and "RF" means Naïve Bayes and Random Forest, respectively. "RAkELd" is used as a short form of Random k-labEL Disjoint Pruned Sets. "BR" and "LC" refers to Binary Relevance and Label Combination which are problem transformation classifiers. The best results obtained overall are presented in bold, whereas, highest results with respect to each single-label algorithm are presented in italic.

Overall, for Hamming Loss and Accuracy evaluation measures, the best results are obtained using the Classifier Chain and RAkELd (Hamming Loss = 0.06 and Accuracy = 0.94). However, for Exact Match measure, highest scores are obtained using Nearest Set Replacement i.e. 0.77. Thus, we can say that when we consider all three evaluation measures the Classifier Chain and RAkELd (Exact Match = 0.76, Hamming Loss = 0.06 and Accuracy = 0.94) classifiers outperform all other multi-target classifiers. Also, these results are significantly higher than the baseline approach i.e. Most Frequent Sense (Accuracy = 0.52). As can be noted that very promising results are obtained for Urdu semantic annotation task indicating that the multi-target classifiers are effective in assigning semantic field tag(s) to Urdu words in our proposed corpus.

threshold is set to 0.1 to 0.9 with a 0.1 step.), is used for the supervised classification task. Furthermore, all experiments are run on a 64-bit computing machine, with 8 GB RAM.

Table 4.3 Results obtained on USA-19 Corpus using Most Frequent Sense and Machine Learning based approaches

	Classifiers		**Evaluation Measures**		
	PT based Multi-target	**Single label**	**EM±σ**	**HL±σ**	**Accuracy±σ**
Approach	**Type: Name**				
MFS					
	–	–	–	–	0.52 ± 0.11
ML					
	BR: Bayesian Classifier Chain	NB	0.58 ± 0.10	0.13 ± 0.28	0.88 ± 0.03
	BR: Classifier Chain	*NB*	0.72 ± 0.05	0.07 ± 0.53	0.93 ± 0.01
	BR: Classifier Chain Probabilities	NB	0.63 ± 0.07	0.15 ± 0.27	0.85 ± 0.04
	BR: Class Relevance	NB	0.61 ± 0.08	0.09 ± 0.47	0.91 ± 0.02
	LC: Nearest Set Replacement	NB	0.63 ± 0.07	0.10 ± 0.29	0.91 ± 0.07
	LC: RAkELd	NB	0.61 ± 0.06	0.10 ± 0.28	0.90 ± 0.07
	LC: Super Class Classifier	NB	0.63 ± 0.08	0.10 ± 0.24	0.90 ± 0.05
ML					
	BR: Bayesian Classifier Chain	RF	0.75 ± 0.06	0.07 ± 0.53	0.93 ± 0.01
	BR: Classifier Chain	***RF***	0.76 ± 0.05	0.06 ± 0.52	0.94 ± 0.01
	BR: Classifier Chain Probabilities	RF	0.75 ± 0.07	0.07 ± 0.49	0.93 ± 0.03
	LC: Class Relevance	RF	0.75 ± 0.08	0.06 ± 0.42	0.93 ± 0.02
	LC: Nearest Set Replacement	RF	0.75 ± 0.07	0.07 ± 0.56	0.94 ± 0.01
	LC: RAkELd	***RF***	0.76 ± 0.05	0.06 ± 0.49	0.94 ± 0.01
	LC: Super Class Classifier	RF	0.72 ± 0.05	0.10 ± 0.37	0.91 ± 0.02
ML					
	BR: Bayesian Classifier Chain	J48	0.59 ± 0.07	0.09 ± 0.07	0.90 ± 0.07
	BR: Classifier Chain	J48	0.72 ± 0.07	0.08 ± 0.07	0.93 ± 0.07
	BR: Classifier Chain Probabilities	J48	0.72 ± 0.07	0.08 ± 0.07	0.93 ± 0.07
	BR: Class Relevance	J48	0.59 ± 0.07	0.10 ± 0.07	0.91 ± 0.07
	LC: Nearest Set Replacement	*J48*	0.75 ± 0.07	0.06 ± 0.49	0.94 ± 0.01
	LC: RAkELd	J48	0.60 ± 0.08	0.10 ± 0.36	0.91 ± 0.03
	LC: Super Class Classifier	J48	0.76 ± 0.06	0.07 ± 0.51	0.93 ± 0.01

BR: Binary Relevance, LC: Label Combination, EM: Exact Match, HL: Hamming Loss

Among the BR and LC sub-classifiers, although best results (based on average) are obtained using Label Combination considering all three evaluation measures (Exact Match, Hamming Loss, and Accuracy), however, the difference in performance is small. The possible reason for this might be its construction style where each member of the ensemble is considered as a small random subset of labels and thus learned a single-label classier for the prediction of each element in the powerset of this subset. This highlights the fact that both BR and LC types of Problem Transformation based multi-target classifiers are effective in Urdu semantic annotations on our proposed corpus.

Regarding single-label ML algorithms (Naïve Bayes, Random Forest and J48) which are used in combination with multi-target classifiers, the best results are obtained using Random Forest on both BR (Classifier Chain) and LC (RAkELd) sub-classifiers. The possible reason for obtaining good results using Random Forest is that it is considered the best ensemble learning algorithm for the single-label classification task, thus when combined with multi-target classifiers (RAkELd and Classifier Chain) it constructs multiple single-label training sets from the multi-targeted USA-19 Corpus.

Table 4.4 presents the Exact Match (EM), Hamming Loss (HL) and Accuracy scores obtained for Urdu semantic annotation tasks using Machine Learning (ML) based approaches applied on our various sub corpora (USA-19-News, USA-19-SMedia, USA-19-Wiki, and USA-19-Historic). For the set of experiments presented here single-label Random Forest algorithm has been used (selected as this has produced better results (see Table 4.3) as compared to two others, NB and J48). All other terms of the table are same as described previously. The best average results obtained overall on the sub corpus is presented in bold, whereas, the second highest average results on sub corpus is presented in italic.

Table 4.4 Results obtained on various sub corpora using Machine Learning approaches

	PT based Multi-target Classifiers	**Evaluation Measures**		
Corpus	**Type: Name**	**EM**	**HL**	**Accuracy**
USA-19-News				
	BR: Bayesian Classifier Chain	0.71	0.08	0.93
	BR: Classifier Chain	0.71	0.08	0.93
	BR: Classifier Chain Probabilities	0.71	0.08	0.92
	BR: Class Relevance	0.71	0.07	0.92
	LC: Nearest Set Replacement	0.74	0.07	0.93
	LC: RAkELd	0.71	0.07	0.93
	LC: Super Class Classifier	0.69	0.08	0.92
	Average score of all classifiers	0.71	0.08	0.93
USA-19-SMedia				
	BR: Bayesian Classifier Chain	0.73	0.07	0.93
	BR: Classifier Chain	0.73	0.07	0.93
	BR: Classifier Chain Probabilities	0.73	0.07	0.93
	BR: Class Relevance	0.73	0.07	0.93
	LC: Nearest Set Replacement	0.74	0.07	0.94
	LC: RAkELd	0.74	0.06	0.94
	LC: Super Class Classifier	0.73	0.07	0.93
	Average score of all classifiers	*0.73*	*0.07*	*0.93*
USA-19-Wiki				
	BR: Bayesian Classifier Chain	0.72	0.08	0.92
	BR: Classifier Chain	0.72	0.08	0.92
	BR: Classifier Chain Probabilities	0.72	0.08	0.92
	BR: Class Relevance	0.72	0.08	0.92
	LC: Nearest Set Replacement	0.73	0.08	0.92
	LC: RAkELd	0.72	0.08	0.92
	LC: Super Class Classifier	0.66	0.15	0.85
	Average score of all classifiers	0.71	0.09	0.91
USA-19-Historic				
	BR: Bayesian Classifier Chain	0.78	0.06	0.94
	BR: Classifier Chain	0.78	0.06	0.94
	BR: Classifier Chain Probabilities	0.78	0.06	0.94
	BR: Class Relevance	0.78	0.06	0.94
	LC: Nearest Set Replacement	0.79	0.06	0.94
	LC: RAkELd	0.78	0.06	0.94
	LC: Super Class Classifier	0.49	0.13	0.87
	Average score of all classifiers	**0.74**	**0.07**	**0.93**

It can be observed, the best average results are obtained on USA-19-Historic sub corpus. Where the average EM, HL, and Accuracy has following scores, 0.74, 0.07 and 0.93, respectively. The lowest average results are observed for USA-19-Wiki sub corpus (EM = 0.71, HL = 0.09, and Accuracy = 0.91). Average results on USA-19-SMedia sub corpus has EM score of 0.73, HL score of 0.07, and Accuracy of 0.93. On USA-19-News sub corpus obtained average results are as, EM: 0.71 HL: 0.08, and Accuracy: 0.93 (see Figure 4.4).

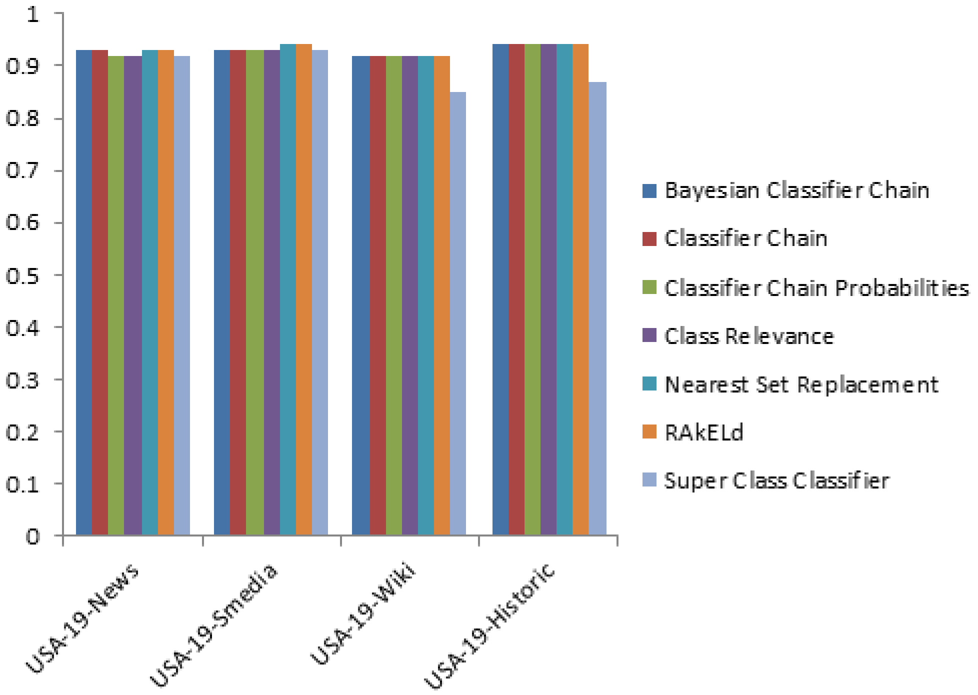


Fig. 4.4 Performance comparison of multi-target classifiers on various sub corpora

Table 4.5 presents some more detailed results (using Exact Match (EM), Hamming Loss (HL) and Accuracy scores) of local, topical and semantic features (see Section 4.3.1.2) which has been used to train and test different multi-target classifiers on the proposed USA-19 Corpus. This analysis is also based on Random Forest single-label algorithm. All others terminologies of table are same as described previously.

Table 4.5 Results obtained on the USA-19 Corpus using local, topical and semantic features

	PT based Multi-target Classifiers	**Evaluation Measures**		
Features	**Type: Name**	**EM**	**HL**	**Accuracy**
Local				
	BR: Bayesian Classifier Chain	0.70	0.12	0.88
	BR: Classifier Chain	0.70	0.12	0.88
	BR: Classifier Chain Probabilities	0.70	0.12	0.88
	BR: Class Relevance	0.70	0.12	0.88
	LC: Nearest Set Replacement	0.71	0.13	0.89
	LC: RAkELd	0.71	0.11	0.89
	LC: Super Class Classifier	0.69	0.13	0.88
	Average score of all classifiers	**0.70**	**0.12**	**0.88**
Topical				
	BR: Bayesian Classifier Chain	0.68	0.14	0.87
	BR: Classifier Chain	0.68	0.14	0.86
	BR: Classifier Chain Probabilities	0.68	0.15	0.87
	BR: Class Relevance	0.68	0.14	0.87
	LC: Nearest Set Replacement	0.68	0.14	0.87
	LC: RAkELd	0.65	0.15	0.86
	LC: Super Class Classifier	0.63	0.17	0.84
	Average score of all classifiers	*0.67*	*0.15*	*0.86*
Semantic				
	BR: Bayesian Classifier Chain	0.65	0.17	0.85
	BR: Classifier Chain	0.65	0.17	0.85
	BR: Classifier Chain Probabilities	0.65	0.17	0.85
	BR: Class Relevance	0.65	0.17	0.85
	LC: Nearest Set Replacement	0.66	0.16	0.86
	LC: RAkELd	0.66	0.15	0.86
	LC: Super Class Classifier	0.65	0.17	0.85
	Average score of all classifiers	0.65	0.17	0.85

The average results are as expected. The best average results on the USA-19 Corpus is obtained using Local features (EM = 0.70, HL = 12, and Accuracy = 88). The lowest results are obtained using Semantic feature i.e. EM = 0.65, HL = 0.17, and Accuracy = 0.85. However, the last Topical feature has also produced similar type of results i.e. EM = 0.67, HL = 0.15, and Accuracy = 0.86.

To conclude, the best results on the USA-19 Corpus are obtained using RAkELd and Classifier Chain, when considering all three evaluation measures. However, when several sub corpora are evaluated using different ML based techniques the best results are obtained for the USA-19-Historic sub-corpus, it reflects that for historic type of text multi-target classifiers are more appropriate. However, the best highest average weighted features for the USA-19 Corpus are Local whereas, the second highest feature for Urdu semantic tagging task is Topical. It also shows the semantic features are less useful for the multi-target semantic tagging task for the Urdu text.

4.5 Chapter Summary

This chapter presents a benchmark corpus for the evaluation of the US Tagger. The proposed USA-19 Corpus contains 8,000 tokens (2,000 tokens each from News, Social Media, Wikipedia, and Historic articles). Each word in the USA-19 Corpus is annotated with one to nine semantic fields tag(s) using the USAS semantic taxonomy (21 major semantic fields and 232 sub-fields). To demonstrate how the newly proposed corpus can be used for the development and evaluation of an Urdu semantic tagging method(s) another contribution of this chapter is extraction of various features (local (raw words, POS tags and lemmas), topical (bag-of-words context, bi/tri-grams collocation) and semantic (domain indicators)) from USA-19 Corpus and applied seven multi-target classifiers including Bayesian classifier chain, classifier chain, classifier chain probabilities, class relevance, nearest set replacement, RAkELd, and super class classifier. Furthermore, all sub corpora has also been evaluated separately to show which sub-corpus is bringing down the accuracy of the whole experiment. Different features for Urdu semantic tagging task have also been evaluated separately.

Results show that RAkELd and Classifier Chain multi-target classifiers outperforms all other classifiers (Hamming Loss = 0.06 and Accuracy = 0.94). Whereas, for the Exact Match measure, highest scores are obtained using Nearest Set Replacement i.e. 0.77. To conclude, results show that RAkELd and Classifier Chain multi-target classifiers outperforms all other classifiers (Exact Match = 0.76, Hamming Loss = 0.06 and Accuracy = 0.94) when combined with Random Forest single-label classifier. The USA-19-Historic sub corpus has attained highest performance (Exact Match = 0.74, Hamming Loss = 0.07, and Accuracy = 0.93). Local features for Urdu semantic tagging task are best on the USA-19 Corpus (Exact Match = 0.70, Hamming Loss = 0.12, and Accuracy = 0.88).

NLP resources and methods for the under-resourced Urdu language have been explored here as follows: (i) to prepare a gold standard corpus and (ii) the first time application of the multi-target ML classifiers for the semantic tagging task. This corpus generation process enabled the development of various tools and resources, thus providing a framework for under resourced languages to follow. Working with language-independent and state-of-the-art methods (multi-target classifiers) provides paradigms that can be applied to many languages at once.

Chapter 5

Semantic Tagset, Semantic Lexicons, Urdu Semantic Tagger and its Evaluation

5.1 Introduction

Chapter 4 presented an Urdu semantically annotated corpus and multi-target classification methods. Promising results are obtained with the multi-target classification methods on the proposed corpus (see Table 4.3). However, the proposed corpus is not used (as no semantic tagging tool is yet available) for the evaluation process of the knowledge-based Urdu semantic tagger. Therefore, this chapter describes the creation of Urdu semantic lexical resources (see Section 5.3) (that act as a knowledge source for the Urdu semantic tagger) and development as well as evaluation of the US Tagger on proposed corpus (see Section 5.4). The aim is to provide a detailed process of automatic or semi-automatic approaches which have been undertaken for the creation of Urdu semantic lexicons along with supporting resources and the US Tagger.

This chapter is divided into six parts. In the first part, the creation process of the Urdu semantic tagset has been described (see Section 5.2). The second part explains the methods used for the development of Urdu semantic lexicons (see Section 5.4). In part three (Section 5.4), similarly to the EST (see Section 2.4.1) the US Tagger has been described which has functioned as a model for proposed Urdu counter part. Part four describes semantic field disambiguation methods (see Section 5.5). The fifth part presents the experimental set-up (Section 5.6). Finally, in the part six (see Section 5.7) evaluation of the US Tagger is carried out using two benchmark corpora: (i) Urdu monolingual corpus (see Chapter 2), and (ii) USA-19 Corpus (see Chapter 4) and we discuss the insights gained from these experiments.

5.2 Creation of the Urdu Semantic Tagset

A two step semi-automatic approach is used to create the Urdu semantic tagset (see Appendix B). In the first step, each English semantic tag is looked up into two bilingual dictionaries: (i) Urdu English dictionary[1] and (ii) online lughat[2]. If both dictionaries return the same Urdu translation then that translation is selected. On the other hand, if there is a conflict in the translation then the prototypical examples as given in the USAS guidelines[3] is used to select the most suitable Urdu translation for that particular English semantic tag i.e. which matches to the nearest prototypical example meaning. Furthermore, if translations of the English prototypical examples or English tags are not found in bilingual dictionaries then machine translation services are used, Google[4] and Bing[5]. Finally, an automatically translated Urdu

[1] http://www.urduenglishdictionary.org/ - Last visited: 24-February-2019

[2] http://www.nlpd.gov.pk/lughat/index.php - Last visited: 11-February-2019

[3] http://ucrel.lancs.ac.uk/usas/usas_guide.pdf - Last visited: 11-February-2019

[4] https://translate.google.com/ - Last visited: 24-February-2019

[5] http://www.bing.com/translator - Last visited: 25-February-2019

semantic tagset is obtained. In the second step, one human expert[6] are provided with the automatically translated Urdu semantic tagset. If both human experts are agreed on a single translation then that one is selected. However, if both experts are not agreed on a single translation then the translation of the linguistics expert is preferred. This resulted in an Urdu semantic tagset (see Appendix B, the Urdu semantic tagset is also be available through the Web[7]).

5.3 Creation of Urdu Semantic Lexicons

For the development of the US Tagger, Urdu semantic lexicons are needed. In this research work, a range of automatic and semi-automatic approaches are used for the creation of Urdu semantic lexicons including mapping, crowd-sourcing, machine translation, GIZA++, word embedding and named entities. The following sections discuss these approaches in detail.

5.3.1 Mapping Approach

A two step semi-automatic approach is used to create the Urdu Semantic Lexicons (single and multi-word). For the mapping approach, existing single word and multi-word English Semantic Lexicons (ESL) are used. The single word English semantic lexicon contains 56,318 entries whereas, the multi-word English semantic lexicon has 16,871 entries. The process of Urdu semantic lexicon creation is as follows.

In the first step, each word in the English semantic lexicon is looked up in a large bi-lingual dictionary. The choice of using a appropriate bi-lingual dictionary is an important factor for this mapping approach, as inappropriate dictionary may lead to

[6] An Urdu linguistic expert teacher in Air Base Inter College Mushaf Sargodha. She has a master degree in Urdu linguistics and has been teaching since 1997. She has expertise in the USAS semantic tagset.

[7] http://ucrel.lancs.ac.uk/usas/ - Last visited: 24-February-2019

inaccurate translations, thus, it may introduce noise into the mapping process [165]. Therefore, a large En-Ur bi-lingual dictionary[8] is used, as this dictionary provides high-quality manually edited word translations. Furthermore, it provides wider lexical coverage for the Urdu language as it contains 160,897 entries. The mapping approach mainly involves transferring semantic tags from an English lexeme to its Urdu translation equivalent. For example, given a pair of word translations (one of which is English), if the English headword is found in the ESL, its semantic field tags are passed to its Urdu translation equivalents. It is worth mentioning here, this way of mapping worked quite well in this experiment, because En-Ur bilingual dictionary provides accurate translation and explicit POS tag information for most of the entries. Using this automatic mapping process 37,549 and 6,572 entries of single and multi-word ESLs respectively are translated into Urdu. Furthermore, these translated Urdu words along with POS as well as semantic tags information are stored. For those entries of ESL whose pair translation does not exist in En-Ur dictionary, such entries are deleted, resulting in a loss of 8,890, and 5,859 entries of single and multi-words of ESLs respectively.

The remaining 9,879 and 4,440 single and multi-word entries whose POS information is not contained in the EN-Ur bilingual dictionary. To make sure that none of the potential relevant semantic tags are lost, all possible POS tags of each English headword needs to be considered, and the same applies to their translation. For instance, the English headword "advance" contains four possible entries in the single-word ESL (Adjective: JJ, Singular-noun: NN1, base form of verb: VV0, and infinitive verb: VVI) with various semantic tags (N4: linear-order, A9: giving, M1: moving, coming and going, A5.1: evaluation: good/bad, Q2.2: speech acts, A2.1: affect, modify, change, S8: helping, Q2.1: speech act: communicative), although with

[8] http://www.nlpd.gov.pk/lughat/index.php - Last visited: 11-February-2019

some overlap see Table 5.1, where, first columns contain word, second contains POS tags[9], and in third USAS semantic tag(s) (see Appendix A).

Table 5.1 Various entries for the word "advance" in the USAS English semantic lexicon

Word	POS tag	Semantic tag(s)
Advance	JJ	N4
Advance	NN1	A9 M1 A5.1 A2.1
Advance	VV0	M1 A9 Q2.2 A5.1 A2.1
Advance	VVI	M1 S8 A9 A5.1 A2.1 Q2.1

For those words whose POS information in unavailable, e.g. for word "advance", each of its possible translations equivalents for the four types of POS tags and their corresponding semantic tag(s) need to be assigned to their corresponding Urdu translation. This process of mapping would lead to passing wrong and redundant semantic tags to their translation equivalents. However, such noise is bearable to increase the chances of allocating the correct semantic tags. However, in the manual annotation task (second step), it would be easier to add missing or remove redundant or irrelevant semantic tags.

The lexical resources used two different POS tagsets. ESL employed the CLAWS C7 POS tagset[10] whereas, En-Ur bi-lingual dictionary used a simplified common POS tagset[11]. To bridge this gap, CLAWS C7 and En-Ur dictionary POS tags are mapped into a common CLE Urdu POS tags [225] (consisting of 35 tags). The reason for selecting the CLE Urdu POS tagset is that it is simple but based on the critical analysis of several previous iterations of Urdu POS tagsets. Furthermore, it is written in previous literature conducted by [165] that simplification of POS tagset does not adversely affect semantic annotation accuracy. After the automatic mapping process,

[9] CLAWS C7 POS tags. http://ucrel.lancs.ac.uk/claws7tags.html - Last visited: 24-February-2019

[10] http://ucrel.lancs.ac.uk/claws/ - Last visited: 14-February-2019

[11] adjective, adverb, determiner, noun, proper noun, verb, pronoun, conjunction, interjection, preposition, particle, numeral, auxiliary, adposition

54,875 entries with the following distribution are obtained: 45,021 entries are for single-word and 9,854 for multi-word Urdu semantic lexicons.

In the second step, manual improvement of single and multi-word Urdu semantic lexicons is performed. The 1,000 most frequent words of the Urdu Monolingual Corpus[12] [92] are identified and the semantic tags of these 1,000 frequent words within the newly created Urdu semantic lexicons are manually corrected. Manual improvements of Urdu semantic lexicons are as follows: (i) filtering entries that have the wrong POS tag i.e. 274 entries are filtered-out after the POS filtering process, and (ii) selecting correct semantic tags and adding missing ones, 726 entries are edited either by adding correct or missing semantic tags. Finally, Urdu single (44,747) and multi-words (9,854) are stored in a UTF-8 txt format with following name, Ur_Map.

5.3.2 Crowdsourcing Approach

In this approach, a four step semi automatic crowdsourcing technique is used to test the wisdom of experts vs non-experts crowd for building single and multi-word Urdu semantic lexicons. Non-expert crowds are those which are unfamiliar with USAS semantic fields before the experiments took place. Expert crowds are already familiar with the USAS semantic fields in advance of the experiments. The process of Urdu semantic lexicon creation based on crowdsourcing approach is as follows.

In the first step, the 2,000 most frequent words of the British National Corpus (BNC[13]), are selected and automatically translated using Google's translation service[14], these translations are verified by three different annotators[15]. If two of the annotators are agreed on a single translation then that translation is selected. How-

[12] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 16-February-2019

[13] http://ucrel.lancs.ac.uk/bncfreq/ - Last visited: 17-February-2019

[14] https://translate.google.com/ - Last visited: 17-February-2019

[15] Two of the annotators are under-graduate NLP students, whereas the third one is a NLP lecturer. All three annotators are translation experts.

ever, if both disagree on one translation then a third annotator is asked to provide the translation of the conflicting word. The decision of the third annotator is considered as final. This resulted in 1,724 and 276 benchmark translations of single and multi-words.

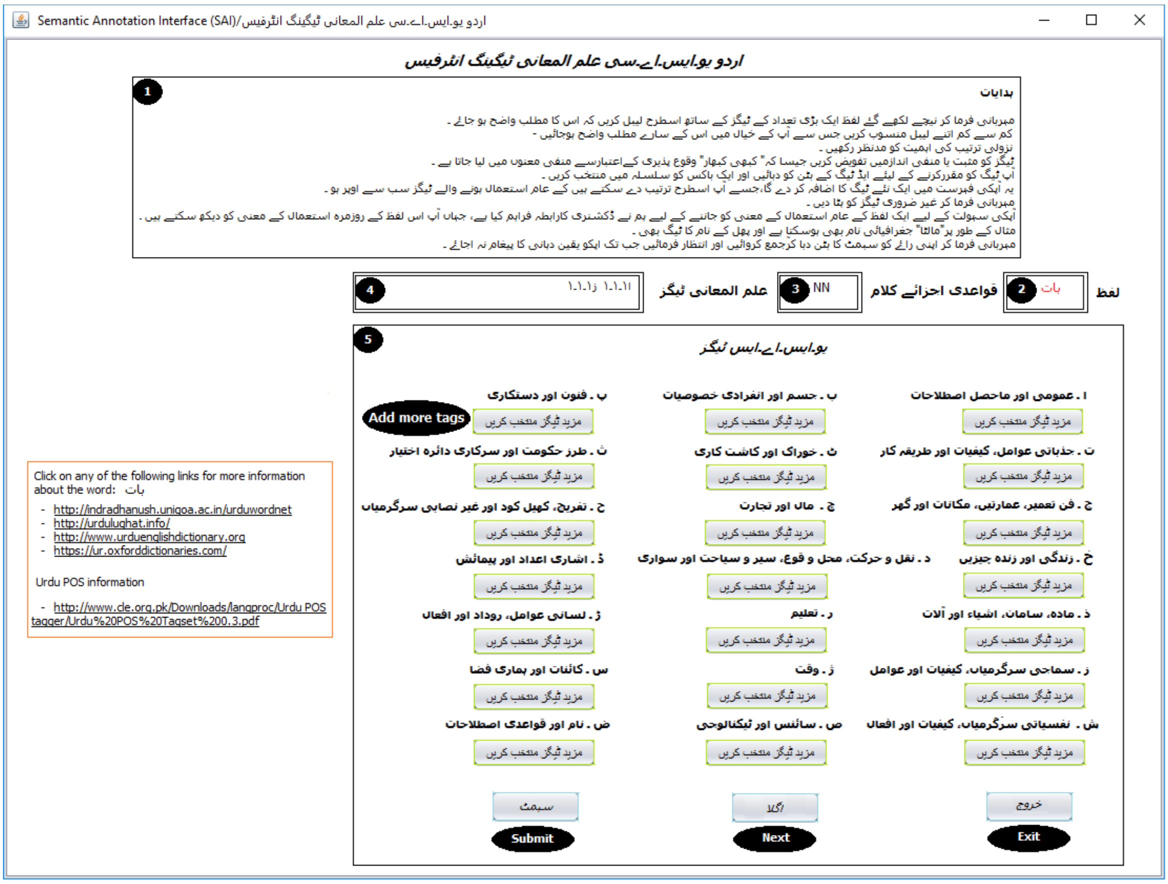
In the next step, an annotation interface is designed. As mentioned in the literature to obtain reliable results from the crowd is still a challenging task [101] that required a careful pre-selection and experimental interface for crowdsourcers. Therefore, in our research experiments, to minimize the manual effort required by participants. A user-friendly Java-based graphical user Semantic Annotation Interface (henceforth called SAI) is designed. Aside from typing the CLE Urdu POS tags [225], everything else is performed using mouse clicks to store user annotation, therefore, requiring less manual effort. More manual effort and poorly design interfaces may negatively affect the quality of annotations [99, 100].

To test the interface's ease of use, a small number of volunteer participants are asked to work through a few example words and provide feedback by answering these questions: (i) do you find the user interface easy to use (yes or no) and how easy do you find the interface (very easy, easy, moderate, difficult, very difficult), (ii) how long you took to read the instruction and complete the tasks? (in minutes, see Figure 5.1, label 1), (iii) report any error which you may have faced during the completion of the task. This information helped to improve the interface and provide more information to make the tasks efficient.

In the third step i.e. the annotation process, experts and non-experts groups are asked to label each word (in the 2,000 translated words of BNC list) presented to them using SAI (see Figure 5.1) with a number of USAS semantic fields tags (see Section 2.4.1.1). This figure shows the SAI for a sample word ("Talk") (see Label 2) along with its assigned POS tag (see Label 3), and semantic tags (see Label 4).

Annotators are asked to attach as many (up to nine), or few (at least one) USAS tag(s), as they deem appropriate for all senses of a word and place them in descending order of importance. To assign semantic field tag(s), the annotators need to click on the ("add more tags") buttons (see Figure 5.2). Furthermore, the references (of dictionaries, and thesauri) are displayed alongside the main-SAI, to understand its appropriate and common senses of a given word, however, participants are free to use any other resources as they wish. To understand CLE Urdu POS tags a link is also given, where the participant can understand POS tags and example annotated words.

Fig. 5.1 Semantic interface used in this study for annotation purpose



Semantic Annotation Interface (SAI)/اردو یو.ایس.اے.سی علم المعانی ٹیگینگ انٹرفیس

اردو یو.ایس.اے.سی علم المعانی ٹیگنگ انٹرفیس

1

ہدایات

مہربانی فرما کر نیچے لکھے گئے لفظ ایک بڑی تعداد کے ٹیگز کے ساتھ اسطرح لیبل کریں کہ اس کا مطلب واضح ہو جائے ۔
کم سے کم اتنے لیبل منسوب کریں جس سے آپ کے خیال میں اس کے سارے مطلب واضح ہوجائیں -
نزولی ترتیب کی اہمیت کو مدنظر رکھیں ۔
ٹیگز کو مثبت یا منفی اندازمیں تفویض کریں جیسا کہ" کبھی کبھار" وقوع پذیری کےاعتبارسے منفی معنوں میں لیا جاتا ہے ۔
آپ ٹیگ کو مقررکرنے کے لیئے ایڈ ٹیگ کے بٹن کو دبائیں اور ایک باکس کو سلسلہ میں منتخب کریں ۔
یہ آپکی فہرست میں ایک نئے ٹیگ کا اضافہ کر دے گا،جسے آپ اسطرح ترتیب دے سکتے ہیں کے عام استعمال ہونے والے ٹیگز سب سے اوپر ہو ۔
مہربانی فرما کر غیر ضروری ٹیگز کو ہٹا دیں ۔
آپکی سہولت کے لیے ایک لفظ کے عام استعمال کے معنی کو جاننے کے لیے ہم نے ڈکشنری کارابطہ فراہم کیا ہے، جہاں آپ اس لفظ کے روزمرہ استعمال کے معنی کو دیکھ سکتے ہیں ۔
مثال کے طور پر"مالٹا" جغرافیائی نام بھی ہوسکتا ہے اور پھل کے نام کا ٹیگ بھی ۔
مہربانی فرما کر اپنی رائے کو سبمٹ کا بٹن دبا کرجمع کروائیں اور انتظار فرمائیں جب تک اپکو یقین دہانی کا پیغام نہ اجائے ۔

لفظ 2 بات قواعدی اجزائے کلام 3 NN علم المعانی ٹیگز 4 ١.١.١١ ز١.١.١

Click on any of the following links for more information about the word: بات

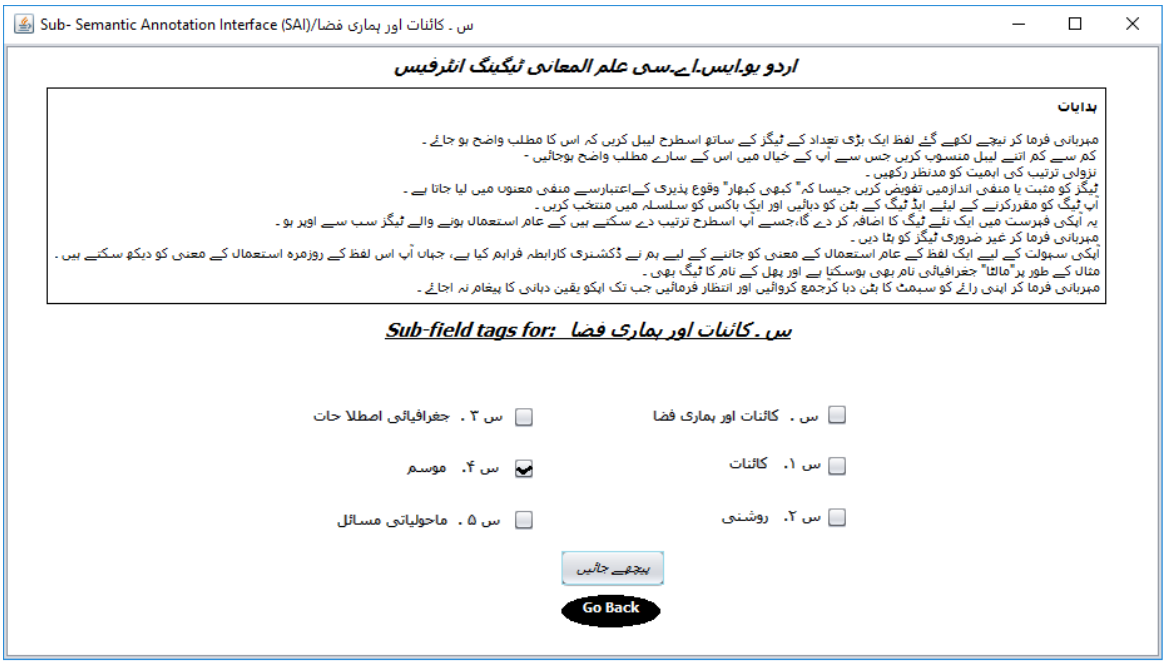
- http://indradhanush.unigoa.ac.in/urduwordnet
- http://urdulughat.info/
- http://www.urduenglishdictionary.org
- https://ur.oxforddictionaries.com/

Urdu POS information

- http://www.cle.org.pk/Downloads/langproc/Urdu POS tagger/Urdu%20POS%20Tagset%200.3.pdf

5
یو.ایس.اے.ایس ٹیگز
ا . عمومی اور ماحصل اصطلاحات
ب . جسم اور انفرادی خصوصیات
پ . فنون اور دستکاری
ت . جذباتی عوامل، کیفیات اور طریقہ کار
ٹ . خوراک اور کاشت کاری
ث . طرز حکومت اور سرکاری دائرہ اختیار
ج . فن تعمیر، عمارتیں، مکانات اور گھر
چ . مال اور تجارت
ح . تفریح، کھیل کود اور غیر نصابی سرگرمیاں
خ . زندگی اور زندہ چیزیں
د . نقل و حرکت، محل و قوع، سیر و سیاحت اور سواری
ڈ . اشاری اعداد اور پیمائش
ذ . مادہ، سامان، اشیاء اور آلات
ر . تعلیم
ڑ . لسانی عوامل، روداد اور افعال
ز . سماجی سرگرمیاں، کیفیات اور عوامل
ژ . وقت
س . کائنات اور ہماری فضا
ش . نفسیاتی سرگرمیاں، کیفیات اور افعال
ص . سائنس اور ٹیکنالوجی
ض . نام اور قواعدی اصطلاحات
مزید ٹیگز منتخب کریں
Add more tags
خروج
Exit
اگلا
Next
سبمٹ
Submit

Fig. 5.2 Sub-SAI to add or remove sub-fields semantic tag(s).



Sub- Semantic Annotation Interface (SAI)/س . کائنات اور ہماری فضا

اردو یو.ایس.اے.سی علم المعانی ٹیگنگ انٹرفیس

ہدایات

مہربانی فرما کر نیچے لکھے گئے لفظ ایک بڑی تعداد کے ٹیگز کے ساتھ اسطرح لیبل کریں کہ اس کا مطلب واضح ہو جائے ۔
کم سے کم اتنے لیبل منسوب کریں جس سے آپ کے خیال میں اس کے سارے مطلب واضح ہوجائیں -
نزولی ترتیب کی اہمیت کو مدنظر رکھیں ۔
ٹیگز کو مثبت یا منفی اندازمیں تفویض کریں جیسا کہ" کبھی کبھار" وقوع پذیری کےاعتبارسے منفی معنوں میں لیا جاتا ہے ۔
آپ ٹیگ کو مقررکرنے کے لیئے ایڈ ٹیگ کے بٹن کو دبائیں اور ایک باکس کو سلسلہ میں منتخب کریں ۔
یہ آپکی فہرست میں ایک نئے ٹیگ کا اضافہ کر دے گا،جسے آپ اسطرح ترتیب دے سکتے ہیں کے عام استعمال ہونے والے ٹیگز سب سے اوپر ہو ۔
مہربانی فرما کر غیر ضروری ٹیگز کو ہٹا دیں ۔
آپکی سہولت کے لیے ایک لفظ کے عام استعمال کے معنی کو جاننے کے لیے ہم نے ڈکشنری کارابطہ فراہم کیا ہے، جہاں آپ اس لفظ کے روزمرہ استعمال کے معنی کو دیکھ سکتے ہیں ۔
مثال کے طور پر"مالٹا" جغرافیائی نام بھی ہوسکتا ہے اور پھل کے نام کا ٹیگ بھی ۔
مہربانی فرما کر اپنی رائے کو سبمٹ کا بٹن دبا کرجمع کروائیں اور انتظار فرمائیں جب تک اپکو یقین دہانی کا پیغام نہ اجائے ۔

Sub-field tags for: س . کائنات اور ہماری فضا

☐ س . کائنات اور ہماری فضا ☐ س ۳ . جغرافیائی اصطلا حات
☐ س ۱. کائنات ☑ س ۴. موسم
☐ س ۲. روشنی ☐ س ۵ . ماحولیاتی مسائل

By clicking the "add more tags" button (see Figure 5.1), annotators are directed to sub-SAI (see Figure 5.2) in order to attach sub-field semantic tag(s) (see Section 5.1), where participants can select check-boxes. Furthermore, by clicking *go back*, it redirects to main-SAI (see Figure 5.1), where an annotator may continue with the remaining annotation process, however, by clicking the *submit* button it finalizes the annotation process for a word and then stores the annotation information i.e, word, POS tag, and semantic tags, in a persistent storage. The *Next* button will load the succeeding word along with its complete information. When the annotation is completed for 2,000 words, the participants are displayed with an end message, where annotators may use the *exit* button to end the semantic annotation process.

For each word, a total of six participants are targeted, three for each expert and non-expert participant group to allow measurement and comparison of the agreement within each group to investigate the variability of task results and participants, rather than to take a simple weighted combination to produce an agreed list.

In the last step, Urdu semantic lexicons (created by each expert or non-expert group of annotators using SAI) are evaluated using a gold-standard test lexicon[16]. It has been analysed and found that non-expert crowd results are comparable with the expert crowd in terms of accuracy[17]. However, it is found that non-experts crowd chose the correct tags but in a different order than the expert's ones. In addition to this, the majority of the non-experts participants got the first-tag (see Section 5.6.1) incorrect. This is as expected due to the fact that the Urdu language is highly inflectional and derivational, which increases ambiguity in knowing the exact sense of an out of context Urdu word as well as presenting a tough challenge to the interpretation of the words for the non-experts group. It is also worth noting that in nearly all of the semantic lexicons the expert crowds selected fewer erroneous (irrelevant) tags than the non-experts ones. Overall, accuracies show that non-expert crowd achieved comparable results to those of expert crowd when performing semantic annotation task. Thus, Urdu semantic lexicons with the highest accuracy are selected (total 4) for each expert (one single and one multi-word) and non-expert crowd (2, each one for single and multi-word).

Using a crowdsourcing approach, Urdu single and multi-word semantic lexicons are developed each of which have 1,724 and 276 entries, respectively. We named them Ur_Crowd_Ex (expert single and multi-word lexicon) and Ur_Crowd_Non-Ex (non-experts signal and multi-words) and the semantic lexicons are saved in a UTF-8 txt format.

[16] A group of three native Urdu speakers and NLP expert are asked, to manually annotate Gold-standard translations of the most frequent 2,000 words in the BNC with CLE Urdu POS and USAS semantic tags (to semantically label each word with the most suitable senses).

[17] Accuracy of the crowd selection of tags are measured by counting the matching tags between the annotator's selection and the gold standards.

5.3.3 Machine Translation Approach

A two step semi-automatic approach is used whose objective is to create Urdu semantic lexicons (single and multi-word) from the existing English semantic lexicons by translating its headwords and synonyms using a machine translation system. The single word and multi-word English semantic lexicons have 56,318 and 16,871 entries respectively. However, only half of the entries (randomly selected) of each English semantic lexicon are used in this approach in order to minimise the manual effort required in the second phase of lexicon editing process. The process of Urdu semantic lexicon generation using this approach is given below.

In the first step, each head word of the English semantic lexicon is used to generate a list of synonyms using WordNet[18] [135]. These head words along with their synonyms are used to generates translation candidates for Urdu using statistical machine translation systems (Google[19] and Bing[20]). The purpose of selecting these machine translation systems are, that they are previously used in several research studies [98, 215], and support translation for English text into Urdu. If both generate the same translation of the head word then that translation is selected. However, it has been observed that these translation systems mostly generate different translations for each of the English head word and its synonyms. Therefore, to select the correct and accurate translation a filter is used. For filtering purposes, only those translations are considered correct if their rank[21] is greater than a threshold value i.e. 0.25, which is identified through empirical analysis. Each word may have multiple candidates, so in this case, a translation candidate with a higher rank is more likely to become a correct translation in the Urdu language.

[18] http://wordnetweb.princeton.edu/perl/webwn - Last visited: 18-February-2019

[19] https://translate.google.com/ - Last visited: 18-February-2019

[20] https://www.bing.com/translator - Last visited: 18-February-2019

[21] The rank of a candidate is computed by dividing its occurrence count by the total number of translation candidates.

When machine translation systems does not generate any translation, then that word and its synonyms are skipped. After the filtration, automatically translated Urdu single and multi-words are either copied into a single or multi-words lexicon respectively, along with English head word POS tag (same POS tag has been assigned to its synonym translations) and semantic tag(s). Moreover, the English CLAWS C7 POS tags (used by the single word English semantic lexicon) are mapped into the CLE Urdu POS tags in a mapping process.

In the second step, Urdu translations are manually verified by one computational linguistics student to remove the incorrect translations, and this resulted in the removal of 8,375 entries for the Urdu single word lexicon and 2,187 entries of the Urdu multi-word semantic lexicon. However, POS and semantic tags have not been rectified in this manual process. This resulted in 39,873 and 2,098 entries for the single and multi-word semantic lexicon respectively. These lexicons are named as Ur_MT and are stored in a UTF-8 txt format.

5.3.4 GIZA++ Approach

5.3.4.1 Parallel Corpus Creation Process

In the Urdu semantic lexicon creation process, a GIZA++ approach is used (see Section 5.3.4.2) based on a sentence-aligned parallel corpus. Several parallel corpora are available in the previous literature for the Urdu language. The Urdu-Nepali-English Parallel Corpus[22] is a sentence-aligned parallel corpus, this corpus contains documents of the PENN Treebank corpus which is translated and sentence aligned for the Urdu language. Another English-Urdu parallel corpus (UMC005: English-Urdu) [93] contains English-Urdu sentence pairs of the Quran, Bible, translation of the

[22] http://www.cle.org.pk/software/ling_resources/UrduNepaliEnglishParallelCorpus.htm - Last visited: 18-February-2019

PENN Treebank documents, and (manually aligned) Enabling Minority Language Engineering (EMILLE) corpus [21]. The Indian-parallel-corpora [170] have English-Urdu parallel sentence pairs, which have been developed from Wikipedia articles using crowdsourcing.

However, all the above mentioned English-Urdu sentence-aligned parallel corpora either have licensing issues, thus, they are not always publicly available or either domain specific, which may affect the lexical coverage of Urdu semantic lexicon or of poor quality. Therefore, in this thesis, an English-Urdu Sentence Aligned Parallel Corpus (hereafter called EUSAP-19 Corpus) is developed as a supporting resource for the Urdu semantic lexicon approach. The corpus is generated using following steps: (i) raw data collection, (ii) pre-processing, (iii) annotation process, (iv), corpus statistics, and (v) corpus standardization.

In the first step, to develop a corpus with a realistic examples various Newspaper sources including The News[23], Pakistan Today[24], The Nation[25], and Tribune[26] are used to collect data with a Web crawler[27]. The newspaper text is useful as it is written in continuous prose and purports to be a mainly factual report of events which have taken place. Collected Newspaper articles are from different genres such as World, Sports, Politics, Showbiz, Technology, Business, Health, and Religion. A total of 7,875 English sentences are collected.

In the second step of the corpus creation process, the 7,875 Newspaper sentences are pre-processed as follows. The text of 7,875 sentences are cleaned by removing multiple spaces, duplicated text, HTML tags, and emoticons. Furthermore, sentences with five or more words are kept, which resulted in the removal of 657 sentences.

[23] https://www.thenews.com.pk/ - Last visited: 18-February-2019

[24] https://www.pakistantoday.com.pk/ - Last visited: 18-February-2019

[25] https://nation.com.pk/ - Last visited: 18-February-2019

[26] https://tribune.com.pk/ - Last visited: 18-February-2019

[27] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 18-February-2019

After the pre-processing, the cleaned raw text (7,218 sentences) are stored in a txt file.

In the third step, the re-processed 7,218 sentences are semi-automatically annotated by three different annotators (A, B, and C). All three annotators are Urdu native speakers but non-native English speakers and have good translation expertise. The corpus has been annotated at a sentence level. The annotators are asked to translate English sentences into Urdu using a machine translation system and then manually correct them for each English-Urdu sentence pair. The annotation is carried-out in three phases, (i) training phase, (ii) annotation, and (iii) conflict resolution.

In the training phase, two annotators (A and B) annotated a subset of 218 sentences from the 7,218 pre-processed sentences using Google and Bing (see Section 5.3.3) online machine translation tools. Annotators are asked to edit sentence translations, if the generated one is incorrect, inappropriate, or incomplete. After annotating an initial subset of 218 sentences, both annotators discussed the annotations (both agreed and conflicting pairs) to further improve the quality of annotations. In the annotation process, the remaining corpus comprising of 7K sentences are semi-automatically annotated by annotators A and B. After the annotation process, the inter-annotator agreement is computed for entire corpus is, 76.46% as annotators have agreement on 5,230 of 7,218 pairs. This score is considered good, considering the difficulty of the translating English sentences into Urdu. In the third and last phase, the conflicting sentences are annotated by a third annotator (C), which resulted in a gold-standard sentence-aligned parallel corpus for the Urdu language.

The Gold-standard EUSAP-19 Corpus is composed of 7,218 English-Urdu sentences pair. In the EUSAP-19 Corpus there are 167,573 tokens for the English language whereas, Urdu texts have 191,688 tokens. The average sentence word length for English and Urdu sentences are 23.21, and 26.56 respectively. For standardization purposes, the corpus is saved as a txt document.

5.3.4.2 Process of Creating Urdu Semantic Lexicon Using GIZA++ Tool

Several methods and tools have been proposed to align parallel texts and extract lexical correspondence from them in computational linguistics. One of these tools named GIZA++ [155, 154] is used to construct a single word Urdu semantic lexicon in this research from the parallel corpus. This tool is a freely available implementation of the IBM models for extracting word alignments. The process of creating the single word Urdu semantic lexicon is composed of two steps as follows.

In the first step, the EUSAP-19 Corpus (see Section 5.3.4.1) along with the Quran and Bible English-Urdu sentences[28] (34,403 pairs) are used. For preprocessing of the English sentences, the Natural Language ToolKit (NLTK[29]) has been used whereas, for Urdu text our Urdu natural language tools (see Chapter 3) are used, these toolkits or tools returned English and Urdu tokens (saved in separate two files) respectively. These tokenized files are given as an input to the GIZA++ tool. This tool is used with a default IBM model 4 [39] for training purpose of English-Urdu words alignment on parallel corpora. In training process of word alignment, GIZA++ tool treats every word in the English language as a possible translation for every word in the Urdu language and assigns the pairs probabilities indicating the likelihood of the translations. A word pair with higher probability can be regarded as a *correct* translation and a word pair with lower probability as an *incorrect* translation.

After the training process, GIZA++ return a lexicon (GIZA-En-Ur-Lex) of English-Urdu word alignments including a probability for each word alignment. In the GIZA-En-Ur-Lex, each English word has an average of 9 possible Urdu translations. It is

[28] http://ufal.ms.mff.cuni.cz/umc/005-en-ur/ - Last visited: 18-February-2019
[29] https://www.nltk.org/ - Last visited: 18-February-2019

cited in literature that most of these translations (with high-probability) are of good quality [144]. However, it has been observed (empirically) that there are several English-Urdu translation pairs with high-probability but with incorrect translations and vice versa for low-probability but correct translations pairs. There are several state-of-the-art methods for cleaning these statistical lexicons such as those mentioned in [10, 160]. However, due to the language constraints and poor-resourced nature of Urdu, this thesis adopted the approach cited in [114, 9] based on a filtering approach where all dictionary entries below a certain level of probability threshold value have been deleted. The threshold value for Urdu semantic lexicon filtering process is set empirically at 0.30.

The filtering process may remove several correct translation equivalents with low-probability. But such loss is bearable to decrease the manual effort required for cleaning the GIZA-En-Ur-Lex bi-lingual lexicon. Furthermore, all entries that contain invalid characters on both languages (En-Ur) are also removed. Those words with digits, symbols, punctuation markers and white-space are also deleted. After applying the filtering process, now each word has 4 possible En-Ur translation pairs.

In the next step to convert the Giza-En-Ur-Lex lexicon into an Urdu semantic lexicon. The 2K most frequent words of the BNC list are used to extract all such entries of the Giza-En-Ur-Lex lexicon which match with these words. Furthermore, these extracted translation pairs are matched with the gold-standard translations (see Section 5.3.2). This resulted in 1,285 correct translation pairs. These English words along with POS tags (manually assigned CLAWS C7 POS tags to English words) are then matched with the words and POS tags of single word English semantic lexicon. If they are identical then semantic tag(s) of the matched word is transferred into the Urdu semantic lexicon. Finally, CLAWS C7 POS tags are mapped into the CLE Urdu POS tags (see Section 5.3.1). This resulted in another single word Urdu semantic lexicon, containing 1,285 entries (Urdu word, POS tag, and semantic tag(s)), and saved in a txt file named, Ur_Giza.

5.3.5 Word Embedding Approach

Unsupervised distributed representations of words can capture important semantic and syntactic information about natural language text [79]. Traditionally, these representations can be learned by training a neural network language model [30]. Recently, a language model based on a neural network architecture has been introduced, *word embeddings* – dense real-valued feature vectors. These models have the property that similar multilingual embedding vectors are learned for similar words from a large amount of raw text during training time [133]. Word embeddings can also be induced for different languages pairs i.e. words with similar distributional semantic and syntactic properties in both languages are represented using similar vector representations. These have been demonstrated to be effective for a number of NLP tasks, for instance, document classification, bi-lingual lexicon induction, and machine-translation [239, 80, 132]. There are several off-the-shelf cross-lingual word embeddings models but for this study, the adopted model is the one cited in [80], Bilingual Bag-of-Words without Word Alignments (BilBOWA).

BilBOWA learned bilingual (English-Urdu in this case) word embeddings with a trivial extension to multilingual embeddings. Furthermore, it does not require any word or document-level alignment training data (which is not available for poor resource languages). Rather, it trained and learned directly on monolingual data (mostly available for under-resourced languages) and extracts the bilingual signal from a limited amount of sentence-aligned parallel data. Due to its simplicity and computationally-efficient characteristics, this word embedding model has been used in this research to create single word Urdu semantic lexicon.

5.3.5.1 Monolingual Corpus

BilBOWA is a data-driven model, therefore the quality of the learned word representation improved as the size of the monolingual training data improves [80]. This model learns useful features about words from raw text to predict words from the context in which they appear. Therefore, this required a large monolingual corpus. To the best of our knowledge, there exists only one large monolingual raw-text corpus for the Urdu language in the previous literature [92]. This corpus has 5.4 million sentences (95.4 million tokens). However, to produce better results, another large monolingual Urdu corpus is developed in this thesis using the following steps.

In the first phase to create another large monolingual Urdu corpus, raw text is collected from various sources (JANG[30], BBC Urdu[31], Urdu Web[32], Express news[33], Dunya[34], Daily Din[35], Urdu Library[36], Urdu Point[37], Awaz-e-Dost[38] and Wikipedia[39], Irfan-Ul-Quran[40], and King James Bible[41]) by using a Web crawler (see Section 5.3.4.1). The genres of the collected text are Commerce, Entertainment, Showbiz, Health, Weather, Science and Technology, Sports, World, Comedy, Life and Style, Politics, Blogs, Opinion, Events, Food, and Religion. The collected text consists of 4.9 million sentences (89.63 million tokens[42]).

In the next phase of the monolingual corpus creation process, the collected raw text is preprocessed. In the preprocessing step, the text is cleaned by removing

[30] https://jang.com.pk/ - Last visited: 19-February-2019
[31] http://www.bbc.com/urdu - Last visited: 19-February-2019
[32] https://www.urduweb.org/planet/ - Last visited: 19-February-2019
[33] http://www.express.pk/ - Last visited: 19-February-2019
[34] https://dunya.com.pk/ - Last visited: 19-February-2019
[35] http://www.dailydinnews.com/home - Last visited: 19-February-2019
[36] http://www.urdulibrary.org/ - Last visited: 19-February-2019
[37] http://www.urduweb.org/planet/ - Last visited: 19-February-2019
[38] http://awaz-e-dost.blogspot.co.uk/ - Last visited: 19-February-2019
[39] https://ur.wikipedia.org/wiki/ - Last visited: 19-February-2019
[40] https://www.irfan-ul-quran.com - Last visited: 19-February-2019
[41] http://www.terakalam.com/ - Last visited: 19-February-2019
[42] Tokenized using UNLTool-WT approach, see Chapter 3.

multiple spaces, duplicated text, and HTML tags. Moreover, noise from the data is removed by discarding ASCII and invalid UTF-8 characters, emoticons, white stars, bullets, right and left arrows. A language detection tool[43] is used to discard foreign words. This resulted in the removal of 1.3 million tokens. The remaining cleaned data is composed of 4.7 million sentences (88.33 million tokens using the Urdu word tokenizer, see Section 3.2). For standardisation, cleaned text is saved in a txt format as the Urdu Mono-Lingual Corpus (UMLi-19 Corpus).

5.3.5.2 Process of Creating Urdu Semantic Lexicon Using BilBOWA

A two step semi-automatic word embedding approach is used to create single word Urdu semantic lexicon as follows. In the first step, a bilingual word embedding model has been induced on the word translation task as used by Gouws et al. (see Section 5.3.5) using the parallel corpus EUSAP-19 (see Section5.3.4.1) as well as Urdu (both corpora are mentioned in Section 5.3.5.1), and English monolingual (contains 6.8 million sentences[44]) corpora. The BilBOWA used word2vec model [133] to capture the monolingual embedding with the following parameters setting, stochastic gradient descent with a default learning rate of 0.025 with linear decay, negative sampling with 5 samples, and a subsampling rate of value $1e-5$. Moreover, it is trained for 10 epochs with 200 embedding dimensions and size of the context window is set to 5. To capture bilingual embedding, the BilBOWA minimizes the sampled L_2-loss between the bag-of-word vectors of English-Urdu parallel corpus.

After the training process, each source word (English) embedding is aligned with multiple target (Urdu) induced representations. However, to create the Urdu semantic lexicon, the 2K most frequent words of the BNC list have been used (see

[43] https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0023-65A9-5 - Last visited: 19-February-2019

[44] https://sites.google.com/site/rmyeid/projects/polyglot#TOC-Download-Wikipedia-Text-Dumps - Last Visited: 19-February-2019

Section 5.3.2). For these words, the top 10 nearest neighbour bilingual embeddings are induced and distance in the embedded space is used to select word translation pairs. Only those bilingual embedding translation pairs (each source English word has 10 translations) are considered correct which are matched with gold-stranded translations (see Section 5.3.4.2). This process resulted in 760 correct translation pairs.

In the last phase of the single word Urdu semantic lexicon creation process, 760 English words are assigned CLAWS C7 POS tags by one human annotator (NLP expert). These English words along with POS tags are looked-up in the single word English semantic lexicon to assign semantic tags to each Urdu translation pair. These translated Urdu words, POS and semantic tag(s) are stored in separate txt file and given a named Ur_Bilbowa. The CLAWS C7 POS in Ur_Bilbowa semantic lexicon are mapped with CLE Urdu POS tags (see Section 5.3.1). The Ur_Bilbowa contain 760 entries (Urdu words, CLE Urdu POS and semantic tags).

5.3.6 Named Entities Approach

A three step automatic approach is used to create the Urdu Semantic Lexicons (single and multi-word). For the named entities approach, three existing different named entities lexicons (developed in another research project [122]) are used. The person named entity lexicon has 18,150 entries of single and multi-word person names. The location named entity lexicon has 18,728 location named entities of single and multi-words. The last organization named entity lexicon has 7,602 mulit-word entities of different organization names. The process of Urdu semantic lexicon creation is as follows.

In the first step, each word from the person name lexicon is automatically annotated with proper noun POS tag (NNP) and Z1 (personal names) semantic tag, which resulted in 18,150 annotated pairs. In the second step, the location name entity lexicon is used and automatically annotated with NNP POS and Z2 (geographical names) semantic tag, resulting in 18,728 multi-words. In the last step, the organization names of third lexicon are annotated with NNP POS and with semantic tag, Z3 (other proper names), resulting in 7,602 entries. All these annotated words (44,480) along with POS and semantic tags are stored in a txt file, and this named entity Urdu semantic lexicon is given a name Ur_NE.

5.4 Proposed Architecture of the Urdu Semantic Tagger

Building on the semantic lexicons (see Section 5.3), the semantic tagger for the Urdu language (Java based tool) is created, by deploying the lexicons into the software architecture (see Figure 5.3), which used a set of existing NLP tools developed at COMSATS and Lancaster Universities. These are the Urdu natural language tools (see Chapter 3) and Urdu lemmatizer[45], which respectively provide functionalities of tokenization as well as POS probabilistic annotation and lemmatization of Urdu text. These annotations are required for preprocessing the input text before the knowledge-based Urdu semantic annotation can be applied (see Section 2.4.1.4). These tools may introduce some errors in the pre-processing step, which is inevitable for automatic NLP tools.

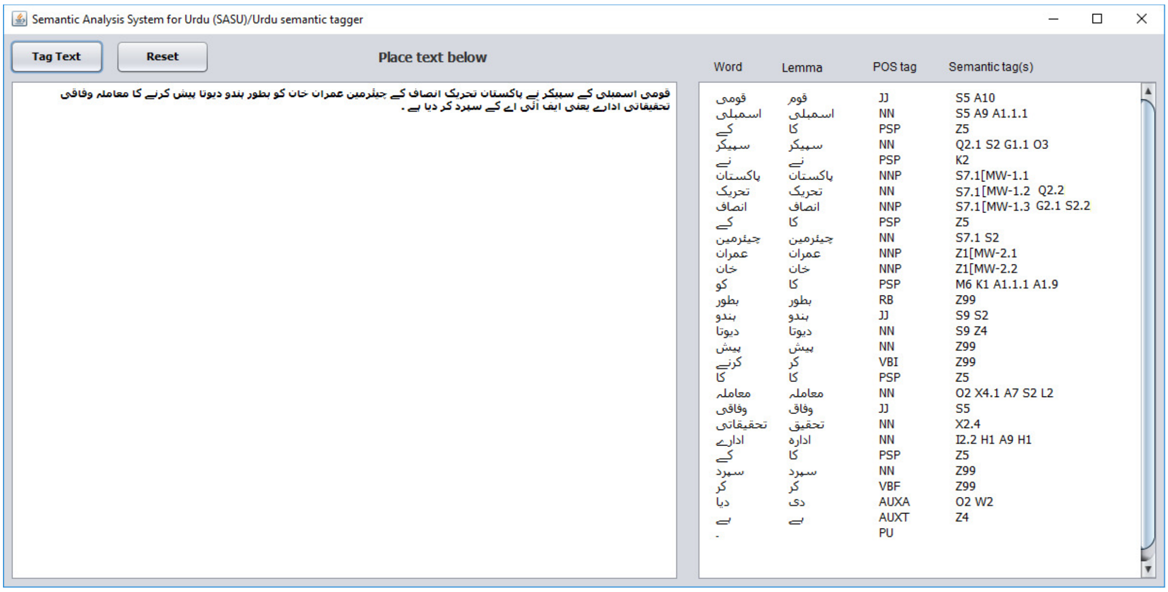
The focus of this chapter is on the performance of the US Tagger, and not to investigate the performance of the individual NLP tools, as they are reported elsewhere in the relevant paper[46] or in Chapter 3. Currently, the US Tagger produces four layers of annotations, as shown in Figure 5.3. For example, the word وفاقی (QAFI, 'federal') of sentence قومی اسمبلی كي سپيكر ني پاكستان تحريک انصاف كي چيئرمن عمران خان كو

[45] http://lemmatization.herokuapp.com/ - Last visited: 18-February-2019

[46] Sharjeel, M. et al. "Developing a Lemmatizer for the Urdu Language" *Digital Scholarship in the Humanities*, Submitted.

بطور ہندو دیوتا پیش کرني کا معاملہ وفاقی تحققاتی اداري کي سپرد کر دیا ہيي (KOMY ASMBLY KE SPYKR NE PAKSTAN THRYK ANSAF KE CHYYRMAN AMRAN KHAN KO BTOR HNDO DYOTA PYSH KRNE KA MAAMLH OFAKY THKYKATY ADARE KE SPRD KR DYA HE. ‘The speaker of national assembly has handover the case of Pakistan Tehreqe Insaf chairman Imran Khan to be as a Hindu Goddess to Federal Investigation Agency (FIA).’), the US Tagger produces, Lemma i.e. وفاق (OFAK, ‘federation’), CLE Urdu POS tag (see Section 3.4.2 of Chapter 3) "JJ" (adjective), multi-word expression (are showed with [MW-, which is not applicable for this case), and semantic tag (see Appendix A) (S5: Groups and affiliation).

Fig. 5.3 US Tagger graphical user interface

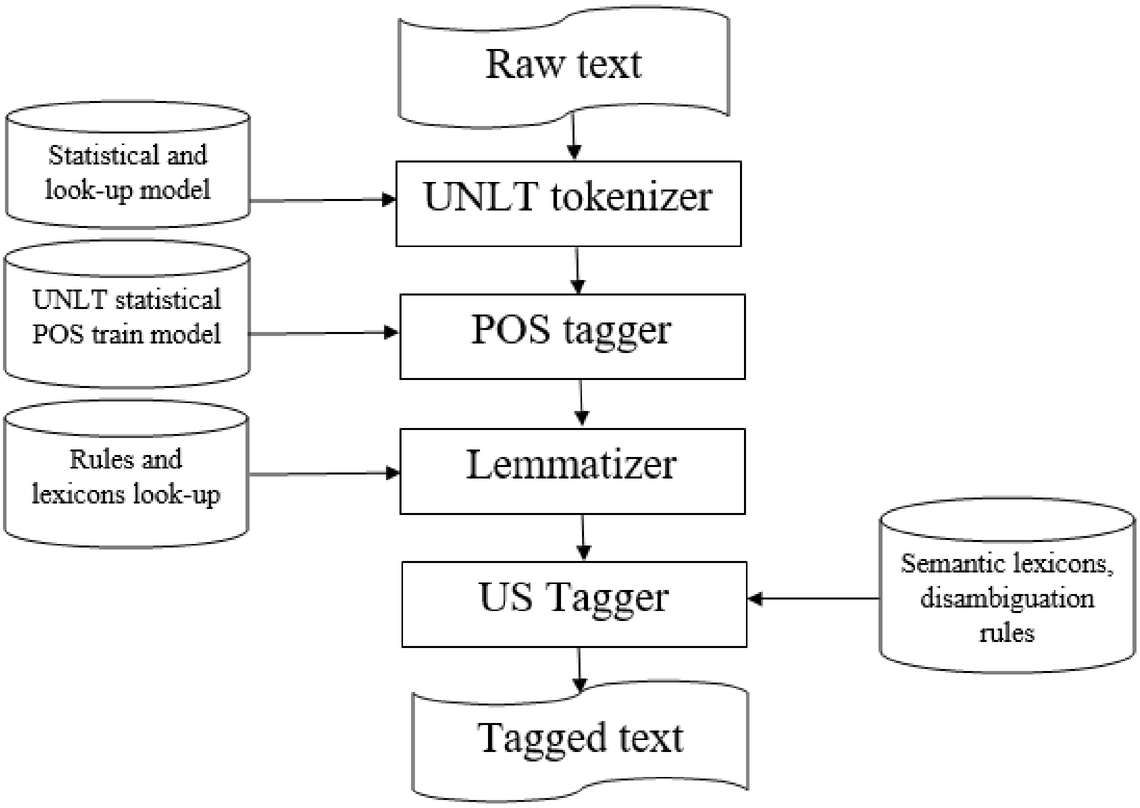

Semantic Analysis System for Urdu (SASU)/Urdu semantic tagger
Tag Text
Reset
Place text below

قومی اسمبلی کے سپیکر نے پاکستان تحریک انصاف کے چیئرمین عمران خان کو بطور ہندو دیوتا پیش کرنے کا معاملہ وفاقی تحقیقاتی ادارے یعنی ایف آئی اے کے سپرد کر دیا ہے ۔

Word	Lemma	POS tag	Semantic tag(s)
قومی	قوم	JJ	S5 A10
اسمبلی	اسمبلی	NN	S5 A9 A1.1.1
کے	کا	PSP	Z5
سپیکر	سپیکر	NN	Q2.1 S2 G1.1 O3
نے	نے	PSP	K2
پاکستان	پاکستان	NNP	S7.1[MW-1.1
تحریک	تحریک	NN	S7.1[MW-1.2 Q2.2
انصاف	انصاف	NNP	S7.1[MW-1.3 G2.1 S2.2
کے	کا	PSP	Z5
چیئرمین	چیئرمین	NN	S7.1 S2
عمران	عمران	NNP	Z1[MW-2.1
خان	خان	NNP	Z1[MW-2.2
کو	کا	PSP	M6 K1 A1.1.1 A1.9
بطور	بطور	RB	Z99
ہندو	ہندو	JJ	S9 S2
دیوتا	دیوتا	NN	S9 Z4
پیش	پیش	NN	Z99
کرنے	کر	VBI	Z99
کا	کا	PSP	Z5
معاملہ	معاملہ	NN	O2 X4.1 A7 S2 L2
وفاقی	وفاق	JJ	S5
تحقیقاتی	تحقیق	NN	X2.4
ادارے	ادارہ	NN	I2.2 H1 A9 H1
کے	کا	PSP	Z5
سپرد	سپرد	NN	Z99
کر	کر	VBF	Z99
دیا	دی	AUXA	O2 W2
ہے	ہے	AUXT	Z4
۔		PU	

Figure 5.4 illustrates the pipeline architecture of the US Tagger. The previously mentioned NLP tools and Urdu semantic lexicons form a pipeline, that are used to annotate Urdu words in the running text. The knowledge sources of the US Tagger consist of single and multi-word semantic lexicons (of different sizes) as described in Section 5.3.

Fig. 5.4 Architecture of an Urdu semantic annotation tool


Raw text
Statistical and look-up model
UNLT tokenizer
UNLT statistical POS train model
POS tagger
Rules and lexicons look-up
Lemmatizer
Semantic lexicons, disambiguation rules
US Tagger
Tagged text

5.5 Semantic Field Disambiguation Methods

The US Tagger employs a combination of two methods to contextually disambiguate which of the potential semantic tags is correct. A primary method is the grammatical category of a word, therefore the Urdu text is pre-processed with the Urdu POS tagger (see Section 3.4). For instance, the word "spring" can be partially disambiguated if it is known that either it is a verb or a noun, to differentiate semantic meanings (movement/action, (verb sense)), (metal/coil, season/water-source (common noun sense)), or (season (temporal noun sense)). By choosing the noun tags, the POS tagger can filter out the verb sense (movement/action). Hence the US Tagger task is simplified to choosing between the noun sense (metal/coil or season).

The other disambiguation method which has been employed is general likelihood ranking, derived from frequency information, past tagging experience, and intuition. In this research, a POS tagged Urdu corpus of 100K words (see Section 2.3.8) has been used to find the most frequent sense of words[47]. For instance, 'spring' referring to season is generally more frequent than 'spring' meaning metal/coil.

5.6 Experimental Set-up

This part describes an evaluation of the Urdu semantic lexicons and the US Tagger, including test data preparation and evaluation criteria, statistical results of the US Tagger performance and the impact of the disambiguation methods as currently implemented (see Section 5.5).

5.6.1 Evaluation Measures

To evaluate results of the US Tagger, two main evaluation measures are used, Lexical Coverage and Annotation Precision (see Section 2.5.1). The Lexical Coverage is a useful metric for the evaluation of Urdu semantic lexicons (see Section 5.3), since it indicates the completeness in terms of vocabulary of the semantic annotation tools.

The US Tagger annotates a word with multiple candidate semantic tags. Therefore, in addition to Lexical Coverage evaluation measure, this thesis has used two Precision metrics to indicate quality of tagged words. These are, first-correct Precision– checks whether the first semantic tag selected by the US Tagger matches with the first tag in the benchmark test corpus, and partially-correct Precision shows whether other tags selected by the US Tagger are contained within the tags of the benchmark test corpus in any order (i.e. shows correct or closely related word senses). In addition

[47] One undergraduate NLP student has manually verified different senses.

to this, standard deviation which is a common dispersion metric has also been used (see Section 2.5).

5.6.2 Test Data

For Lexical Coverage evaluation, as the test data, this thesis has used UMC monolingual corpus [91]. The choice is based upon several important requirements, it is a large and freely available benchmark corpus, which provides very recent language data thus, reflects features from the domain of News, Religion, Blogs, Literature, Science, Education and numerous others. The corpus is pre-processed and free from noise, which negatively affects the lexical coverage. Subsections of about 500K Urdu words (randomly selected from different domains) are also extracted from UMC monolingual corpus and given a name of UMC-500K test corpus. From these 500K Urdu words, the 1,000 and 2,000 most frequent words are also extracted and given a name: UMC-1K and UMC-2K test corpus respectively.

For more detailed analysis (first-correct Precision and Error rate) of the US Tagger, USA-19 Corpus (see Chapter 4) is used. Currently, this is the only large and available test corpus for the Urdu language which is semi-automatically annotated with USAS semantic tags/fields (see Appendix A). In it each word has a POS tag, thus appears with multiple possible semantic tags[48] to show multiple memberships of categories for fine-grained analysis. The USA-19 Corpus has text from Newspaper, Social Media, Wikipedia, and Historic domains, contains 8K annotated words/tokens. For the partially-correct Precision metric, the raw text of the USA-19 Corpus (see Section 4.2.2) is annotated and is manually checked.

[48] For instance, a word "officer" can be tagged with G3/S7.1/S2, since its can be considered to belong to the semantic category "Warfare, defence and the army; Weapons" (G3), as well as to the category "Power, organizing" (S7.1), and to the category "People" (S2).

5.6.3 Evaluation Methodology

The problem of Urdu semantic tagging is treated as a supervised task. Therefore, in the experiments performed, the Urdu semantic lexicons (see Section 5.3) have been used which act as a knowledge base from which to select or derive potential word level sense annotations. Urdu semantic lexicons created either automatically or semi-automatically (see Section 5.3) have different statistics. Urdu semantic lexicons which are created through mapping process Ur_Map have 44,747 (single-word) and 9,854 (multi-words) entries. The expert crowdsourcing lexicons (Ur_Crowd_Ex) have 1,724 and 276 entries, and the same number of entries exists for the non-expert crowd lexicon (Ur_Crowd_NonEx). The machine translation Urdu semantic lexicon (Ur_MT) have 39,873 and 2,098 single and multi-word entries, respectively. The Ur_Giza lexicon has 1,285 single word entries. The word embedding single word Urdu semantic lexicon (Ur_Bilbowa) has 760 entries. The named entities Urdu semantic lexicon (Ur_NE) has 44,480 entries.

To evaluate the newly developed Urdu semantic lexicons, a software tool is built i.e. the US Tagger (see Section 5.4). The US Tagger used the Urdu semantic lexicons (each experiment used different Urdu semantic lexicons however, Ur_NE is used with each Urdu semantic lexicon), a set of existing NLP tools (see Chapter 3), and semantic field disambiguation algorithms (Section 5.5) to annotate Urdu text at word level. Lexical Coverage on UMC-500K, UMC-1K, and UMC-2K test corpora (see Section 5.6.2) is calculated using the US Tagger. For more detail analysis of the US Tagger, the USA-19 Corpus (see Chapter 4) is used to evaluate lexical coverage and first-correct Precision evaluation measures (see Section 5.6.1). For partially-correct Precision (see Section 5.6.1), annotated text (see Section 5.6.2) of the US Tagger is manually verified by one human expert. Different types of error analysis related to semantic field disambiguation methods are also performed on the test USA-19

Corpus. It is important to note that punctuation marks are excluded in system evaluation process.

5.7 Results and Analysis

Table 5.2 presents the evaluation results of the US Tagger on the UMC-500K, UMC-1K, UMC-2K, and USA-19 corpora for the semantic tagging task, which employs baseline POS and general-likelihood disambiguation methods. Ur_Map, Ur_Cro_Exp, Ur_Cro_Non-Exp, Ur_MT, Ur_Giza, and Ur_Bil means that results are obtained using Urdu semantic lexicons which are developed using, mapping, crowdsourcing, machine translation, GIZA++, and word embedding approaches (see Section 5.3), respectively. TC means test corpus on which the US Tagger is evaluated. LC (Lexical Coverage) is the estimated percentage of the words in the test texts that can be tagged with the US Tagger and calculates the percentage of the words that are assigned to the meaningful semantic tags. FC (First Correct) means an evaluation of the newly build US Tagger using Precision evaluation measure to check the tagging cases where the first candidate tag is correct. PC (Partially Correct) means the evaluation (Precision) of the US Tagger in order to check the cases where the other semantic tags in the list are correct or closely related to the true word senses. The term ErrPOS in the table refers to the errors which are generated by the POS disambiguation process. Term ErrGL means the error which is produced by general-likelihood disambiguation method. The best results obtained overall are presented in bold. Whereas, the second highest results are presented in italic.

The results from Table 5.2 are as expected, overall, the best results for the US Tagger has been achieved using mapping approach based semantic lexicon (LC = 88.59 (UMC-500K), 99.63 (UMC-1K), 96.71 (UMC-2K), 89.63 (USA-19 Corpus), FC= 79.47, PC = 26.96). It demonstrates that the US Tagger obtained encouraging Lexical

Table 5.2 Evaluation results on different test corpora assessed using the US Tagger

Lexicon	TC	LC	FC	PC	ErrPOS	ErrGL
Ur_Map	UMC-500K	**88.59**	–	–	–	–
	UMC-1K	**99.63**	–	–	–	–
	UMC-2K	**96.71**	–	–	–	–
	USA-19	**89.63**	**79.47**	**26.96**	*13.56*	*38.94*
Ur_Cro_Exp	UMC-500K	21.17	–	–	–	–
	UMC-1K	81.63	–	–	–	–
	UMC-2K	73.87	–	–	–	–
	USA-19	41.35	71.13	20.47	**8.42**	**30.96**
Ur_Cro_Non-Exp	UMC-500K	18.62	–	–	–	–
	UMC-1K	79.65	–	–	–	–
	UMC-2K	67.25	–	–	–	–
	USA-19	34.57	69.87	19.26	15.63	39.94
Ur_MT	UMC-500K	*83.41*	–	–	–	–
	UMC-1K	*96.39*	–	–	–	–
	UMC-2K	*91.47*	–	–	–	–
	USA-19	*81.94*	*76.69*	15.26	14.98	41.96
Ur_Giza	UMC-500K	21.86	–	–	–	–
	UMC-1K	68.64	–	–	–	–
	UMC-2K	59.13	–	–	–	–
	USA-19	46.36	69.13	14.07	10.78	34.09
Ur_Bil	UMC-500K	14.53	–	–	–	–
	UMC-1K	41.44	–	–	–	–
	UMC-2K	27.08	–	–	–	–
	USA-19	21.13	63.74	11.53	10.98	32.38

TC: Test Corpus, LC: Lexical Coverage, FC: First Correct, PC: Partially Correct

Coverage and Precision for Urdu text when tested with the Ur_Map semantic lexicon. It also shows that the US Tagger has stable Lexical Coverage on different types of text. Furthermore, Lexical Coverage on UMC-1K and UMC-2K is also encouraging, that can help us to identify the practical usefulness of the US Tagger for general language analysis. However, after applying various semantic field disambiguation methods, overall best results are achieved for Ur_Cro_Exp semantic lexicon i.e. ErrPOS= 8.42 and ErrGL= 30.96. Such type of error analysis helps to identify error occurrences as well as used to improve the accuracy of the tool. For all other Urdu semantic lexicons, the same pattern of differences in the result has been observed (see Figure 5.5).

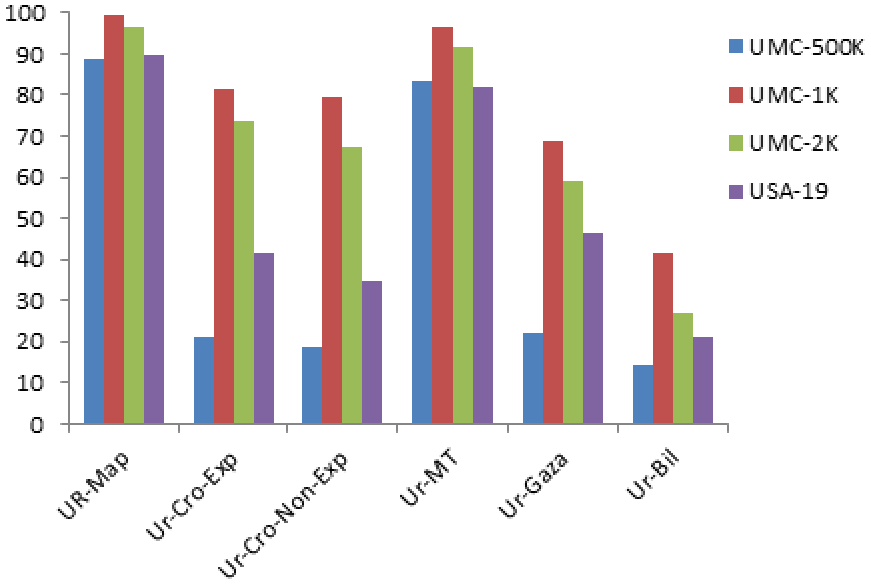


Fig. 5.5 Lexical coverage of Urdu semantic lexicons on several test corpora

The performance of the US Tagger using word embedding based semantic lexicon (Ur_Bil) (LC = 14.53 (UMC-500K), 41.44 (UMC-1K), 27.08 (UMC-2K), 21.13 (USA-19 Corpus), FC = 63.74, PC = 11.53, ErrPOS = 10.98, ErrGL = 12.38), shows the lowest results. The main reason of such low lexical coverage is the size of Urdu semantic lexicon used in the experiment (760 entries) as the cross-lingual word embedding technique generates accuracy of 55% [80]. However, it is worth mentioning here that with such small semantic lexicon the results are still comparable.

Integration of crowd sourced Urdu semantic lexicons into the US Tagger produced reasonable results. Where the annotation tool using Ur_Cro_Exp generates the following results: LC = 21.17 (UMC-500K), 81.63 (UMC-1K), 73.87 (UMC-2K), 41.35 (USA-19 Corpus), FC = 71.13, PC = 20.47, ErrPOS = 8.42, ErrGL = 10.96. Ur_Cro_Non_Exp generates an almost similar pattern of results (LC = 18.62 (UMC-500K), 79.65 (UMC-1K), 67.25 (UMC-2K), 34.57 (USA-19 Corpus), FC = 69.87, PC =

19.26, ErrPOS = 15.63, ErrGL = 19.94). This demonstrates that the untrained crowd can produce results that are comparable to those of expert annotators.

The results using the Ur_MT semantic lexicon, achieve Lexical Coverage on UMC-500, 1K, and 2K of 83,41, 96.39, and 91.47, respectively. Lexical Coverage on USA-19 Corpus is 81.94. The Precision using two different metrics FC and PC is 76.69 and 15.26, respectively (see Figure 5.6). The ErrPOS and ErrGL produce an error of 10.78 and 14.09, respectively on the USA-19 test corpus.

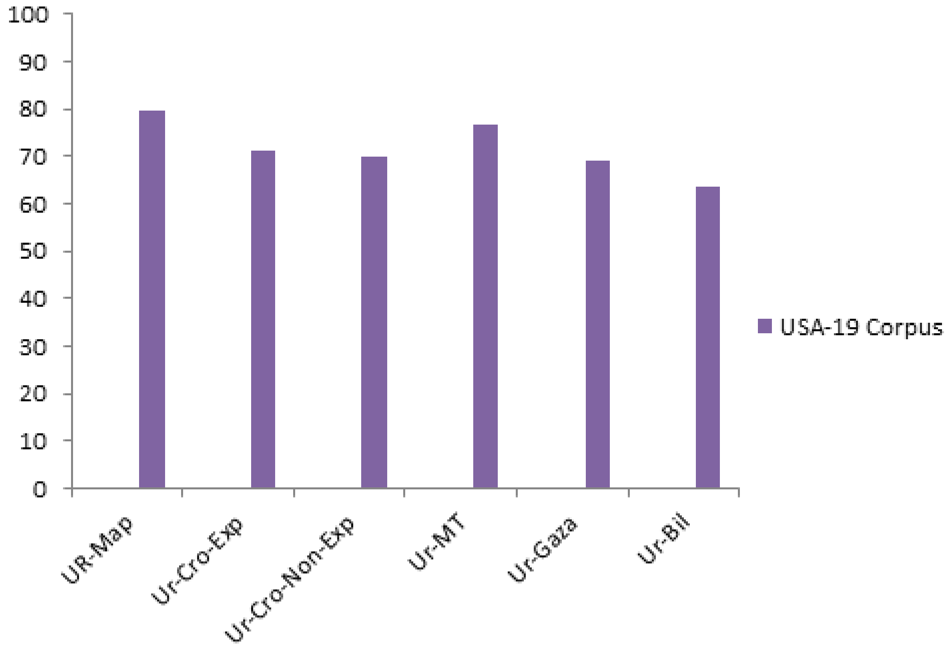


Fig. 5.6 Precision of the US Tagger using several Urdu semantic lexicons on the USA-19 test corpus

The performance of the US Tagger using Ur_Giza is as follows: Lexical Coverage on UMC-500, UMC-1K, UMC-2K, and USA-19 Corpora are 21.86, 68.64, 59.13, 46.36 respectively. FC gives a score of 63.74, whereas, PC scores 11.53. The error rate is 10.98 and 12.38 for POS and general likelihood disambiguation methods respectively.

Table 5.3 provides the final comprehensive results of the US Tagger when all previously mentioned Urdu semantic lexicons (Ur_Map, Ur_Cro_Exp, Ur_Cro_Non-Exp, Ur_MT, Ur_Giza, and Ur_Bil) have been merged into a single lexicon i.e. Ur_Merged (all other terms of this table are same as mentioned previously). It can be seen that Lexical Coverage on UMC-500K, UMC-1K, UMC-2K (most frequent words[49]) and USA-19 Corpora are 91.37%, 99.89%, 98.01%, 90.37%, respectively. The Precision obtained on the USA-19 Corpus based on FC and PC factors are 80.97% and 27.37% respectively. ErrPOS and ErrGL on an annotated test text are 18.91% and 42.06% respectively.

Table 5.3 Evaluation results on various test corpora assessed using the US Tagger when all Urdu semantic lexicons are merged

Lexicon	TC	LC	FC	PC	ErrPOS	ErrGL
Ur_Merged	UMC-500K	91.37	–	–	–	–
	UMC-1K	99.89	–	–	–	–
	UMC-2K	98.01	–	–	–	–
	USA-19	90.37	80.97	27.37	18.91	42.06

TC: Test Corpus, LC: Lexical Coverage, FC: First Correct, PC: Partially Correct

Given that the US Tagger and lexicons are built over a short period of time, such Lexical Coverage and Precision is highly encouraging. However, the lower Precision of partially correct tags scores is expected due to the fact that Urdu is highly inflectional and derivational, which increases ambiguity and presents challenges to the interpretations of the words. It is also important to note that the general-likelihood disambiguation methods are not appropriate for the Urdu semantic disambiguation task. It can be stated that the proposed approach to developing a prototype semantic annotation tool using rapidly generated semantic lexicons can be expected to achieve stable results, and thus, need significant expansion. It is worth mentioning here, although the Precision is still low and errors are high, however, the US Tagger is starting to approach the precision of USAS English semantic system at 91% and error rate at 8.95% [180].

[49] The lexical coverage of the frequent words can help to assess the practical usefulness of the Urdu semantic lexicons for general language analysis.

Finally, to estimate the reliability of evaluation results, the test data of USA-19 Corpus has been further divided into four sub-divisions, USA-19-News, USA-19-SMedia, USA-19-Wiki, and USA-19-Historic (see Section 4.3.3). For each of the sub-divisions, lexical coverage and the standard deviation score have been calculated, so that if the LC (Lexical Coverage) of the individual sub-divisions close to each other, or have a small statistical variation score, then it would indicate that the US Tagger and Urdu semantic lexicons have stable LC on different types of text and vice versa. Table 5.4 shows the lexical coverage (LC) of the each sub-division and the Standard Deviation (σ) scores. It can be observed from the table, the lexical coverage achieved small variation scores (0.06), which indicates that our Urdu semantic lexicons have rather stable LC across different sub-divisions of the USA-19 Corpus.

Table 5.4 Lexical coverage standard deviation across four sub-divisions of USA-19 Corpus

Test Corpus	LC
USA-19-News	94.83
USA-19-SMedia	92.17
USA-19-Wiki	87.04
USA-19-Historic	79.97
σ	0.06

LC: Lexical Coverage
σ: Standard Deviation

5.8 Chapter Summary

This chapter investigated the feasibility of rapidly bootstrapping a semantic tagging tool by automatically generating semantic lexicons and creating a software architecture for the Urdu language. Six different automatic or semi-automatic approaches are used to construct Urdu semantic lexicons, these are mapping, crowdsourcing, machine translation, GIZA++, word embedding, and named entity approaches. The semantic lexicons which have been developed in this thesis provide the knowledge base for the US Tagger. Furthermore, a software framework for the Urdu semantic tagging task (US Tagger) has also been developed. The US Tagger annotates text at word level with the following information: POS tag, Lemma, multi-words and semantic tag(s). This chapter concluded by presenting evaluation results, it shows that it is feasible to rapidly generate a prototype tool and semantic lexicons using automatic and semi-automatic approaches.

The results demonstrate that the best results for the US Tagger are achieved using a mapping approach based semantic lexicon (Lexical Coverage = 88.59 (UMC-500K), 99.63 (UMC-1K), 96.71 (UMC-2K), 89.63 (USA-19 Corpus), First Correct = 79.47, Partially Correct = 26.96). However, for better precision, a certain amount of manual improvement and cleaning of Urdu semantic lexicons is indispensable.

The performance of the Urdu semantic tagger mainly depends on the richness of the developed knowledge bases i.e. the Urdu semantic lexicon. Without such types of comprehensive resources that encodes human knowledge, in fact, it is really difficult for semantic tagging tools to effectively understand the meaning associated with natural language text. However, to create such resources manually is an expensive, laborious and time consuming task. Therefore, several ways to automatically-build such knowledge bases have been presented to speed up the creation of taggers particularly for resource-poor languages, since this can help to reduce effort as well as expense of creating large-scale and high-quality resources and tools.

Chapter 6

Conclusions and Future Directions

Semantic tagging can be defined as a dictionary-based process of identifying and labelling the meaning of words in a given text. Over the past two decades, various applications of semantic tagging tools, annotated corpora and resources have been on the increase, including empirical language studies at the semantic level ([180, 158, 106, 171, 166, 213, 165]) and studies in information technology ([227, 71]) amongst others. Consequently, the research community has explored the development of semantic tagging tools, corpora and lexical resources that can carry out semantic analysis of natural languages. However, much of the existing work is for English and major European languages.

In this thesis, algorithm, techniques, corpora, lexicons, supporting resources, and tools have been developed that can be used to carry out semantic analysis of Urdu language text with a unified semantic annotation scheme. Therefore, this thesis aims to address this issue by extending an existing English semantic tagger [180] to cover the Urdu language. All resources of this Ph.D. thesis have been made freely available for the research community at: http://passdropit.com/8SNGiT8L[1] under the terms of the Creative Commons Attribution 4.0 International License[2].

6.1 Summary of the Work

This section presents chapter wise summary of the thesis along with contributions. However, the overall objective contributes the development of an Urdu semantic tagger and supporting resources which are required to perform semantic analysis of Urdu language text.

Chapter 1 of this thesis provided an introduction, by describing the context, problem, objectives, organization, and significance of this research. Furthermore, we detailed the importance and characteristics of the Urdu language. Finally, this chapter ends with several research questions as well as major contributions which have been undertaken in this research work.

In Chapter 2, the background has been established for this thesis by providing definitions of the fundamental and related concepts. Thereafter, the related work of WSD and semantic tagging of corpora as well as techniques, lexical resources, and NLP tools are given. This chapter also provides a survey of word and sentence tokenization, POS tagging methods and corpora which have been developed for the Urdu NLP task. Subsequently, the UCREL Semantic Analysis System has been presented (as a model for the development of the Urdu counterpart), along with its core components (word and sentence tokenizer, POS tagger, lemmatizer, semantic lexicons, and semantic tag disambiguation methods) and its multilingual extension. The chapter concluded with a brief account of evaluation measures used in this research work.

[1] The password can be obtained through following email: jawadshafi@cuilahore.edu.pk

[2] https://creativecommons.org/licenses/by/4.0/ - Last visited: 21-January-2020

In Chapter 3, a detailed development of the Urdu natural language tools (word, sentence tokenizers and POS taggers) has been presented, these tools are core components of the US Tagger. The Urdu word tokenization algorithm is a rule-based morpheme matching approach to solve the space omission which is coupled with a *tri*-gram stochastic language model that backed-off to *bi*-gram maximum likelihood estimation, supplemented by smoothing technique for unknown words. To solve the space insertion problem a dictionary look-up approach is used. For the word tokenization algorithm, a large compound word and morphemes dictionary has also been generated automatically. Apart from algorithms and dictionaries, large benchmark training and testing datasets are also developed. The training dataset consists of 1,361K N-grams whereas, the test dataset contains 59K manually tokenized words. The results of the proposed word tokenizer shows a precision of 0.96, recall of 0.92, F_1 of 0.94, and accuracy of 0.97. The Urdu sentence tokenizer composed of a rule base, regular expressions, and a dictionary look-up approach. To test the Urdu sentence tokenizer, a large dataset is also developed composed of 8K manually annotated sentences. The proposed sentence tokenizer obtained promising results on test dataset, precision = 91.08%, recall = 94.14%, F_1 = 92.59%, and error rate = 6.85%. For the Urdu POS tagging task sixteen different stochastic and two baseline models have been developed. These proposed Urdu POS taggers are based on two stochastic machine learning models that are further supplemented with various language features as well as smoothing estimations. In addition, a large gold-standard training/testing dataset has been formed. The best accuracy of the Urdu POS Tagger is 95.14%, which is based on *tri*-gram Hidden Markov Model, linear interpolation, suffix, and morphological information.

Chapter 4 outlined the development of a benchmark semantically annotated corpus for the Urdu language, USA-19. The USA-19 Corpus follows standard practice for the corpus creation process i.e. data collection, data preprocessing, corpus annotation and inter-annotator agreement, corpus design and standardization. The proposed corpus contains 8K tokens in the following domains: news, social media, Wikipedia, and historical text (each domain having 2K tokens). Furthermore, the USA-19 Corpus is annotated semi-automatically at word level and with 21 major semantic fields and 232 sub-fields with the USAS (UCREL Semantic Analysis System) semantic taxonomy which provides a comprehensive set of semantic fields for coarse-grained annotation. Each word of the proposed corpus is annotated with at least one and up to nine semantic field tags to provide a detailed semantic analysis of the language data, which allowed us to treat the problem of semantic tagging as a supervised multi-target classification task. To demonstrate how a proposed corpus can be used for the development and evaluation of Urdu semantic tagging methods, another contribution of this chapter is to extract local, topical and semantic from the proposed corpus and applied seven different supervised multi-target classifiers on them and compared results. The evaluation showed that best results are obtained using Classifier Chain and Random k-labEL Disjoint Pruned Sets classifiers (Exact Match = 0.76, Hamming Loss = 0.06 and Accuracy = 0.94). It is further observed that regarding single-label ML methods the best results are obtained using Random Forest algorithm.

Chapter 5 described the detailed creation process of the Urdu semantic tagset, lexicons (both single and multi-words) and the US Tagger. However, the main focus of this chapter is to investigate the feasibility of rapidly constructing Urdu semantic lexicons by using automatic and semi-automatic approaches, which act as a knowledge source for the US Tagger. The lexicons are developed using mapping, crowdsourcing, machine translation, GIZA++, word embedding, and named entity approaches. These lexicons have the following statistics: 54,601 (mapping), 2,000 (crowdsourcing), 41,971 (machine translation), 1,285 (GIZA++), 760 (word embedding), and 44,480 (named entities). Entries of the most frequent words in these lexicons are also manually edited. Further to this, a large English-Urdu sentence aligned parallel corpus (7,218 sentences) and the Urdu monolingual corpus (88.33 million tokens) have also been developed as a supporting resources for GIZA++ and word embedding approaches. Aside from these resources, the US Tagger has also been developed which integrated NLP tools, Urdu semantic lexicons, and semantic tag disambiguation methods to annotate at word level. The US Tagger annotates Urdu text with the following four annotations, POS tag, lemma, single or multi-words and semantic tag(s). The US Tagger and Urdu semantic lexicons are evaluated using two corpora, (i) Urdu monolingual , and (ii) USA-19 Corpora and with several evaluation measures. Best average results for the US Tagger (Lexical Coverage = 89.63%, and Precision = 79.47%) are obtained using Urdu mapping based semantic lexicons. Thus, this chapter shows that it is feasible to rapidly generate Urdu semantic lexicons with good lexical coverage. It has also been observed that to achieve a high precision a certain amount of manual improvement and cleaning of Urdu semantic lexicons is also required.

6.2 Thesis Contributions

The main contributions of this thesis are:

1. Development of various Urdu natural language tools along with supporting resources. A state-of-the-art algorithm for word tokenizer has been proposed along with automatically created lexicons. The algorithm is composed of bi-gram morpheme match, tri-gram MLE, which back-off to *bi*-gram MLE as well and Laplace smoothing estimation, and dictionary look-up techniques. The

sentence tokenizer is a rule based, whereas, a tri-gram HMM based POS tagger is proposed along with several smoothing estimations. All tools are tested on newly developed corpora.

2. Creation of the Urdu semantic tagset by automatically translating an existing English semantic tagset into the Urdu language by using machine translation and bilingual dictionaries. Automatically translated tagset is manually verified by two annotators.

3. Development of the Urdu semantic lexicons using automatic or semi-automatic approaches. These approaches are, mapping, crowdsourcing, machine translation, GIZA++, word embedding, and named entities. A large English-Urdu sentence aligned parallel and an Urdu monolingual corpora has been proposed for GIZA++ and word embedding approaches.

4. Development of a multi-target semantically annotated corpus annotated at word level with the USAS semantic tags. A tagged word can have one to nine Urdu semantic field tags to indicate different components of one sense. To demonstrate the development and evaluation of Urdu semantic tagging task topical, semantic and local features are extracted from the proposed corpus and applied seven different multi-target classifiers on them.

5. Development of the Urdu semantic tagger by integrating Urdu semantic lexicons, NLP tools, POS and general-likelihood semantic tag disambiguation methods. The newly created tagger is evaluated on multi-target semantically annotated and other corpora.

6.3 Research Goals Revisited

The main objective of this thesis is the development of the Urdu semantic lexicons, semantic tagger, supporting tools and resources; they function as the core components and knowledge base on which the US Tagger relies. Furthermore, the semantic tagger employs baseline semantic tag disambiguation methods. In order to meet the overall objective, this research has undertaken an investigation as to whether and how it is possible to create resources for the Urdu language which are compatible with the existing English semantic tagger. Therefore, related to meeting the main objective, this thesis has defined the following eight research goals (see Chapter 1).

- **Research goal 1: To explore the in-depth problem of the automatic semantic tagging task for Urdu text to see what new methods and frameworks are required.**

  This research goal has been defined in Chapter 2. Where the fundamental concepts related to the semantic tagging task are presented and we provided a literature review of semantic annotations. The research field which is closely related to semantic tagging is WSD. Therefore, in this chapter corpora and methods for WSD and semantic tagging tasks are presented. Semantic tagger annotate text using semantic lexical resources and NLP tools. Therefore, existing pre-processing tools (sentence/word and POS tagger), semantic lexical resources and datasets are reviewed. In addition to this, the USAS and English semantic tagger are presented along with its key components, application and its multilingual extension, which have functioned as a model for the development of the Urdu counterparts. To evaluate these resources, various evaluation measures are explored.

- **Research goal 2: To develop efficient algorithms and methods as well as extract rules for automatically detecting word and sentence boundaries as well as to assign POS tags to Urdu language text.**

  This research goal has been addressed in Chapter 3. As the primary units for semantic tagger are words and sentences. To the best of our knowledge, there are no word and sentence tokenization tools available which can be embedded in the US Tagger. Therefore, word and sentence tokenizers are developed. To disambiguate semantic tags, a POS tagger is required. To fulfil this need, several Urdu POS taggers are produced based on various state-of-the-art techniques.

- **Research goal 3: To develop large-scale supporting resources (e.g. lexicons, word lists, and annotated corpora) for Urdu word, sentence segmentation and POS tagging.**

  This research goal is also addressed in Chapter 3. As the proposed word, sentence tokenizers and POS taggers are based on statistical, dictionary look-up, rules and machine learning based techniques. These techniques required lexicons, word-lists and annotated corpora to perform annotations. Therefore, to achieve this, several supporting resources for word, sentence tokenizers and POS taggers are developed.

- **Research goal 4: To develop annotated training and testing corpora for multi-target classifiers and to evaluate the US Tagger.**

  This research goal is addressed in Chapter 4. A multi-target semantically annotated corpus is presented to test the performance of multi-target classifiers, the US Tagger, and lexical coverage of several proposed Urdu semantic lexicons. The corpus is annotated at word level and with one to nine semantic tag(s) to show multiple membership categories (different components of one sense) of the USAS semantic taxonomy. Furthermore, the newly developed corpus is used to train and test baseline and feature extraction approaches on seven supervised multi-target classifiers.

- **Research goal 5: To create an Urdu semantic tagset for Urdu semantic tagging task.**

  This research goal is described in Chapter 5. An English USAS semantic tagset is carefully ported semi-automatically for the Urdu language. In order to make sure that Urdu semantic tagset is of good quality a two step approach is used. In the first step, automatic translation of English tags into Urdu is performed. Furthermore, these translation are verified by two human experts.

- **Research goal 6: To develop Urdu semantic lexicons (single and multi-word) using automatic or semi-automatic approaches as well as supporting resources and to determine how extensive are these lexicons in terms of lexical coverage.**

  This research goal is described in Chapter 5, where six different automatic or semi-automatic approaches are used for the creation of Urdu single and multi-word semantic lexicons. The Urdu semantic lexicons are also manually edited by human annotator(s). An English-Urdu sentence aligned parallel corpus and an Urdu monolingual corpus are also developed as a supporting resources for the two approaches. The developed lexicons show encouraging lexical coverage on two test corpora.

- **Research goal 7: To evaluate methods for the semantic tag disambiguation task for Urdu text.**

  This research goal is defined in Chapter 5. The task of the semantic tagger is broadly subdivided into two steps: (i) tag assignment and (ii) tag disambiguation. In tag assignment, a tagger attaches a set of potential semantic tags to each word whereas, in tag disambiguation the contextually appropriate tag is selected. For the second step, various baseline statistical and knowledge based approaches have been applied to improve semantic tag disambiguation, i.e. POS and general-likelihood. The error rate of semantic tag disambiguation methods are also calculated.

- **Research goal 8: To develop a new software framework for the US Tagger and its evaluation.**

  The above research goal is answered in Chapter 5. The US Tagger is developed by integrating Urdu semantic lexicons, NLP tools, and context rules. Furthermore, lexical coverage and precision of newly created semantic lexicons and the US Tagger is also calculated on several benchmark corpora.

6.4 Limitations and Future Directions

Despite favourable results of the proposed tools, methods, lexicons and corpora, however, the following limitations have been observed. A word tokenization method did not handle out-of-vocabulary words in morpheme matching process of space omission problem. Sentence tokenizations are rule based which are not able to dealt with non-sentence boundary markers and period markers used between different abbreviations. Whereas, the POS tagger did not completely handle unknown words. Multi-target classifiers did not explore feature extraction approaches and has only been tested on a small dataset. In addition, state-of-the-art deep learning methods have not been explored for the multi-target task. Furthermore, future research will need to focus on the creation of a semantic multi-word lexicon and the manual cleaning of the single word Urdu semantic lexicon.

This thesis focused on the development process of the US Tagger, Urdu semantic lexical resources, corpora for evaluation, and supporting resources as well as NLP tools to meet the need of semantic analysis of Urdu text. However, semantic tagging is a wide area and there are a number of interesting possibilities for future work and research as follows.

The English semantic tagger (EST) has been used successfully in many corpus and computational linguistics applications, for instance, for the analysis of interview transcripts in market research [247], in the stylistic analysis of written and spoken English [246] in Automatic Content Analysis of Spoken Discourse (ACASD) and Automatic Content Analysis of Market Research Interview Transcripts (ACAMRIT) projects, ussed in a pilot study of a large corpus of doctor patient interactions [230], also EST is utilized in the Requirements Reverse Engineering to Support Business Process Change (REVERE) project [181] in research area of software engineering, in Benedict project[3], where an EST and Finnish semantic tagger have been used together to built a context-sensitive search tool for a new type of intelligent electronic dictionary, used to create historical thesaurus-based semantic tagger for deep semantic annotation [166], to create a historical semantic tagger for English [12], analysis of personal weblogs in Singapore English [158], analysis and standardisation of SMS spelling variation [226], analysis of the semantic content and persuasive composition of extremist media [172], detecting gender and spelling differences in Twitter and SMS [26], for discourse analysis [159, 11], for finding contextual translation equivalents for words in the Russian and English languages [217], in key domain analysis [183], in Metaphors in political discourse [121], for ontology learning [71]; phraseology [82], in Political science research [106], for the protection of children from paedophiles in on-line social networks [176], psychological profiling [127],

[3] The project reference is IST-2001-34237. For more information, see ftp://ftp.cordis.europa.eu/pub/ist/docs/ic/benedict-ist-results_en.pdf. - Last visited: 28-December-2019

for sentiment analysis task [219], to train chatbots and comparing human-human and human-machine dialogues [218], and in deception detection research [127]. It would now be possible to apply the US Tagger for similar purposes. Among these applications, the research community is mainly focusing towards sentiment analysis and cyber security. Therefore, a possible future research venture can be the development of a social media based content monitoring application such as hate speech detection, studying and analysing the speech of the selected targeted group to control terrorism activities, etc. Furthermore, another research interest for research community can be to investigate the financial text mining[4] using the US Tagger. There is a dire social need for such applications.

In terms of supporting tools which have been proposed in this thesis, possible future work extensions for word tokenization can be the use of some other machine learning approaches (conditional random field, maximum entropy, neural networks etc.) to learn the morphological pattern of the valid morphemes (instead of morpheme look-up) and extend experiments to larger datasets as well as handle out-of-vocabulary words in the morpheme matching process of the space omission problem. For the Urdu sentence tokenization task, a possible extension is to develop a hybrid Urdu sentence tokenizer i.e. using the rule-based algorithm along with machine learning-based classifiers (such as conditional random field, sequential minimal optimization). In terms of Urdu POS tagging, to handle unknown words is a challenging task that needs to be addressed in the future. Another possible extension can be the development of a hybrid POS tagger, in which various ML statistical methods (CRF, SVM, etc.) along with heuristic rules can be adopted to improve POS tagging.

[4] As in the Corporate Financial Information Environment (CFIE) project, where the English semantic tagger has been used to perform analysis of UK corporate news stories: http://ucrel.lancs.ac.uk/cfie/ - Last visited: 13-April-2019

In the case of the semantically annotated corpus, other feature extraction approach(es) and multi-label classifiers can also be explored. Increasing the size of the corpus is another avenue for future work. Considering the Urdu semantic lexicons, a possible extension can be to generate large-scale multi-word semantic lexicons. Furthermore, the development of a hybrid Urdu semantic tagger is a area which needs further research. Moreover, further collocations feature should need to be explore.

The EST has been extended for Czech, Chinese, Dutch, French, Italian, Malay, Portuguese, Russian, Spanish, Finnish, Welsh, Urdu and Arabic. However, there is a further plan to extend the EST framework to cover Swedish, Norwegian, and Turkish languages. As a consequence, now there are equivalent semantic taggers based on equivalent semantic lexicons which are capable of processing several languages. These semantic taggers available for multiple languages enable the development of multi-lingual and cross-lingual applications, as the semantic tagset acts as a kind of a "meta-dictionary" or "lingua-franca" between the languages. This would make it possible to use these semantic taggers for cross-lingual applications, for instance, machine translation, plagiarism detection, and information extraction as well as retrieval tasks.

References

[1] Abbas, Q. (2016a). Morphologically rich Urdu grammar parsing using Earley algorithm. *Natural Language Engineering*, 22(5):775–810.

[2] Abbas, Q. (2016b). Semi-semantic annotation: A guideline for the Urdu. KON-TB treebank POS annotation. *Acta Linguistica Asiatica*, 6(2):97–134.

[3] Abdelhamid, A. A., Abdulla, W. H., and MacDonald, B. (2012). WFST-based large vocabulary continuous speech decoder for service robots. In *Proceedings of the International Conference on Imaging and Signal Processing for Healthcare and Technology (ISPHT'12), Baltimore, USA*, pages 150–154. ACTA Press.

[4] Abid, M., Habib, A., Ashraf, J., and Shahid, A. (2018). Urdu word sense disambiguation using machine learning approach. *Cluster Computing*, 21(1):515–522.

[5] Agirre, E. and Martinez, D. (2000). Exploring automatic word sense disambiguation with decision lists and the web. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content, Luxembourg*, pages 11–19. Association for Computational Linguistics.

[6] Agirre, E., Martínez, D., de Lacalle, O. L., and Soroa, A. (2006). Two graph-based algorithms for state-of-the-art WSD. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'04), Sydney, Australia*, pages 585–593. Association for Computational Linguistics.

[7] Agirre, E. and Stevenson, M. (2007). Knowledge sources for WSD. *Word Sense Disambiguation*, 33:217–251.

[8] Ahmed, T. (2009). Roman to Urdu transliteration using wordlist. In *Proceedings of the Conference on Language and Technology (CLT'09), Lahore, Pakistan.*, volume 305, page 309.

[9] Aker, A., Feng, Y., and Gaizauskas, R. (2012). Automatic bilingual phrase extraction from comparable corpora. *Proceedings of the 24th International Conference on Computational Linguistics of Posters Demonstration (COLING'12), Mumbai, India*, pages 23–32.

[10] Aker, A., Paramita, M. L., Pinnis, M., and Gaizauskas, R. (2014). Bilingual dictionaries for all EU languages. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland*, pages 2839–2845. ELRA.

[11] Al-Hejin, B. (2015). Covering Muslim women: Semantic macrostructures in BBC news. *Discourse & Communication*, 9(1):19–46.

[12] Alexander, M., Dallachy, F., Piao, S., Baron, A., and Rayson, P. (2015). Metaphor, popular science, and semantic tagging: Distant reading with the Historical Thesaurus of English. *Digital Scholarship in the Humanities (DSH)*, 30(suppl_1):i16–i27.

[13] Ali, A. R. and Ijaz, M. (2009). Urdu text classification. In *Proceedings of the 7th international conference on frontiers of information technology, (FIT'09), Abbottabad, Pakistan*, page 21. ACM.

[14] Alias-I (2008). LingPipe 4.1.0. *http://alias-i.com/lingpipe (Last visited: 23-December-2017)*.

[15] Allan, J. (2012). *Topic detection and tracking: Event-based information organization*, volume 12. Springer Science & Business Media.

[16] Anwar, W., Wang, X., Li, L., and Wand, X. (2007b). Hidden Markov model based part of speech tagger for Urdu. *Information Technology Journal*, 6(8):1190–1198.

[17] Anwar, W., Wang, X., Li, L., and Wang, X.-L. (2007a). A statistical based part of speech tagger for Urdu language. In *International Conference on Machine Learning and Cybernetics (ICMLC'07), Hong Kong, China*, volume 6, pages 3418–3424. IEEE.

[18] Archer, D., Wilson, A., and Rayson, P. (2002). Introduction to the USAS category system. *Benedict project report, October 2002*.

[19] Artzi, Y., Lee, K., and Zettlemoyer, L. (2015). Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, (EMNLP'15), Lisbon, Portugal*, pages 1699–1710. Association for Computational Linguistics (ACL).

[20] Azimizadeh, A., Arab, M. M., and Quchani, S. R. (2008). Persian part of speech tagger based on Hidden Markov Model. In *Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data (JADT'08), Lyon, France*, pages 121–128.

[21] Baker, P., Hardie, A., McEnery, T., Cunningham, H., and Gaizauskas, R. J. (2002). EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Markup and Harmonisation. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02), Canary Islands - Spain*, pages 819–825.

[22] Baker, P., Hardie, A., McEnery, T., and Jayaram, B. (2003). Corpus Data for South Asian Language Processing. In *Proceedings of the 10th Annual Workshop for South Asian Language Processing (EACL'03), Budapest, Hungary*, pages 1–8. European Chapter of the ACL.

[23] Balossi, G. (2014). *A corpus linguistic approach to literary language and characterization: Virginia Woolf's The Waves*, volume 18. John Benjamins Publishing Company.

[24] Barberá, P. (2015). Birds of the same feather tweet together: Bayesian ideal point estimation using Twitter data. *Political Analysis*, 23(1):76–91.

[25] Barbu, E. (2007). Automatic building of WordNets EduArd BarbU* &: Verginica BarbU MiTiTElU*** Graphitech Italy" Romanian Academy, Research Institute for Artificial Intelligence. *Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005*, 292:217–226.

[26] Baron, A., Tagg, C., Rayson, P., Greenwood, P., Walkerdine, J., and Rashid, A. (2011). Using verifiable author data: Gender and spelling differences in Twitter and SMS. In *International Computer Archive of Modern and Medieval English (ICAME 31), Oslo, Norway*, pages 61–73.

[27] Basili, R., Della Rocca, M., and Pazienza, M. T. (1997). Towards a bootstrapping framework for corpus semantic tagging. In *Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics: What, why and how?", Washington, D.C. USA*, pages 66–73. The Association for Neuro Linguistic Programming (ANLP).

[28] Baudiš, P. (2015). YodaQA: A modular question answering system pipeline. In *POSTER 2015-19th International Student Conference on Electrical Engineering, Prague, Czech Republic*, pages 1156–1165. Faculty of Electrical Engineering, CTU Prague.

[29] Becker, D. and Riaz, K. (2002). A study in Urdu corpus construction. In *Proceedings of the 3rd workshop on Asian language resources and international standardization, COLING 2002 post conference workshop, Taipei, Taiwan*, volume 12, pages 46–50. Association for Computational Linguistics (ACL).

[30] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

[31] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. " O'Reilly Media, Inc.".

[32] Board, U. D. (2008). Urdu Lughat. *Urdu Lughat Board, Karachi, Pakistan*.

[33] Bögel, T., Butt, M., Hautli, A., and Sulger, S. (2007). Developing a finite-state morphological analyzer for Urdu and Hindi. In *Proceedings of the Sixth International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP'07), Potsdam, Germany*, pages 86–96. The Linguistics Department, Potsdam University.

[34] Bond, F. and Ogura, K. (2008). Combining linguistic resources to create a machine-tractable Japanese-Malay dictionary. *Language Resources and Evaluation*, 42(2):127–136.

[35] Bontcheva, K., Tablan, V., Maynard, D., and Cunningham, H. (2004). Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering*, 10(3-4):349–373.

[36] Bordag, S. (2006). Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06), Trento, Italy*, pages 137–144. Association for Computational Linguistics.

[37] Boutell, M., Shen, X., Luo, J., and Brown, C. (2003). Multi-label semantic scene classification. Technical report, technical report, department of computer sciences. u. Rochester.

[38] Brants, T. (2000). TnT: A statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, Seattle, Washington, USA*, pages 224–231. Association for Computational Linguistics (ACL).

[39] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

[40] Bruce, R. F. and Wiebe, J. M. (1999). Decomposable modeling in natural language processing. *Computational Linguistics*, 25(2):195–207.

[41] Butt, M. (2014). The structure of Urdu–case. Technical report, Universität Konstanz Germany.

[42] Cai, J. F., Lee, W. S., and Teh, Y. W. (2007). NUS-ML: Improving word sense disambiguation using topic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, Czech Republic*, pages 249–252. Association for Computational Linguistics.

[43] Cambria, E., Grassi, M., Hussain, A., and Havasi, C. (2012). Sentic computing for social media marketing. *Multimedia tools and applications*, 59(2):557–577.

[44] Cambria, E. and Hussain, A. (2015). Sentic computing. *Cognitive Computation*, 7(2):183–185.

[45] Carpenter, B. and Baldwin, B. (2011). Text analysis with LingPipe 4. *New York: Ling Pipe Publishing*.

[46] Charte, F., Rivera, A., del Jesus, M., and Herrera, F. (2018). *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer.

[47] Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL'17), Vancouver, Canada*, pages 1870–1879. ACL.

[48] Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

[49] Chiticariu, L., Li, Y., and Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13), Seattle, Washington, USA*, pages 827–832. Association for Computational Linguistics (ACL).

[50] Christensen, H. (2014). HC Corpora. *http://www.corpora.heliohost.org/ (Last visited: 05-March-2017)*.

[51] Clare, A. and King, R. (2001). Knowledge discovery in multi-label phenotype data. *Principles of data mining and knowledge discovery*, pages 42–53.

[52] Clough, P., Gaizauskas, R. J., and Piao, S. S. (2002). Building and annotating a corpus for the study of journalistic text reuse. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02), Las Palmas, Canary Islands, Spain*, pages 1678–1685. European Language Resources Association (ELRA).

[53] Cohen, J. (1968). Weighted Kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213–220.

[54] Cunningham, H., Maynard, D., and Bontcheva, K. (2011). *Text processing with GATE*. Gateway Press CA.

[55] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, USA*, pages 168–175. Association for Computational Linguistics (ACL).

[56] Cunningham, H., Tablan, V., Roberts, A., and Bontcheva, K. (2013). Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. *PLoS Computational Biology*, 9(2):1–16.

[57] Curran, J. R. and Clark, S. (2003). Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics (EACL'03), Budapest, Hungary*, volume 1, pages 91–98. Association for Computational Linguistics (ACL).

[58] Dandapat, S. (2007). Part of specch tagging and chunking with Maximum Entropy model. In *Proceedings of the IJCAI Workshop on Shallow Parsing for South Asian Languages (IJCAI'08), Hyderabad, India.*, pages 29–32.

[59] Daud, A., Khan, W., and Che, D. (2016). Urdu language processing: A survey. *Artificial Intelligence Review*, 47(3):279–311.

[60] de Carvalho, A. and Freitas, A. (2009). A tutorial on multi-label classification techniques. *Foundations of Computational Intelligence*, 5:177–195.

[61] Decadt, B., Hoste, V., Daelemans, W., and Van den Bosch, A. (2004). GAMBL, genetic algorithm optimization of memory-based WSD. In *3rd International workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3); held in conjunction with the 42nd Annual meeting of the Association for Computational Linguistics (ACL'04), Barcelona, Spain*, pages 108–112. Association for Computational Linguistics.

[62] Demetriou, G. and Atwell, E. S. (2001). A domain-independent semantic tagger for the study of meaning associations in English text. In *Proceedings of the Fourth International Workshop on Computational Semantics (IWCS'4), Prague, Czech Republic*, pages 67–80. Association for Computational Linguistics (ACL) Special Interest Group in Computational Semantics (SIGSEM).

[63] Derczynski, L., Maynard, D., Aswani, N., and Bontcheva, K. (2013). Microblog-genre noise and impact on semantic annotation accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media, (HT '13), Paris, France*, pages 21–30. ACM.

[64] Dermatas, E. and Kokkinakis, G. (1995). Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–163.

[65] Doherty, N., Lockett, N., Rayson, P., and Riley, S. (2006). Electronic-CRM: A simple sales tool or facilitator of relationship marketing? In *29th Institute for Small Business & Entrepreneurship Conference. International Entrepreneurship-from local to global enterprise creation and development (ISBE'06), Cardiff, Wales, United Kingdom*.

[66] Durrani, N. and Hussain, S. (2010). Urdu Word Segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, California, USA*, pages 528–536. Association for Computational Linguistics (ACL).

[67] Ekbal, A., Haque, R., and Bandyopadhyay, S. (2008). Maximum Entropy based Bengali part of speech tagging. *A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal*, 33:67–78.

[68] El-Haj, M., Rayson, P., Piao, S., and Wattam, S. (2017). Creating and validating multilingual semantic representations for six languages: Expert versus non-expert crowds. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications (SENSE'17), Valencia, Spain*, pages 61–71. Association for Computational Linguistics.

[69] Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10), Los Angeles, California, USA*, pages 80–88. Association for Computational Linguistics (ACL).

[70] Fu, G., Kit, C., and Webster, J. J. (2008). Chinese word segmentation as morpheme-based lexical chunking. *Information Sciences*, 178(9):2282–2296.

[71] Gacitua, R., Sawyer, P., and Rayson, P. (2008). A flexible framework to experiment with ontology learning techniques. In *Research and Development in Intelligent Systems XXIV*, pages 153–166. Springer.

[72] Gale, W. A., Church, K. W., and Yarowsky, D. (1992a). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439.

[73] Gale, W. A., Church, K. W., and Yarowsky, D. (1992b). One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics.

[74] Garside, R., Leech, G. N., and McEnery, T. (1997). *Corpus annotation: Linguistic information from computer text corpora*. Taylor & Francis.

[75] Garside, R. and Rayson, P. (1997). Higher-level annotation tools. *Corpus Annotation: Linguistic Information from Computer Text Corpora. Longman, London*, pages 179–193.

[76] Garside, R. and Smith, N. (1997). A hybrid grammatical tagger: CLAWS4. *Corpus annotation: Linguistic information from computer text corpora, Longman, London*, pages 102–121.

[77] Gentile, A. L., Basile, P., Iaquinta, L., and Semeraro, G. (2008). Lexical and semantic resources for NLP: From words to meanings. In *Proceedings of the 6th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES'08), Zagreb, Croatia*, pages 277–284. Springer.

[78] Giménez, J. and Marquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal*, pages 43–46. European Language Resources Association (ELRA).

[79] Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *The Journal of Artificial Intelligence Research (JAIR)*, 57:345–420.

[80] Gouws, S., Bengio, Y., and Corrado, G. (2015). BilBOWA: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15), Lille, France*, pages 748–756.

[81] Graff, D. and Cieri, C. (2003). English gigaword corpus. *Corpus number LDC2003T05, Linguistic Data Consortium, Philadelphia.*

[82] Granger, S., Paquot, M., and Rayson, P. (2006). Extraction of multi-word units from EFL and native English corpora: The phraseology of the verb 'make'. *Phraseology in motion I: Methoden und Kritik*, pages 57–68.

[83] Hardie, A. (2003). Developing a tagset for automated part-of-speech tagging in Urdu. In *In Archer, D, Rayson, P, Wilson, A, and McEnery, T (eds.) Proceedings of the Corpus Linguistics 2003 conference. UCREL Technical Papers, Lancaster, UK*, volume 16, pages 298–307. Department of Linguistics, Lancaster University, UK.

[84] Hardie, A. (2004). *The computational analysis of morphosyntactic categories in Urdu.* PhD thesis, Lancaster University, UK.

[85] Hautli, A. and Sulger, S. (2011). Extracting and classifying Urdu multiword expressions. In *Proceedings of the the ACL-HLT Student Session (ACL-HLT'11), Portland, OR, USA.*, pages 24–29. Association for Computational Linguistics.

[86] Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. (1998). University of Sheffield: Description of the LaSIE-ii system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998.*

[87] Hussain, S. (2008). Resources for Urdu Language Processing. In *Proceedings of the 6th Workshop on Asian Language Resources, International Joint Conference on Natural Langauge Processing IJCNLP'08, Hyderabad, India*, pages 99–100. Asian Federation of Natural Language Processing (AFNLP).

[88] Ide, N. and Véronis, J. (1998). Word sense disambiguation: The state of the art. *Computational linguistics*, 24(1):1–41.

[89] Jarmasz, M. and Szpakowicz, S. (2004). Roget's thesaurus and semantic similarity. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 3(1):111–120.

[90] Javed, I. (1985). Nai Urdu Qawaid. *Urdu Development Board, New Delhi.*

[91] Jawaid, B., Kamran, A., and Bojar, O. (2014a). A Tagged Corpus and a Tagger for Urdu. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'09), Reykjavík, Iceland.*, pages 2938–2943. European Language Resources Association (ELRA).

[92] Jawaid, B., Kamran, A., and Bojar, O. (2014b). Urdu Monolingual Corpus. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[93] Jawaid, B. and Zeman, D. (2011). Word-order issues in English-to-Urdu statistical machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:87–106.

[94] Jeffreys, H. (1998). *The theory of probability*, volume 3rd. Oxford University Press.

[95] Jessica Butt, M. (1995). *The structure of complex predicates in Urdu*. PhD thesis, Centre for the Study of Language (CSLI), department of linguistics, Stanford University.

[96] Joshi, N., Darbari, H., and Mathur, I. (2013). HMM based POS tagger for Hindi. In *Proceedings of the 2013 International Conference on Artificial Intelligence and Soft Computing (AISC'13), Bangalore, India*, pages 341–349.

[97] Jurafsky, D. and Martin, J. (2014). *Speech & language processing, 2nd edition*, volume 3. Pearson London.

[98] Kamholz, D., Pool, J., and Colowick, S. M. (2014). PanLex: Building a Resource for Panlingual Lexical Translation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland*, pages 3145–3150.

[99] Kazai, G. (2011). In search of quality in crowdsourcing for search engine evaluation. In *Proceedings of the 33rd European Conference on Information Retrieval, Dublin, Ireland*, pages 165–176. Springer.

[100] Kazai, G., Kamps, J., Koolen, M., and Milic-Frayling, N. (2011). Crowdsourcing for book search evaluation: Impact of hit design on comparative system ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR'11), Beijing, China*, pages 205–214. ACM.

[101] Kazai, G., Milic-Frayling, N., and Costello, J. (2009). Towards methods for the collective gathering and quality control of relevance assessments. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGR'09), New York, USA*, pages 452–459. ACM.

[102] Khan, S. A., Anwar, W., Bajwa, U. I., and Wang, X. (2012). A light weight stemmer for Urdu language: A scarce resourced language. In *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), (COLING'12), Mumbai, India*, pages 69–78. Association for Computational Linguistics (ACL).

[103] Khana, W., Daudb, A., Nasira, J. A., and Amjada, T. (2016). Named entity dataset for Urdu named entity recognition task. In *Proceedings of the 6th Conference on Language and Technology (CLT'16), Lahore, Pakistan*, pages 51–56. Society for Natural Language Processing (SNLP).

[104] Kilgarriff, A. (2004). How dominant is the commonest sense of a word? In *International Conference on Text, Speech and Dialogue (TSD'04), Brno, Czech Republic*, pages 103–111. Springer.

[105] Kilgarriff, A. and Yallop, C. (2000). What's in a Thesaurus? In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00), Athens, Greece*, pages 1–8. European Language Resources Association (ELRA).

[106] Klebanov, B. B., Diermeier, D., and Beigman, E. (2008). Automatic annotation of semantic fields for political science research. *Journal of Information Technology & Politics*, 5(1):95–120.

[107] Kupietz, M., Belica, C., Keibel, H., and Witt, A. (2010). The German Reference Corpus DeReKo: A Primordial Sample for Linguistic Research. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta*, pages 1848–1854. European Language Resources Association (ELRA).

[108] Kwartler, T. (2017). *Text Mining in Practice with R (1st ed.)*. John Wiley & Sons.

[109] Lam, K. N. (2014). Automatically creating multilingual lexical resources. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence and the 26th Innovative Applications of Artificial Intelligence Conference (AAAI'14), Québec, Canada*, pages 3077–3078.

[110] Lam, K. N., Al Tarouti, F., and Kalita, J. (2014). Creating lexical resources for endangered languages. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages, Maryland, USA*, pages 54–62.

[111] Leech, G. (2005). *Adding linguistic annotation. In M. Wynne, editor, Developing Linguistic Corpora: A Guide to Good Practice, pages 17–29.* Oxbow Books.

[112] Leech, G., Rayson, P., and Wilson, A. (2001). *Word frequencies in written and spoken English: based on the British National Corpus*. London: Longman.

[113] Lehal, G. S. (2010). A word segmentation system for handling space omission problem in Urdu script. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), the 23rd International Conference on Computational Linguistics, Beijing, China*, pages 43–50. COLING.

[114] Lindemann, D., Manterola, I., Nazar, R., San Vicente, I., and Saralegi, X. (2014). Bilingual Dictionary Drafting. The example of German-Basque, a medium-density language pair. In *Proceedings of the XVI EURALEX International Congress: The User in Focus, (EURALEX'14), Bolzano Bozen, Italy*, pages 563–576. Institute for Specialised Communication and Multilingualism.

[115] Liu, F., Zhang, X., Ye, Y., Zhao, Y., and Li, Y. (2015). Mlrf: Multi-label classification through random forest with label-set partition. In *International Conference on Intelligent Computing (ICIC'15) Fuzhou, China*, pages 407–418. Springer.

[116] Löfberg, L., Archer, D., Piao, S., Rayson, P., McEnery, T., Varantola, K., and Juntunen, J.-P. (2005a). Porting an English semantic tagger to the Finnish language. In *Proceedings of the Corpus Linguistics 2005 conference, Birmingham, UK*, pages 457–464.

[117] Lofberg, L., Juntunen, J.-P., Nykanen, A., Varantola, K., Rayson, P., and Archer, D. (2004). Using a semantic tagger as dictionary search tool. In *11th EURALEX (European Association for Lexicography) International Congress (Euralex 2004)*, pages 127–134.

[118] Löfberg, L., Piao, S., Rayson, P., Juntunen, J.-P., Nykanen, A., and Varantola, K. (2005b). A semantic tagger for the Finnish language. In *Proceedings of Corpus Linguistics, Birmingham, UK*, pages 1–12. Organised jointly by the universities of Birmingham and Lancaster.

[119] Lowe, J. B. (1997). A frame-semantic approach to semantic annotation. In *Proceedings of the SIGLEX workshop "Tagging Text with Lexical Semantics, (SIGLEX'97), Washington D.C., USA*, pages 18–24.

[120] Löfberg, L. (2017). *Creating large semantic lexical resources for the Finnish language*. PhD thesis, Lancaster University.

[121] L'Hôte, E. and Lemmens, M. (2009). Reframing treason: Metaphors of change and progress in new Labour discourse. *CogniTextes*, (3):1–29.

[122] 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):1–13.

[123] Mamakis, G., Malamos, A. G., and Ware, J. A. (2011). An alternative approach for statistical single-label document classification of newspaper articles. *Journal of Information Science*, 37(3):293–303.

[124] Mamakis, G., Malamos, A. G., Ware, J. A., and Karelli, I. (2012). Document classification in summarization. *Journal of Information and Computing Science*, 7(1):25–36.

[125] Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*, volume 999. Cambridge Massachusetts:MIT Press.

[126] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland USA*, pages 55–60. Association for Computational Linguistics (ACL).

[127] Markowitz, D. M. and Hancock, J. T. (2014). Linguistic traces of a scientific fraud: The case of Diederik Stapel. *PloS one*, 9(8):e105937.

[128] Maynard, D., Greenwood, M. A., Roberts, I., Windsor, G., and Bontcheva, K. (2015). Real-time social media analytics through semantic annotation and linked open data. In *Proceedings of the ACM Web Science Conference (WebSci'15), Oxford, UK*, pages 46–47. Association for Computing Machinery (ACM).

[129] McArthur, T. (1981). *Longman lexicon of contemporary English*. Longman London.

[130] McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2007). Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

[131] McHugh, M. L. (2012). Interrater reliability: The Kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282.

[132] Mikolov, T., Le, Q. V., and Sutskever, I. (2013a). Exploiting similarities among languages for machine translation. In *In International Conference on Learning Representations (ICLR'13), Scottsdale, USA*, pages 1–10.

[133] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems (NIPS'13), Stateline, USA*, pages 3111–3119.

[134] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–312.

[135] Miller, G. and Fellbaum, C. (1998). *WordNet: An electronic lexical database*. MIT Press Cambridge.

[136] Mohamed, G., Potts, A., and Hardie, A. (2013). AraSAS: A semantic tagger for Arabic. In *Proceedings of the 2nd Workshop on Arabic Corpus Linguistics, Lancaster, UK*, pages 1–6.

[137] Mooney, R. J. (1996). Comparative experiments on disambiguating word senses: An illustration of the role of Bias in Machine Learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP'96), Philadelphia, USA*, pages 1–10. ACL.

[138] Moro, A., Navigli, R., Tucci, F. M., and Passonneau, R. J. (2014). Annotating the MASC Corpus with BabelNet. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland*, pages 4214–4219. European Language Resources Association (ELRA).

[139] Muaz, A., Ali, A., and Hussain, S. (2009). Analysis and development of Urdu POS tagged corpus. In *Proceedings of the 7th Workshop on Asian Language Resources (ALR'7), Suntec, Singapore*, pages 24–29. Association for Computational Linguistics (ACL).

[140] Mudraya, O., Babych, B., Piao, S., Rayson, P., and Wilson, A. (2006). Developing a Russian semantic tagger for automatic semantic annotation. In *Proceedings of Corpus Linguistics 2006, St. Petersburg, Russian Federation*, pages 290–297.

[141] Muhammad, H., Rao Muhammad, A. N., Muhammad, U., Saba, A., and Omer, F. (2016a). Urdu summary corpus. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia*, pages 796–800. European Language Resources Association (ELRA).

[142] Muhammad, S., Rayson, P. E., and Nawab, R. M. A. (2016b). UPPC–Urdu Paraphrase Plagiarism Corpus. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia*, pages 1832–1836. European Language Resources Association (ELRA).

[143] Mukund, S., Srihari, R., and 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.

[144] Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Sydney, Australia*, pages 81–88. ACL.

[145] Naseer, A. and Hussain, S. (2010). Supervised Word Sense Disambiguation for Urdu Using Bayesian Classification. In *Proceedings of the Conference on Language and Technology, (CLT'10), Lahore, Pakistan*, pages 1–5. Center for Research in the Urdu Language Processing.

[146] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

[147] Navigli, R. (2012). A quick tour of word sense disambiguation, induction and related approaches. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 115–129. Springer.

[148] Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

[149] Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

[150] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, Baltimore, Maryland*, pages 1–14. Association for Computational Linguistics (ACL).

[151] Ng, V. and Cardie, C. (2003). Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'03), Edmonton, Canada*, pages 173–180.

[152] Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages: co-located with the 8th Extended Semantic Web Conference (ESWC'11), Heraklion, Crete, Greece's*, pages 1–6.

[153] Niu, Z.-Y., Ji, D.-H., and Tan, C. L. (2005). Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05), Michigan, USA*, pages 395–402. Association for Computational Linguistics.

[154] Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hongkong*, pages 440–447. ACL.

[155] Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

[156] Ofoghi, B., López-Campos, G., Martín-Sánchez, F. J., and Verspoor, K. (2014). Mapping biomedical vocabularies: A semi-automated term matching approach. In *Proceedings of the International Conference on Informatics, Management, and Technology in Healthcare (ICIMTH'14), Athens, Greece*, pages 16–19. European Federation for Medical Informatics (EFMI).

[157] Oliver, A. and Climent, S. (2012). Parallel corpora for WordNet construction: Machine translation vs. automatic sense tagging. In *International Conference on Intelligent Text Processing and Computational Linguistics, New Delhi, India*, pages 110–121. Springer.

[158] Ooi, B. Y. V., Tan, K. W. P., and Chiang, K. L. A. (2007). Analyzing personal weblogs in Singapore English: The Wmatrix approach.

[159] O'Halloran, K. (2011). Limitations of the logico-rhetorical module: Inconsistency in argument, online discussion forums and electronic deconstruction. *Discourse Studies*, 13(6):797–806.

[160] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics, Pennsylvania, USA*, pages 311–318. ACL.

[161] Pelevina, M., Arefyev, N., Biemann, C., and Panchenko, A. (2017). Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP (RepL4NLP'16), Berlin, Germany*, pages 174–183. Association for Computational Linguistics (ACL).

[162] Pfeifer, C. (2011). Author attribution in US supreme court decisions. *Frontiers in Artificial Intelligence and Applications*, 235(Legal Knowledge and Information Systems):145–149.

[163] Philip, E. and Eneko, A. (2006). *Word sense disambiguation: Algorithms and Applications*, volume 33. E. Agirre and P. Edmonds, Eds. Springer Verlag. Text, Speech and Language Technology Series, New York, NY. USA.

[164] Piao, S., Archer, D., Mudraya, O., Rayson, P., Garside, R., McEnery, T., and Wilson, A. (2005). A large semantic lexicon for corpus annotation. *Corpus Linguistics 2005*.

[165] Piao, S., Bianchi, F., Dayrell, C., D'egidio, A., and Rayson, P. (2015). Development of the multilingual semantic annotation system. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL-HLT'15), Denver, Colorado, USA*, pages 1268–1274. Association for Computational Linguistics (ACL).

[166] Piao, S., Dallachy, F., Baron, A., Demmen, J., Wattam, S., Durkin, P., McCracken, J., Rayson, P., and Alexander, M. (2017). A time-sensitive historical thesaurus-based semantic tagger for deep semantic annotation. *Computer Speech & Language*, 46(2017):113–135.

[167] Piao, S., Rayson, P., Archer, D., and McEnery, A. (2004). Evaluating lexical resources for a semantic tagger. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal*, pages 499–502.

[168] Piao, S. S., Rayson, P., Archer, D., Wilson, A., and McEnery, T. (2003). Extracting multiword expressions with a semantic tagger. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18 (ACL'03), Sapporo, Japan*, pages 49–56. Association for Computational Linguistics.

[169] Platts, J. T. (1909). *A grammar of the Hindustani or Urdu language*. London: Crosby Lockwood and Son, republished in 2002 by Sang-e-Meel Publications, Lahore.

[170] Post, M., Callison-Burch, C., and Osborne, M. (2012). Constructing parallel corpora for six Indian languages via crowdsourcing. In *Proceedings of the 7th Workshop on Statistical Machine Translation, Montreal, Canada*, pages 401–409. ACL.

[171] Potts, A. and Baker, P. (2012). Does semantic tagging identify cultural change in British and American English? *International journal of corpus linguistics*, 17(3):295–324.

[172] Prentice, S., Taylor, P. J., Rayson, P., Hoskins, A., and O'Loughlin, B. (2011). Analyzing the semantic content and persuasive composition of extremist media: A case study of texts produced during the Gaza conflict. *Information Systems Frontiers*, 13(1):61–73.

[173] Procter, P. (1978). *Longman dictionary of contemporary English*. Harlow England: Longman.

[174] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

[175] Raj, S., Rehman, Z., Rauf, S., Siddique, R., and Anwar, W. (2015). An Artificial Neural Network Approach for Sentence Boundary Disambiguation in Urdu. *The International Arab Journal of Information Technology*, 12(4):395–400.

[176] Rashid, A., Greenwood, P., Walkerdine, J., Baron, A., and Rayson, P. (2012). Technological solutions to offending. In *Understanding and preventing online sexual exploitation of children*, pages 244–259. Routledge.

[177] Rashid, R. and Latif, S. (2012). A dictionary based Urdu word segmentation using maximum matching algorithm for space omission problem. In *Proceedings of the International Conference on Asian Language Processing (IALP'17), Hanoi, Vietnam*, pages 101–104. IEEE.

[178] Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'96), New Jersey, USA*, volume 1, pages 133–142. Association for Computational Linguistics (ACL).

[179] Rayson, P. (2003). *Matrix: A statistical method and software tool for linguistic analysis through corpus comparison*. PhD thesis, Lancaster University.

[180] Rayson, P., Archer, D., Piao, S., and McEnery, A. (2004). The UCREL Semantic Analysis System. In *proceedings of the workshop on Beyond Named Entity Recognition Semantic labelling for NLP tasks in association with 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal*, pages 7–12. European Language Resources Association (ELRA).

[181] Rayson, P., Emmet, L., Garside, R., and Sawyer, P. (2000). The REVERE Project: Experiments with the application of probabilistic NLP to Systems Engineering. In *Proceedings of the 5th International Conference on Applications of Natural Language to Information Systems (NLDB'00), Versailles, France*, pages 288–300. Springer.

[182] Rayson, P., Garside, R., and Sawyer, P. (1999). Recovering legacy requirements. In *Proceedings of the of the 5th International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'99), Heidelberg, Germany*, pages 49–54. Foundation for Software Quality.

[183] Rayson, P. and Smith, N. (2006). The key domain method for the study of language varieties. In *The Third Inter-Varietal Applied Corpus Studies (IVACS'06) group International Conference on "Language at the Interface", University of Nottingham, UK*, pages 1–7. University of Nottingham.

[184] Rayson, P. and Stevenson, M. (2008). Sense and semantic tagging. In *A. Lüdeling M. Kytö (eds.) Corpus linguistics. Berlin: De Gruyter.*, pages 564–579. Mouton de Gruyter.

[185] Read, J., Martino, L., and Luengo, D. (2014). Efficient Monte Carlo methods for multi-dimensional learning with classifier chains. *Pattern Recognition*, 47(3):1535–1546.

[186] Read, J., Pfahringer, B., and Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08), Pisa, Italy*, pages 995–1000. IEEE.

[187] Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359.

[188] Read, J., Reutemann, P., Pfahringer, B., and Holmes, G. (2016). MEKA: A multi-label/multi-target extension to WEKA. *The Journal of Machine Learning Research*, 17(1):667–671.

[189] Recasens, M. and Martí, M. A. (2010). AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language Resources and Evaluation (LRE)*, 43(4):315–345.

[190] Rehman, Z. and Anwar, W. (2012). A hybrid approach for Urdu sentence boundary disambiguation. *The International Arab Journal of Information Technology*, 9(3):250–255.

[191] Rehman, Z., Anwar, W., and Bajwa, U. I. (2011). Challenges in Urdu text tokenization and sentence boundary disambiguation. In *Proceedings of the 2nd Workshop on South Southeast Asian Natural Language Processing (WSSANLP'11), Chiang Mai, Thailand*, pages 40–45.

[192] Rehman, Z., Anwar, W., Bajwa, U. I., Xuan, W., and Chaoying, Z. (2013). Morpheme matching based text tokenization for a scarce resourced language. *PLoS One*, 8(8):1–8.

[193] Riaz, K. (2008a). Baseline for Urdu IR evaluation. In *Proceedings of the 2nd ACM workshop on Improving Non-English Web Searching (iNEWS'08), CA, USA*, pages 97–100. Association for Computing Machinery (ACM).

[194] Riaz, K. (2008b). Concept search in Urdu. In *Proceedings of the 2nd PhD workshop on Information and Knowledge Management (PIKM'08), California, USA*, pages 33–40. ACM.

[195] Riaz, K. (2010). Rule-based named entity recognition in Urdu. In *Proceedings of the 2010 Named Entities WorkShop (NEWS'10), Uppsala, Sweden*, pages 126–135. Association for Computational Linguistics (ACL).

[196] Riaz, K. (2012). Comparison of Hindi and Urdu in computational context. *International Journal of Computational Linguistics and Natural Language Processing (IJCLNLP)*, 1(3):92–97.

[197] Rivest, R. L. (1987). Learning decision lists. *Machine learning*, 2(3):229–246.

[198] Rizzo, G. and Troncy, R. (2012). NERD: A framework for unifying Named Entity Recognition and Disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12), Avignon, France*, pages 73–76. Association for Computational Linguistics (ACL).

[199] Roget, P. M. (2008). *Roget'S International Thesaurus, 3/E*. Oxford and IBH Publishing.

[200] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, (EMNLP'15), Lisbon, Portugal*, pages 1–11. Association for Computational Linguistics (ACL).

[201] Russo, F., Di Bella, S., Bonnici, V., Laganà, A., Rainaldi, G., Pellegrini, M., Pulvirenti, A., Giugno, R., and Ferro, A. (2014). A knowledge base for the discovery of function, diagnostic potential and drug effects on cellular and extracellular miRNAs. *BMC genomics*, 15(3):1–7.

[202] Saeed, A., Nawab, R. M. A., Stevenson, M., and Rayson, P. (2018). A word sense disambiguation corpus for Urdu. *Language Resources and Evaluation*, 39(1):1–22.

[203] Saeed, A., Nawab, R. M. A., Stevenson, M., and Rayson, P. (2019a). A sense annotated corpus for all-words urdu word sense disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(4):40.

[204] Saeed, A., Nawab, R. M. A., Stevenson, M., and Rayson, P. (2019b). A word sense disambiguation corpus for urdu. *Language Resources and Evaluation*, 53(3):397–418.

[205] Sajjad, H. and Schmid, H. (2009). Tagging Urdu Text with Parts of Speech: A Tagger Comparison. In *Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, Greece*, pages 692–700. Association for Computational Linguistics (ACL).

[206] Salathé, M. and Khandelwal, S. (2011). Assessing vaccination sentiments with online social media: Implications for infectious disease dynamics and control. *PLoS computational biology*, 7(10):1–7.

[207] Sanfilippo, A. (1994). LKB encoding of lexical knowledge. In *Inheritance, defaults and the lexicon*, pages 190–222. Cambridge University Press.

[208] Saputra, A. et al. (2010). Building synsets for Indonesian WordNet with monolingual lexical resources. In *Proceedings of the International Conference on Asian Language Processing (IALP'10), Harbin, China*, pages 297–300. IEEE.

[209] Scharl, A., Hubmann-Haidvogel, A., Jones, A., Fischl, D., Kamolov, R., Weichselbraun, A., and Rafelsberger, W. (2016). Analyzing the public discourse on works of fiction–Detection and visualization of emotion in online coverage about HBO's Game of Thrones. *Information Processing & Management*, 52(1):129–138.

[210] Schmid, H. (1994). Probabilistic part-of-speech tagging using Decision Trees. In *Proceedings of the international conference on new methods in language processing (NeMLaP), Manchester, UK.*, volume 12, pages 44–49. Citeseer.

[211] Schmid, H. and Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 08), Manchester, UK*, volume 1, pages 777–784. Association for Computational Linguistics (ACL).

[212] Schmidt, R. L. (1999). *Urdu, an Essential Grammar (Routledge Essential Grammars)*, volume 1. Psychology Press.

[213] Scott, P., Paul, R., Dawn, A., Francesca, B., Carmen, D., Mahmoud, E.-H., Ricardo-María, J., Dawn, K., Michal, K., Laura, L., Rao Muhammad, A. N., Jawad, S., Phoey, L. T., and Olga, M. (2016). Lexical coverage evaluation of large-scale multilingual semantic lexicons for twelve languages. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), Portoroẑ, Slovenia*, pages 2614–2619. European Language Resources Association (ELRA).

[214] Semino, E., Demjén, Z., Demmen, J., Koller, V., Payne, S., Hardie, A., and Rayson, P. (2017). The online use of Violence and Journey metaphors by patients with cancer, as compared with health professionals: A mixed methods study. *BMJ supportive & palliative care*, 7(1):60–66.

[215] Shah, R., Lin, B., Gershman, A., and Frederking, R. (2010). SYNERGY: A named entity recognition system for resource-scarce languages such as Swahili using online machine translation. In *Proceedings of the Second Workshop on African Language Technology (AfLaT '10), Valletta, Malta*, pages 21–26.

[216] Sharjeel, M., Nawab, R. M. A., and Rayson, P. (2017). COUNTER: COrpus of Urdu News TExt Reuse. *Language Resources and Evaluation*, 51(3):777–803.

[217] Sharoff, S., Babych, B., Rayson, P., Mudraya, O., and Piao, S. (2006). ASSIST: Automated semantic assistance for translators. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations (EACL'06), Trento, Italy*, pages 139–142. Association for Computational Linguistics.

[218] Shawar, B. A. and Atwell, E. (2003). Using dialogue corpora to train a Chatbot. In *Proceedings of the Corpus Linguistics 2003 conference, Lancaster, UK*, pages 681–690. Lancaster University, UK.

[219] Simm, W., Ferrario, M.-A., Piao, S., Whittle, J., and Rayson, P. (2010). Classification of short text comments by sentiment and actionability for voiceyourview. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 552–557. IEEE.

[220] Spyromitros, E., Tsoumakas, G., and Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. In *Proceedings of the 5th Hellenic conference on artificial intelligence (SETN'08), Berlin, Germany*, pages 401–406. Springer.

[221] Steven, B. and Edward, L. (2006). NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions, Sydney, Australia*, pages 69–72. Association for Computational Linguistics (ACL).

[222] Stevenson, M. and Wilks, Y. (2001). The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–349.

[223] Stoyanov, V., Gilbert, N., Cardie, C., and Riloff, E. (2009). Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the The Asian Federation of Natural Language Processing (ACL-AFNLP'09), Suntec, Singapore*, volume 2, pages 656–664. Association for Computational Linguistics (ACL).

[224] Tablan, V., Roberts, I., Cunningham, H., and Bontcheva, K. (2013). GATE-Cloud. net: A platform for large-scale, open-source text processing on the cloud. *Philosophical Transactions of the Royal Society Series A, Mathematical, Physical, and Engineering Sciences*, 371(1983):20120071.

[225] Tafseer, A., Urooj, S., Hussain, S., Mustafa, A., Parveen, R., Adeeba, F., Hautli, A., and Butt, M. (2014). The CLE Urdu POS Tagset. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland*, pages 2920–2925. European Language Resources Association (ELRA).

[226] Tagg, C., Baron, A., and Rayson, P. (2014). "i didn't spel that wrong did i. Oops": Analysis and normalisation of SMS spelling variation. *In L.-A. Cougnon C. Fairon (Eds.), SMS Communication: A Linguistic Approach, Lingvisticæ Investigationes*, 35(2):217–237.

[227] Taïani, F., Grace, P., Coulson, G., and Blair, G. (2008). Past and future of reflective middleware: Towards a corpus-based impact analysis. In *Proceedings of the 7th workshop on Reflective and adaptive middleware (Middleware'08), Leuven, Belgium*, pages 41–46. ACM.

[228] Tang, D., Qin, B., and Liu, T. (2015). Learning semantic representations of users and products for document level sentiment classification. In *The 53rd Annual Meeting of the Association for Computational Linguistic and the 7th International*

Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP'15), Beijing, China, pages 1014–1023. Association for Computational Linguistics (ACL).

[229] Thede, S. M. and Harper, M. P. (1999). A second-order hidden Markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL'99), College Park, Maryland*, pages 175–182. Association for Computational Linguistics (ACL).

[230] Thomas, J. and Wilson, A. (1996). Methodologies for studying a corpus of doctor-patient interaction. *In J. Thomas and M. Short (eds.) Using corpora for language research, Longman, London*, pages 92–109.

[231] Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. (2008). Multi-label Classification of Music into Emotions. In *Proceedings of the 9th International Conference of Music Information Retrieval (ISMIR'08), Philadelphia, USA*, volume 8, pages 325–330.

[232] Tsatsaronis, G., Vazirgiannis, M., and Androutsopoulos, I. (2007). Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India*, volume 7, pages 1725–1730.

[233] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

[234] Urooj, S., Shams, S., Hussain, S., and Adeeba, F. (2014). Sense Tagged CLE Urdu Digest Corpus. In *Proceedings of the Conference on Language and Technology (CLT'14), Karachi, Pakistan*, pages 1–8. Centre for Language Engineering (CLE).

[235] Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H. (2008). Decision Trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214.

[236] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

[237] Vossen, P. (1998). A multilingual database with lexical semantic networks. *Dordrecht: Kluwer Academic Publishers.*, 10(1):978–994.

[238] Vossen, P., Izquierdo, R., and Görög, A. (2013). Dutchsemcor: In quest of the ideal sense-tagged corpus. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP'13), Hissar, Bulgaria*, pages 710–718.

[239] Vulić, I. and Moens, M.-F. (2015). Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), (ACL-IJCNLP'15), Beijing, China*, volume 2, pages 719–725.

[240] Wang, K., Thrasher, C., Viegas, E., Li, X., and Hsu, B.-j. P. (2010). An overview of Microsoft Web N-gram corpus and applications. In *Proceedings of the NAACL HLT 2010: Demonstration Session, Los Angeles, California*, pages 45–48. Association for Computational Linguistics (ACL).

[241] Weston, J., Bordes, A., Yakhnenko, O., and Usunier, N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13), Washington, USA*, pages 133–137. ACL-SIGDAT (the Association for Computational Linguistics special interest Group for linguistic data and corpus-based approaches to natural language processing).

[242] Wicaksono, A. F. and Purwarianti, A. (2010). HMM based part-of-speech tagger for Bahasa Indonesia. In *Proceedings of the Fourth International MALINDO Workshop, Jakarta, Indonesia*, pages 1–7.

[243] Widdows, D., Dorow, B., and Chan, C.-K. (2002). Using parallel corpora to enrich multilingual lexical resources. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02), Canary Islands, Spain*, pages 240–245. ELRA.

[244] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Towards universal paraphrastic sentence embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR'16), San Diego, CA, USA*, pages 1–19. the Computational and Biological Learning Society.

[245] Wilks, Y. (1973). Preference semantics. Technical report, Stanford University, Department of Computer Science.

[246] Wilson, A. and Leech, G. (1993). Automatic content analysis and the stylistic analysis of prose literature. *Revue: Informatique et Statistique dans les Sciences Humaines*, 29:219–234.

[247] Wilson, A. and Rayson, P. (1993). Automatic content analysis of spoken discourse: A report on work in progress. *Corpus based computational linguistics*, pages 215–226.

[248] Xiao, R., McEnery, A., Baker, J., and Hardie, A. (2004). Developing Asian language corpora: Standards and practice. In *Proceedings of the 4th Workshop on Asian Language Resources (ALR'04) affiliated to IJC-NLP-04, the 1st International Joint Conference on Natural Language Processing, Hainan Island, China*, pages 1–8. Asia Federation of Natural Language Processing (AFNLP).

[249] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT 2016, San Diego, California*, pages 1480–1489. Association for Computational Linguistics (ACL).

[250] Yi, C. (2015). An English POS tagging approach based on Maximum Entropy. In *Proceedings of the International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS'15), Halong Bay, Vietnam.*, pages 81–84. IEEE.

[251] Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING'16), Osaka, Japan*, pages 1374–1385. Association of Natural Language Processing (ANLP).

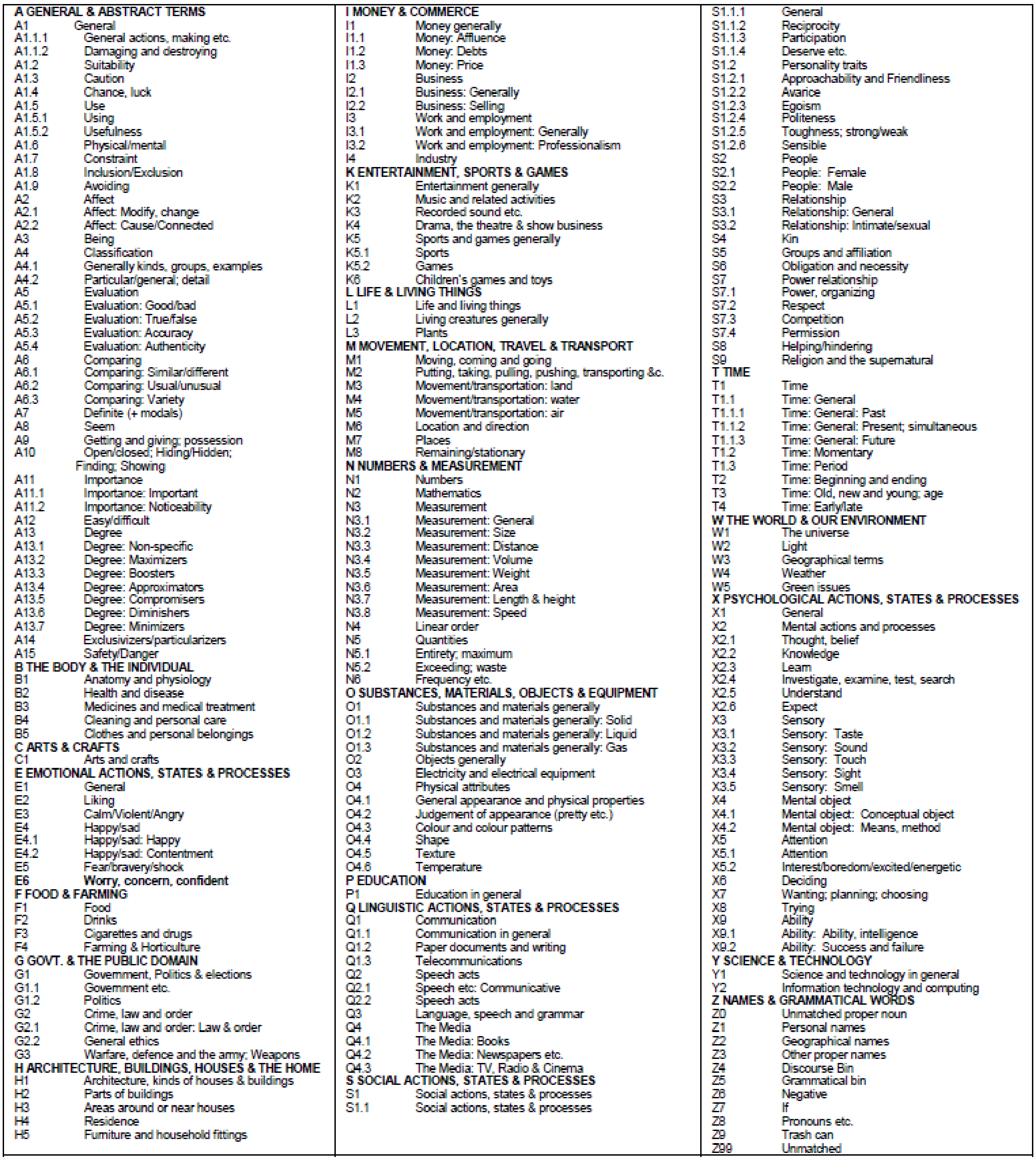
[252] Zeng, Z., Liang, N., Yang, X., and Hoi, S. (2018). Multi-target deep neural networks: Theoretical analysis and implementation. *Neurocomputing*, 273:634–642.

[253] Zhang, M.-L. and Zhou, Z.-H. (2014). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.

[254] Zia, H. B., Raza, A. A., and Athar, A. (2018). Urdu word segmentation using conditional random fields (crfs). *arXiv preprint arXiv:1806.05432*.

Appendix A

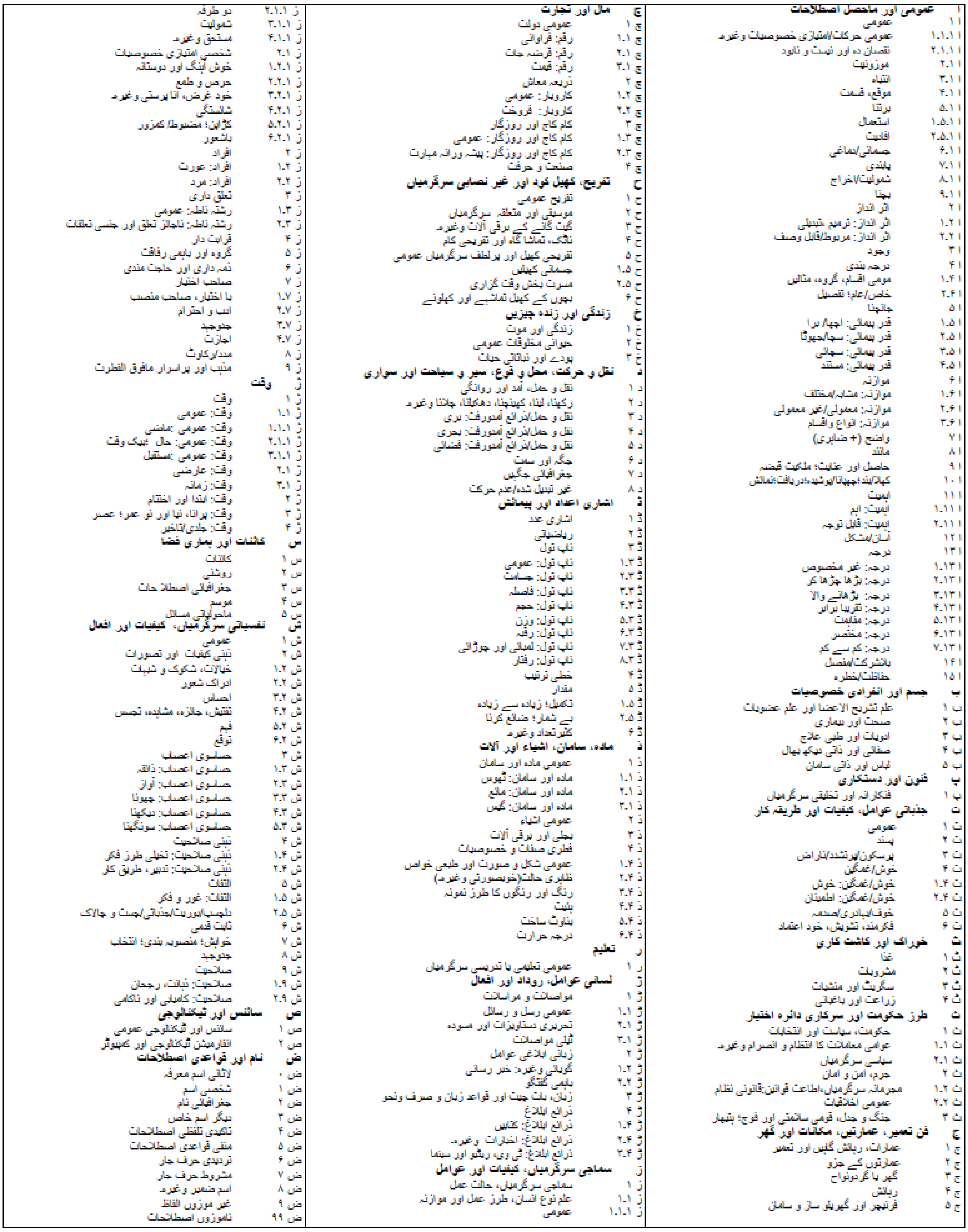
USAS Semantic Tagset in English



A GENERAL & ABSTRACT TERMS	
A1	General
A1.1.1	General actions, making etc.
A1.1.2	Damaging and destroying
A1.2	Suitability
A1.3	Caution
A1.4	Chance, luck
A1.5	Use
A1.5.1	Using
A1.5.2	Usefulness
A1.6	Physical/mental
A1.7	Constraint
A1.8	Inclusion/Exclusion
A1.9	Avoiding
A2	Affect
A2.1	Affect: Modify, change
A2.2	Affect: Cause/Connected
A3	Being
A4	Classification
A4.1	Generally kinds, groups, examples
A4.2	Particular/general; detail
A5	Evaluation
A5.1	Evaluation: Good/bad
A5.2	Evaluation: True/false
A5.3	Evaluation: Accuracy
A5.4	Evaluation: Authenticity
A6	Comparing
A6.1	Comparing: Similar/different
A6.2	Comparing: Usual/unusual
A6.3	Comparing: Variety
A7	Definite (+ modals)
A8	Seem
A9	Getting and giving; possession
A10	Open/closed; Hiding/Hidden; Finding; Showing
A11	Importance
A11.1	Importance: Important
A11.2	Importance: Noticeability
A12	Easy/difficult
A13	Degree
A13.1	Degree: Non-specific
A13.2	Degree: Maximizers
A13.3	Degree: Boosters
A13.4	Degree: Approximators
A13.5	Degree: Compromisers
A13.6	Degree: Diminishers
A13.7	Degree: Minimizers
A14	Exclusivizers/particularizers
A15	Safety/Danger
B THE BODY & THE INDIVIDUAL	
B1	Anatomy and physiology
B2	Health and disease
B3	Medicines and medical treatment
B4	Cleaning and personal care
B5	Clothes and personal belongings
C ARTS & CRAFTS	
C1	Arts and crafts
E EMOTIONAL ACTIONS, STATES & PROCESSES	
E1	General
E2	Liking
E3	Calm/Violent/Angry
E4	Happy/sad
E4.1	Happy/sad: Happy
E4.2	Happy/sad: Contentment
E5	Fear/bravery/shock
E6	**Worry, concern, confident**
F FOOD & FARMING	
F1	Food
F2	Drinks
F3	Cigarettes and drugs
F4	Farming & Horticulture
G GOVT. & THE PUBLIC DOMAIN	
G1	Government, Politics & elections
G1.1	Government etc.
G1.2	Politics
G2	Crime, law and order
G2.1	Crime, law and order: Law & order
G2.2	General ethics
G3	Warfare, defence and the army; Weapons
H ARCHITECTURE, BUILDINGS, HOUSES & THE HOME	
H1	Architecture, kinds of houses & buildings
H2	Parts of buildings
H3	Areas around or near houses
H4	Residence
H5	Furniture and household fittings
I MONEY & COMMERCE	
I1	Money generally
I1.1	Money: Affluence
I1.2	Money: Debts
I1.3	Money: Price
I2	Business
I2.1	Business: Generally
I2.2	Business: Selling
I3	Work and employment
I3.1	Work and employment: Generally
I3.2	Work and employment: Professionalism
I4	Industry
K ENTERTAINMENT, SPORTS & GAMES	
K1	Entertainment generally
K2	Music and related activities
K3	Recorded sound etc.
K4	Drama, the theatre & show business
K5	Sports and games generally
K5.1	Sports
K5.2	Games
K6	Children's games and toys
L LIFE & LIVING THINGS	
L1	Life and living things
L2	Living creatures generally
L3	Plants
M MOVEMENT, LOCATION, TRAVEL & TRANSPORT	
M1	Moving, coming and going
M2	Putting, taking, pulling, pushing, transporting &c.
M3	Movement/transportation: land
M4	Movement/transportation: water
M5	Movement/transportation: air
M6	Location and direction
M7	Places
M8	Remaining/stationary
N NUMBERS & MEASUREMENT	
N1	Numbers
N2	Mathematics
N3	Measurement
N3.1	Measurement: General
N3.2	Measurement: Size
N3.3	Measurement: Distance
N3.4	Measurement: Volume
N3.5	Measurement: Weight
N3.6	Measurement: Area
N3.7	Measurement: Length & height
N3.8	Measurement: Speed
N4	Linear order
N5	Quantities
N5.1	Entirety; maximum
N5.2	Exceeding; waste
N6	Frequency etc.
O SUBSTANCES, MATERIALS, OBJECTS & EQUIPMENT	
O1	Substances and materials generally
O1.1	Substances and materials generally: Solid
O1.2	Substances and materials generally: Liquid
O1.3	Substances and materials generally: Gas
O2	Objects generally
O3	Electricity and electrical equipment
O4	Physical attributes
O4.1	General appearance and physical properties
O4.2	Judgement of appearance (pretty etc.)
O4.3	Colour and colour patterns
O4.4	Shape
O4.5	Texture
O4.6	Temperature
P EDUCATION	
P1	Education in general
Q LINGUISTIC ACTIONS, STATES & PROCESSES	
Q1	Communication
Q1.1	Communication in general
Q1.2	Paper documents and writing
Q1.3	Telecommunications
Q2	Speech acts
Q2.1	Speech etc: Communicative
Q2.2	Speech acts
Q3	Language, speech and grammar
Q4	The Media
Q4.1	The Media: Books
Q4.2	The Media: Newspapers etc.
Q4.3	The Media: TV, Radio & Cinema
S SOCIAL ACTIONS, STATES & PROCESSES	
S1	Social actions, states & processes
S1.1	Social actions, states & processes
S1.1.1	General
S1.1.2	Reciprocity
S1.1.3	Participation
S1.1.4	Deserve etc.
S1.2	Personality traits
S1.2.1	Approachability and Friendliness
S1.2.2	Avarice
S1.2.3	Egoism
S1.2.4	Politeness
S1.2.5	Toughness; strong/weak
S1.2.6	Sensible
S2	People
S2.1	People: Female
S2.2	People: Male
S3	Relationship
S3.1	Relationship: General
S3.2	Relationship: Intimate/sexual
S4	Kin
S5	Groups and affiliation
S6	Obligation and necessity
S7	Power relationship
S7.1	Power, organizing
S7.2	Respect
S7.3	Competition
S7.4	Permission
S8	Helping/hindering
S9	Religion and the supernatural
T TIME	
T1	Time
T1.1	Time: General
T1.1.1	Time: General: Past
T1.1.2	Time: General: Present; simultaneous
T1.1.3	Time: General: Future
T1.2	Time: Momentary
T1.3	Time: Period
T2	Time: Beginning and ending
T3	Time: Old, new and young; age
T4	Time: Early/late
W THE WORLD & OUR ENVIRONMENT	
W1	The universe
W2	Light
W3	Geographical terms
W4	Weather
W5	Green issues
X PSYCHOLOGICAL ACTIONS, STATES & PROCESSES	
X1	General
X2	Mental actions and processes
X2.1	Thought, belief
X2.2	Knowledge
X2.3	Learn
X2.4	Investigate, examine, test, search
X2.5	Understand
X2.6	Expect
X3	Sensory
X3.1	Sensory: Taste
X3.2	Sensory: Sound
X3.3	Sensory: Touch
X3.4	Sensory: Sight
X3.5	Sensory: Smell
X4	Mental object
X4.1	Mental object: Conceptual object
X4.2	Mental object: Means, method
X5	Attention
X5.1	Attention
X5.2	Interest/boredom/excited/energetic
X6	Deciding
X7	Wanting; planning; choosing
X8	Trying
X9	Ability
X9.1	Ability: Ability, intelligence
X9.2	Ability: Success and failure
Y SCIENCE & TECHNOLOGY	
Y1	Science and technology in general
Y2	Information technology and computing
Z NAMES & GRAMMATICAL WORDS	
Z0	Unmatched proper noun
Z1	Personal names
Z2	Geographical names
Z3	Other proper names
Z4	Discourse Bin
Z5	Grammatical bin
Z6	Negative
Z7	If
Z8	Pronouns etc.
Z9	Trash can
Z99	Unmatched

Appendix B

USAS Semantic Tagset in Urdu



ا	**عمومی اور ماحصل اصطلاحات**
ا ۱	عمومی
ا ۱.۱.۱	عمومی حرکات/امتیازی خصوصیات وغیرہ
ا ۲.۱.۱	نقصان دہ اور نیست و نابود
ا ۲.۱	موزونیت
ا ۳.۱	اشتباہ
ا ۴.۱	موقع، قسمت
ا ۵.۱	برتنا
ا ۱.۵.۱	استعمال
ا ۲.۵.۱	افادیت
ا ۶.۱	جسمانی/دماغی
ا ۷.۱	پابندی
ا ۸.۱	شمولیت/اخراج
ا ۹.۱	بچنا
ا ۲	اثر انداز
ا ۱.۲	اثر انداز: ترمیم ،تبدیلی
ا ۲.۲	اثر انداز: مربوط/قابل وصف
ا ۳	وجود
ا ۴	درجہ بندی
ا ۱.۴	مومی اقسام، گروہ، مثالیں
ا ۲.۴	خاص/عام؛ تفصیل
ا ۵	جانچنا
ا ۱.۵	قدر پیمائی: اچھا/ برا
ا ۲.۵	قدر پیمائی: سچا/جھوٹا
ا ۳.۵	قدر پیمائی: سچائی
ا ۴.۵	قدر پیمائی: مستند
ا ۶	موازنہ
ا ۱.۶	موازنہ: مشابہ/مختلف
ا ۲.۶	موازنہ: معمولی/غیر معمولی
ا ۳.۶	موازنہ: انواع واقسام
ا ۷	واضح (+ ضابری)
ا ۸	مانند
ا ۹	حاصل اور عنایت؛ ملکیت قبضہ
ا ۱۰	کھلا/بند؛چھپانا/پوشیدہ؛دریافت؛نمائش
ا ۱۱	اہمیت
ا ۱.۱۱	اہمیت: اہم
ا ۲.۱۱	اہمیت: قابل توجہ
ا ۱۲	آسان/مشکل
ا ۱۳	درجہ
ا ۱.۱۳	درجہ: غیر مخصوص
ا ۲.۱۳	درجہ: بڑھا چڑھا کر
ا ۳.۱۳	درجہ: بڑھانے والا
ا ۴.۱۳	درجہ: تقریبا برابر
ا ۵.۱۳	درجہ: مقابت
ا ۶.۱۳	درجہ: مختصر
ا ۷.۱۳	درجہ: کم سے کم
ا ۱۴	بالشرکت/منفصل
ا ۱۵	حفاظت/خطرہ
ب	**جسم اور انفرادی خصوصیات**
ب ۱	علم تشریح الاعضا اور علم عضویات
ب ۲	صحت اور بیماری
ب ۳	ادویات اور طبی علاج
ب ۴	صفائی اور ذاتی دیکھ بھال
ب ۵	لباس اور ذاتی سامان
پ	**فنون اور دستکاری**
پ ۱	فنکارانہ اور تخلیقی سرگرمیاں
ت	**جذباتی عوامل، کیفیات اور طریقہ کار**
ت ۱	عمومی
ت ۲	پسند
ت ۳	پرسکون/پرتشدد/ناراض
ت ۴	خوش/غمگین
ت ۱.۴	خوش/غمگین: خوش
ت ۲.۴	خوش/غمگین: اطمینان
ت ۵	خوف/بہادری/صدمہ
ت ۶	فکرمند، تشویش، خود اعتماد
ث	**خوراک اور کاشت کاری**
ث ۱	غذا
ث ۲	مشروبات
ث ۳	سگریٹ اور منشیات
ث ۴	زراعت اور باغبانی
ٹ	**طرز حکومت اور سرکاری دائرہ اختیار**
ٹ ۱	حکومت، سیاست اور انتخابات
ٹ ۱.۱	عوامی معاملات کا انتظام و انصرام وغیرہ
ٹ ۲.۱	سیاسی سرگرمیاں
ٹ ۲	جرم، امن و امان
ٹ ۱.۲	مجرمانہ سرگرمیاں،اطاعت قوانین؛قانونی نظام
ٹ ۲.۲	عمومی اخلاقیات
ٹ ۳	جنگ و جدل، قومی سلامتی اور فوج؛ ہتھیار
ج	**فن تعمیر، عمارتیں، مکانات اور گھر**
ج ۱	عمارات، رہائش گاہیں اور تعمیر
ج ۲	عمارتوں کے جزو
ج ۳	گھر یا گردونواح
ج ۴	رہائش
ج ۵	فرنیچر اور گھریلو ساز و سامان
چ	**مال اور تجارت**
چ ۱	عمومی دولت
چ ۱.۱	رقم: فراوانی
چ ۲.۱	رقم: قرضہ جات
چ ۳.۱	رقم: قیمت
چ ۲	ذریعہ معاش
چ ۱.۲	کاروبار: عمومی
چ ۲.۲	کاروبار: فروخت
چ ۳	کام کاج اور روزگار
چ ۱.۳	کام کاج اور روزگار: عمومی
چ ۲.۳	کام کاج اور روزگار: پیشہ ورانہ مہارت
چ ۴	صنعت و حرفت
ح	**تفریح، کھیل کود اور غیر نصابی سرگرمیاں**
ح ۱	تفریح عمومی
ح ۲	موسیقی اور متعلقہ سرگرمیاں
ح ۳	گیت گانے کے برقی آلات وغیرہ
ح ۴	ناٹک، تماشا گاہ اور تفریحی کام
ح ۵	تفریحی کھیل اور پرلطف سرگرمیاں عمومی
ح ۱.۵	جسمانی کھیلیں
ح ۲.۵	مسرت بخش وقت گزاری
ح ۶	بچوں کے کھیل تماشے اور کھلونے
خ	**زندگی اور زندہ چیزیں**
خ ۱	زندگی اور موت
خ ۲	حیوانی مخلوقات عمومی
خ ۳	پودے اور نباتاتی حیات
د	**نقل و حرکت، محل و قوع، سیر و سیاحت اور سواری**
د ۱	نقل و حمل، آمد اور روانگی
د ۲	رکھنا، لینا، کھینچنا، دھکیلنا، چلانا وغیرہ
د ۳	نقل و حمل/ذرائع آمدورفت: بری
د ۴	نقل و حمل/ذرائع آمدورفت: بحری
د ۵	نقل و حمل/ذرائع آمدورفت: فضائی
د ۶	جگہ اور سمت
د ۷	جغرافیائی جگہیں
د ۸	غیر تبدیل شدہ/عدم حرکت
ڈ	**اشاری اعداد اور پیمائش**
ڈ ۱	اشاری عدد
ڈ ۲	ریاضیاتی
ڈ ۳	ناپ تول
ڈ ۱.۳	ناپ تول: عمومی
ڈ ۲.۳	ناپ تول: جسامت
ڈ ۳.۳	ناپ تول: فاصلہ
ڈ ۴.۳	ناپ تول: حجم
ڈ ۵.۳	ناپ تول: وزن
ڈ ۶.۳	ناپ تول: رقبہ
ڈ ۷.۳	ناپ تول: لمبائی اور چوڑائی
ڈ ۸.۳	ناپ تول: رفتار
ڈ ۴	خطی ترتیب
ڈ ۵	مقدار
ڈ ۱.۵	تکمیل؛ زیادہ سے زیادہ
ڈ ۲.۵	بے شمار؛ ضائع کرنا
ڈ ۶	کثیرتعداد وغیرہ
ذ	**مادہ، سامان، اشیاء اور آلات**
ذ ۱	عمومی مادہ اور سامان
ذ ۱.۱	مادہ اور سامان: ٹھوس
ذ ۲.۱	مادہ اور سامان: مائع
ذ ۳.۱	مادہ اور سامان: گیس
ذ ۲	عمومی اشیاء
ذ ۳	بجلی اور برقی آلات
ذ ۴	فطری صفات و خصوصیات
ذ ۱.۴	عمومی شکل و صورت اور طبعی خواص
ذ ۲.۴	ظاہری حالت(خوبصورتی وغیرہ)
ذ ۳.۴	رنگ اور رنگوں کا طرز نمونہ
ذ ۴.۴	ہیئت
ذ ۵.۴	بناوٹ ساخت
ذ ۶.۴	درجہ حرارت
ر	**تعلیم**
ر ۱	عمومی تعلیمی یا تدریسی سرگرمیاں
ڑ	**لسانی عوامل، روداد اور افعال**
ڑ ۱	مواصلات و مراسلات
ڑ ۱.۱	عمومی رسل و رسائل
ڑ ۲.۱	تحریری دستاویزات اور مسودہ
ڑ ۳.۱	ٹیلی مواصلات
ڑ ۲	زبانی ابلاغی عوامل
ڑ ۱.۲	گویائی وغیرہ: خبر رسانی
ڑ ۲.۲	باہمی گفتگو
ڑ ۳	زبان، بات چیت اور قواعد زبان و صرف ونحو
ڑ ۴	ذرائع ابلاغ
ڑ ۱.۴	ذرائع ابلاغ: کتابیں
ڑ ۲.۴	ذرائع ابلاغ: اخبارات وغیرہ
ڑ ۳.۴	ذرائع ابلاغ: ٹی وی، ریڈیو اور سینما
ز	**سماجی سرگرمیاں، کیفیات اور عوامل**
ز ۱	سماجی سرگرمیاں، حالت عمل
ز ۱.۱	علم نوع انسان، طرز عمل اور موازنہ
ز ۱.۱.۱	عمومی
ز ۲.۱.۱	دو طرفہ
ز ۳.۱.۱	شمولیت
ز ۴.۱.۱	مستحق وغیرہ
ز ۲.۱	شخصی امتیازی خصوصیات
ز ۱.۲.۱	خوش آہنگ اور دوستانہ
ز ۲.۲.۱	حرص و طمع
ز ۳.۲.۱	خود غرض، انا پرستی وغیرہ
ز ۴.۲.۱	شائستگی
ز ۵.۲.۱	کڑاپن؛ مضبوط/ کمزور
ز ۶.۲.۱	باشعور
ز ۲	افراد
ز ۱.۲	افراد: عورت
ز ۲.۲	افراد: مرد
ز ۳	تعلق داری
ز ۱.۳	رشتہ ناطہ: عمومی
ز ۲.۳	رشتہ ناطہ: ناجائز تعلق اور جنسی تعلقات
ز ۴	قرابت دار
ز ۵	گروہ اور باہمی رفاقت
ز ۶	ذمہ داری اور حاجت مندی
ز ۷	صاحب اختیار
ز ۱.۷	با اختیار، صاحب منصب
ز ۲.۷	ادب و احترام
ز ۳.۷	جدوجہد
ز ۴.۷	اجازت
ز ۸	مدد/رکاوٹ
ز ۹	مذہب اور پراسرار مافوق الفطرت
ژ	**وقت**
ژ ۱	وقت
ژ ۱.۱	وقت: عمومی
ژ ۱.۱.۱	وقت: عمومی :ماضی
ژ ۲.۱.۱	وقت: عمومی: حال ایک وقت
ژ ۳.۱.۱	وقت: عمومی :مستقبل
ژ ۲.۱	وقت: عارضی
ژ ۳.۱	وقت: زمانہ
ژ ۲	وقت: ابتدا اور اختتام
ژ ۳	وقت: پرانا، نیا اور نو عمر؛ عصر
ژ ۴	وقت: جلدی/تاخیر
س	**کائنات اور ہماری فضا**
س ۱	کائنات
س ۲	روشنی
س ۳	جغرافیائی اصطلا حات
س ۴	موسم
س ۵	ماحولیاتی مسائل
ش	**نفسیاتی سرگرمیاں، کیفیات اور افعال**
ش ۱	عمومی
ش ۲	ذہنی کیفیات اور تصورات
ش ۱.۲	خیالات، شکوک و شبہات
ش ۲.۲	ادراک شعور
ش ۳.۲	احساس
ش ۴.۲	تفتیش، جائزہ، مشاہدہ، تجسس
ش ۵.۲	فہم
ش ۶.۲	توقع
ش ۳	حساسوی اعصاب
ش ۱.۳	حساسوی اعصاب: ذائقہ
ش ۲.۳	حساسوی اعصاب: آواز
ش ۳.۳	حساسوی اعصاب: چھونا
ش ۴.۳	حساسوی اعصاب: دیکھنا
ش ۵.۳	حساسوی اعصاب: سونگھنا
ش ۴	ذہنی صلاحیت
ش ۱.۴	ذہنی صلاحیت: تخیلی طرز فکر
ش ۲.۴	ذہنی صلاحیت: تدبیر، طریق کار
ش ۵	التفات
ش ۱.۵	التفات: غور و فکر
ش ۲.۵	دلچسپی/بوریت/جذباتی/چست و چالاک
ش ۶	ثابت قدمی
ش ۷	خواہش؛ منصوبہ بندی؛ انتخاب
ش ۸	جدوجہد
ش ۹	صلاحیت
ش ۱.۹	صلاحیت: ذہانت، رجحان
ش ۲.۹	صلاحیت: کامیابی اور ناکامی
ص	**سائنس اور ٹیکنالوجی**
ص ۱	سائنس اور ٹیکنالوجی عمومی
ص ۲	انفارمیشن ٹیکنالوجی اور کمپیوٹر
ض	**نام اور قواعدی اصطلاحات**
ض ۰	لاثانی اسم معرفہ
ض ۱	شخصی اسم
ض ۲	جغرافیائی نام
ض ۳	دیگر اسم خاص
ض ۴	تاکیدی تلفظی اصطلاحات
ض ۵	منقی قواعدی اصطلاحات
ض ۶	تردیدی حرف جار
ض ۷	مشروط حرف جار
ض ۸	اسم ضمیر وغیرہ
ض ۹	غیر موزوں الفاظ
ض ۹۹	ناموزوں اصطلاحات