Urdu Noun Phrase Chunking

MS Thesis

Submitted in Partial Fulfillment
of the Requirements for
the Degree of

Master of Science (Computer Science)

at the

National University of Computer and Emerging Sciences

by

Shahid Siddiq
MS (CS) 06-0817
2009

Approved:

____

Head
(Department of Computer Sciences)

__________________ 20 ________

Approved by Committee Members:

Advisor

____

Dr. Sarmad Hussain
Professor
National University of Computer & Emerging Sciences

Co –Advisor

____

Mr. Shafiq-ur-Rahman
Associate Professor
National University of Computer & Emerging Sciences

Acknowledgements

If the oceans turn into ink and all the wood becomes pen, even then, the praises of Allah Almighty cannot be expressed. Unfeigned and meek thanks are before him, who created the universe and bestowed the mankind with knowledge and wisdom to search for its secret and invigorated me with fortunate and capability to complete my research work on this interesting topic.

The warmest and deepest sense of obligation are to respected advisor Prof. Dr. Sarmad Hussain, for his valuable guidance in this thesis work, kind and humble attitude during different stages of this work.

Many thanks to all of my friends in the field of Natural Language processing for guiding me during experimentation and literature review, particularly to Mr. Asim Ali, PhD student, National University of Computer and Emerging Sciences, Mr. Umer Khalid Qureshi, Center of Research in Urdu Language Processing and Ahmed Muaz, Center of Research in Urdu Language Processing for kind guidance.

Special Thanks to Co-advisor Mr. Shafiq-ur-Rahman for his kind guidance during this work and then review of this report.

Shahid Siddiq

Table of Contents

List of Tables and Figures

1 Introduction

Chunking is a technique used to help in development of natural language processing applications. The technique uses part of speech tags extensively for determining the phrase boundaries. It helps in tasks of machine translation, named entity recognition, information extraction and many other natural language applications. Keeping in view the importance of chunking task, a lot of research has been made for many languages. The aim of this work is to investigate the accuracy of corpus based NP chunking for Urdu language so that further research to get maximum benefits of chunking would be made. Following subsection introduces the reader about organization of the Report.

1.1 Organization of Thesis Report

This report is divided into six sections. Section 2 includes background which consists of parts of speech, POS tagging, different phrases of Urdu, free word order property and case markers of Urdu. Section 3 introduces chunking particularly NP chunking with examples. Section 4 contains techniques, tools, and comparison of tag sets studied as literature review in this work. Section 5 explains current work; it includes motivation, its scope and the problem statement sub-sections to introduce reader about problem of this work and its scope. Second part of this section explains the methodology of this work. It includes detail of experiments, methodology adopted to solve the problem, computational model. The overall architecture explains the whole system of problem in consideration. Section 6 elaborates the results obtained after execution of experiments. It contains evaluation metrics to evaluate the methodology. This section also contains discussion as a subsection to introduce the reader about the analysis of author of report about results. Section 7 concludes this report with conclusion and future directions for future work. At the end references and appendices are placed for further readings.

2 Background

This section is about some concepts and basic building block of languages particularly Urdu language. The input of chunking task is part of speech tags most of the time. Major portion of this section introduces the reader about part of speech tags. Following is the list of language aspects which are discussed in this section:

1. Parts of speech
    a. Parts of speech tags
        i. Parts of speech tags of English
        ii. Parts of speech tags of Urdu
2. Phrases in Urdu
    a. Other Phrases
    b. Noun Phrase
3. Important Characteristics of Urdu
    a. Free word order property
    b. Case Markers

2.1 Parts of Speech

Quirk (1985) explains parts of speech in terms of general classes of words. It is a traditional term for classification of words. For example, nouns, pronouns, verbs, adjectives, adverbs, and prepositions are some major part of speech in English language. He divides POS of English language into two major categories of classes that are: closed word classes and open word classes. Closed word classes include preposition, pronoun, determiner, conjunction and modal verb. Open word classes include nouns, adjectives, full verb and adverb. He separately introduces numerals and interjections

Thomson (1986) categorizes parts of speech for English language into twelve classes as: articles, noun, adjective, adverb, Wh- words, possessive pronoun, personal pronoun, reflexive pronoun, relative pronoun, prepositions, verbs, and auxiliaries.

Platts (1909) claims that Urdu grammarians classify part of speech of Urdu into three main head of Verbs, Nouns and Particles. Conjunctive Participle is classified under the Verbs. Noun class has the substantive, the adjective, the numerical adjective, the personal pronoun, the demonstrative pronoun, the relative pronoun, the interrogative pronoun, the indefinite pronoun, the infinitive pronoun and the deverbal noun. The adverbs, the prepositions, conjunction and interjections are under the term of the particles.

Robins (1989) describes parts of speech in following words:

"The classification of words into lexical categories"

Parts of speech assignment is on the basis of context in which the word is being used. For example,

1. In the sentence, "He heard the running water.", running is an adjective.
2. In the sentence, "He is running.", running is a verb.
3. It can even be a noun. In the sentence, "Running is good for you.", running is a noun.

In above example same word is obtained different parts of speech in different sentences. Word "running" is marked as adjective in sentence 1, verb in sentence 2 and noun in sentence 3. Such an example shows that word solely cannot categorized into parts of speech but using context parts of speech of a word is determined.

2.1.1 Parts of Speech Tags

Most of parts of speech (POS) are common in all languages of world. But some classes of POS are distinct and vary language to language. Parts of speech are extensively used in natural language processing tools and applications development. For the purpose of usage in automated tools, part of speech tags (POS tags) are developed. Thus POS tagging is labeling of words into POS classes for computational tasks. These tags are different for different languages. Discussion on some parts of speech tags in English and Urdu languages is done in the proceeding subsections.

2.1.1.1 Part of Speech Tags of English

There are different parts of speech tagsets for English like Brown Corpus and Penn Tree Bank. The Brown corpus used 87 tags to represent English part of speech tagset. Penn Tree Bank tagset is most widely used tagset consists of 45 tags for English language. An example using Penn Tree Bank POS Tag set is given below:

The <**DT>** task <**NN>** of <**IN>** tagging <**NN>**is <**VBZ>** to <**TO>** assign <**VB>**
part-of-speech <**JJ>** tags <**NNS>** to <**TO>** words <**NNS>** reflecting <**VBG>** syntactic <**JJ>**
category <**NN>**

Following list explains parts of speech for respective POS tags:

Table 1: Some Parts Of Speech with corresponding POS Tags for English using Penn Tree Bank Tagset

Parts of Speech	Parts of Speech Tags
Determiner	<DT>
Noun, Singular of Mass	<NN>
Preposition or Subordinate Conjunction	<IN>
Verb, 3dr Person Singular Present	<VBZ>
To	<TO>
Verb, Base Form	<VB>
Adjective	<JJ>
Noun, Plural	<NNS>
Verb, Gerund or Present Participle	<VBG>

2.1.1.2 Part of Speech Tags of Urdu

Different POS tag sets of Urdu are developed by different groups using different analysis. Hardie (2003) developed 282 tags for Urdu. Sajjad (2007) introduced tag set of 42 tags for Urdu. Recently a new tag set is introduced by Muaz et al (2009) consists of 32 tags. This work uses tag set of Sajjad (2007) because the tag set introduced by Hardie (2003) is large one with low accuracies than tag set of Sajjad (2007). Tag set of Muaz et al (2009) reports better accuracies than Sajjad (2007) but the tag set of Muaz et al(2009) is published during report writing of this work. Some of tags used by Sajjad (2007) are elaborated through following examples:

1- پولیس<NN>کے<P>ہاتھوں<NN>ظلم<NN>و<CC>زیادتی<NN>کی<P>خبروں<NN>نے<P>لوگوں
<NN>کا<P>اعتماد<NN>مجروح<NN>کیا<VB>ہے<TA>۔<SM>پولیس<NN>کو<P>لامحدود<ADJ>
اختیارات<NN>حاصل<NN>ہیں<VB>۔<SM>

2- جنہیں<REP>وہ<PD>تفتیش<NN>کے<P>دوران<NN>بے جا<ADJ>استعمال<NN>کرتی<VB>
ہے<TA>۔<SM>

3- وہ<PP>محض<ADV>مغرب<NN>کو<P>دھوکہ<NN>دینے<VB>کے<P>لئے<NN>ہے<VB>
کیونکہ<SC>اس<PP>سے<SE>قبل<ADV>کاروکاری<NN>پر<P>جو<RD>بل<NN>ہم<PP>نے<P>
پیش<NN>کیا<VB>تھا<TA>وہ<PP>ٹھوس<ADJ>اور<CC>جامع<ADJ>تھا<VB>۔<SM>

4- اسرائیل<PN>کو<P>محفوظ<ADJ>بنانے<VB>کے<P>لئے<NN>جولائی <PN> 2002ء<DATE>میں
<P>ایوان<NN>نمائندگان<NN>نے<P>بھاری<ADV>مالی<ADJ>امداد<NN>فراہم<NN>کر<VB>دی<AA>او

ر<CC>فلسطینی<ADJ>اتھارٹی<NN>کے<P>رہنماؤں<NN>کو<P>سنگین<ADJ>نتائج<NN>کی<P>دھمکیاں<NN>دی<VB>گئیں<AA>-<SM>

5- اس<PD>طرح<NN>مرنے<VB>والے<WALA>افراد<NN>کے<P>عزیز<NN>و<CC>اقرباء<NN>کو<P>کرب<NN>سے<SE>گذرنا<VB>پڑتا<AA>ہے<TA>-<SM>

6- اسرائیلی<ADJ>جریدے<NN>ڈیہائس<PN>نے<P>انکشاف<NN>کیا<VB>ہے<TA>کہ<SC>امریکی<ADJ>صدر<NN>جارج<PN>بش<PN>قدامتپسند<ADJ>یہودیوں<NN>کے<P>ساتھ<NN>گہرے<ADJ>تعلقات<NN>رکھتے<VB>ہیں<TA>-<SM>ان<PD>تعلقات<NN>کو<P>قائم<NN>رکھنے<VB>کے<P>لئے<NN>یہودیوں<NN>نے<P>بھی<I>مسیح<ADJ>موعود<NN>کے<P>نظریئے<NN>کو<P>تسلیم<NN>کر<VB>لیا<AA>ہے<TA>-<SM>

Following table elaborates POS tags in above example and their respective parts of speech:

Table 2: Some Parts of Speech and respective POS tags of tagset developed by Sajjad (2007) for Urdu

Parts of Speech	Parts of Speech Tags
Simple nouns	<NN>
Particles (Semantic Markers)	<P>
Coordinating Conjunction	<CC>
Verb	<VB>
Tense Auxiliary	<TA>
Sentence Marker	<SM>
Adjectives	<ADJ>
Relative Pronoun	<REP>
Personal Pronoun	<PP>
Adverb	<ADV>
Subordinate Conjunction	<SC>
Special Semantic Marker (SE)	<SE>
Relative Demonstrative	<RD>
Proper Noun	<PN>
Date	<DATE>
Aspectual Auxiliary	<AA>
WALA and its inflections	<WALA>
Personal Demonstrative	<PD>

These POS tags are extensively used in this work. All experiments are based on POS tag set Input one or the other way.

2.2 Phrases in Urdu

The list of major phrases for Urdu language is given below:

- Noun phrases (NP): A unit of one or more words in a relationship having noun as head word of the unit
- Verb phrases (VP): A unit of one or more words in a relationship having verb as head word of the unit
- Postpositional phrases (PP): The Postpositional/Prepositional Phrase (PP) is called ترکیبِ جار in Urdu. The trend of Postpositional Phrase is more popular than Prepositions, in Urdu, therefore, most of the times it is discussed as Postpositional Phrase
- Adverbial phrases (ADVP): A unit of one or more words in a relationship having adverb as head word of the unit

The Noun Phrase is termed as اسمی ترکیب in Urdu. It may be so complex that it may comprise other phrases as its constituents, e.g., ترکیبِ توصیفی (Adjectival Phrase) and ترکیبِ اضافی (Genitive Phrase) etc. However, the basic components of Noun Phrase in Urdu are The Noun, The Determiners & Demonstratives, Numerals and other non-word items, The Pronouns, and Adjectives. Following are some examples of non-recursive noun phrases:

1- (سفیدپوش طبقے) کی (زندگی) (مشکلات) کا (شکار) ہے -

2- (ذخیرہاندوزوں)کے (خلاف) (قانون) کو (حرکت) میں لایا جائے -

3- (اسی طرح) سے (ساری گلی) (مٹی) سے بھر گئی ہے اور (اب) (سولنگ) اور (کچی گلی) میں(کوئی فرق) نہیں رہا -

In above example, noun phrases are marked by parenthesis in the sentences. It is to note that without parts of speech it is difficult to mark the noun phrases. If parts of speech tags for each word are marked then detection of noun phrase boundaries is made easy. For example, above sentences are rewritten with their respective POS tags as:

1- (سفیدپوش<ADJ> طبقے<NN>) کی<P> (زندگی <NN>) (مشکلات <NN>) کا<P> (شکار<NN>)
ہے <TA) - <SM>

2- (ذخیرہاندوزوں <NN>)کے<P> (خلاف <NN>) (قانون <NN>) کو <P> (حرکت <NN>) میں <P>
لایا <VB>جائے <AA>- <SM>

3- (اسی <PD>طرح <NN>) سے<SE> (ساری <ADJ>گلی <NN>) (مٹی <NN>) سے
<SE>بھر< ADV > گئی <VB> ہے <TA> اور <CC> (اب <AP>) (سولنگ <NN>) اور <CC> (کچی

<SM> - <VB> رہا <NEG> نہیں (<NN>فرق <KD> کوئی) <P> میں (<NN> گلی<ADJ>

Above example contains all three sentences of previous example annotated with POS tags. It is to note that marking words with their respective POS tags is more convincing while marking the noun phrases or any other phrases in contrast to without POS tags annotation.

This work is related to automated detection of noun phrases boundaries. For the purpose marking boundaries, POS tags are helpful.

2.3 Important Characteristics of Urdu

Free word order property of Urdu, and semantic markers are very important for computational linguists. Because these two properties provide benefits in some situations and are troublesome in some others. These two properties are discussed in proceeding subsections.

2.3.1 Free Word Order Property

Urdu is partially free word order language. This language is free word order because of its feature of case markers. For example:

English Sentence	**Urdu Alternatives of English Sentence**
Ahmad gave the book to Ali.	احمد نے کتاب علی کو دی۔
To Ali the book Ahmed gave	علی کو کتاب احمد نے دی۔
The book Ahmed gave to Ali	کتاب احمد نے علی کو دی۔
The book to Ali Ahmed gave	کتاب علی کو احمد نے دی۔
The book gave Ahmed Ali to	کتاب دی احمد نے علی کو۔
Ahmed to Ali the book gave	احمد نے علی کو کتاب دی۔
To Ali Ahmed the book gave	علی کو احمد نے کتاب دی۔

In above example, it is to note that English is not free word order language because by changing order of the words the meaning has been totally changed, but in Urdu sentence same meanings are conveyed by changing order of words rather constituents. In Urdu this property is present due to semantic markers, which enable it to convey single meaning. Constituents are units which cannot further reorder in sentence. All the above constructions are valid and used in Urdu. It is to note that all variations of sentence convey same meaning as original sentence; the only difference is of variation in emphasis. The main theme of these examples is:

(1) "to give" is the Verb, the predicate of the sentence.

(2) "Ahmad" is the Subject/doer, because of the case marker "نے"

(3) "the book" is the Object, because it is thing being "given".

(4) "Ali" is the 2nd Object (receiver), because of the case marker "کو".

Some other constructions convey the same concept are considered informal but convey the same meaning of the sentence as in original sentence. (A true beauty of this language):

دی کتاب احمد نے علی کو۔

دی کتاب علی کو احمد نے۔

دی احمد نے علی کو کتاب۔, etc.

2.3.2 Case Markers

Croft (2003) explains Case markers as relational morphemes which mark grammatical function of marked word. On the basis of case markers different grammatical relations can be detected. Platts (1909) considers that the relation, in which a noun stands to the other parts of a sentence, is denoted by its "Case (حالت)" This can be explained in the following examples:

(i) لڑکے نے گھوڑا دیکھا (The boy saw the horse)

(ii) گھوڑےنے لڑکا دیکھا (The horse saw the boy)

Both the sentences are valid, and refer to singular item that is seen and the one who has seen is also singular. But the form of the word used is different.

When "the boy" is the doer it is written as "لڑکے", which is a special case of the singular word "لڑکا". This case is caused by the following case marker "نے". Same applied to the "the horse (گھوڑا)". So a word may have as many as ten different cases. However basic seven cases described by Haq (1987) are as under:

1. فاعلی حالت (The Nominative): When the noun occurs as Subject.

[لڑکے] گئے۔

[لڑکیوں] نے کھانا پکایا۔

[لڑکوں] نے کھانا کھایا۔

2. مفعولی حالت (The Accusative): When the noun occurs as Object.

لڑکوں نے [کھانا] کھایا۔

حلیمہ نے [چاند] دیکھا۔

علی نے [احمد] کو دیکھا۔

3. اضافی حالت (The Genitive): Two nouns appear in relationship with each other.

[احمد کا گھوڑا] تندرُست ہے۔

[دروازے کا رنگ] بدل دو۔

[جمیلہ کی بلّی] بیمار ہے۔

4. خبری حالت (The Predicative): When a noun is a news about other noun.

لڑکے [بیمار] ہیں۔

کتا [جانور] ہے۔

5. ندائی حالت (The Vocative): It is used to call someone; typically used in imperative sentences and dialogues.

[لڑکو] کھانا کھاؤ۔

[لڑکے] تمہارا نام کیا ہے؟

6. ظرفی حالت (The Locative): It tells the time, duration, direction, and location etc.

وہ [گھر] میں ہے۔

وہ [شام] تک بیٹھا رہا۔

اس نے [گھڑے] سے شکر نکالی۔

7. طوری حالت (The Ablative): It shows the manner, comparison, cause, etc.

احمد [شوق]سے پڑھتا ہے۔

علی [مجھ] سے بڑا ہے۔

وہ [دولت] سے پڑھا۔

3 Chunking

Abney (1994) describes chunking as a natural phenomenon in the following words:

"(When I Read) (a sentence), (I Read it) (a chunk) (at a time)."

Ramshaw (1995) elaborates chunking as:

"Dividing sentences into non-overlapping phrases on the basis of fairly superficial analysis is called text chunking."

Grover (2007) describes that chunking is identification of word sequences in a sentence to form phrases using shallow syntactic analysis.

Following is an example of Chunking for an English sentence:

Sentence:

`Pierre Vinken, 61 years old, will join the board as a nonexecutive director` *`Nov. 29`*`.`

Following is one of the ways to mark phrase boundaries of above sentence:

[***NP*** Pierre Vinken ***NP***], [***NP*** 61 years ***NP***] old, [***VP*** *will join* ***VP***] [***NP*** *the board* ***NP***] *as* [***NP*** *a nonexecutive director* ***NP***] *[****NP*** *Nov. 29* ***NP****].*

In above example, NP in square brackets explains a separate noun phrase and VP in square brackets explains a separate verb phrase.

Following is another way to mark phrase boundaries using chunk tagging. Above sentence can be written as:

Pierre ***I_NP*** Vinkin ***I_NP, O*** 61 ***I_NP*** years ***I_NP*** old ***O*** , ***O*** will ***I_VP*** join ***I_VP*** the ***I_NP*** board ***I_NP*** as ***O*** a ***I_NP*** nonexecutive ***I_NP*** director ***I_NP*** *Nov.* ***B_NP*** *29* ***I_NP .O***

Each tag is informing about the role of preceding token/ word in above example.

The tag set used in above example is given below:

I_NP: (Inside NP); it means the token is included in the noun phrase

O: Outside NP; it means the token is not included in the noun phrase

B_NP: Inside NP, the preceding token starts a new noun phrase (NP)

I_VP: (Inside VP); it means the token is included in the verb phrase

`B_VP: Inside VP, but the proceeding word is in another VP; it shows beginning of a new verb phrase and boundary of previous verb phrase`

3.1 Benefits of Chunking

Following are some benefits of chunking:

1. Efficient and fast in terms of processing in contrast of full tree parsing as mentioned by Munoz et al (1999)
2. Can be used in development of following applications mentioned by Singh (2001), Rao (2007), Voutelainen (1993), Veenstra et al (1998), Grover (2007), Dalal (2006) and Schmid et al (2000)
    a. Named Entity Recognition (NER)
    b. Information Retrieval (IR)
    c. Question Answer Applications (QA)
    d. Machine Translation (MT)
    e. Speech Synthesis and Recognition
    f. Index Term Generation (ITG)
    g. Syntactic Analysis
3. Stav (2006) considers that chunks reduce search space of solution sets of full parse tree

3.2 NP Chunking

Noun phrase chunking deals with extracting the noun phrases from a sentence. NP chunking is much simpler than parsing but building an accurate and fast NP chunker is a difficult and challenging task.

According to Veenstra et al (1998), NP chunking is conversion of a sentence into non-overlapping noun phrases (called baseNP) using superficial analysis.

Following is an example of a sentence from Urdu Language which includes word tokens with part of speech tags (POS)[1].

جس<REP>کے<P>نتیجے<NN>میں<P>امریکی <ADJ> ریاست <NN>

<PN>پہلے<OR>سے<SE>زیادہ<ADV>غیرمحفوظ<ADJ>ہو<VB> گئ<AA> - <SM>

Following is explanation of above example in context of chunk tags.

جس<B><REP>کے<O><P>نتیجے<B> <NN>میں<O> <P> امریکی<B> <ADJ> ریاست<I>

[1] The tag set for Urdu is taken from Sajjad (2007)

<NN> پہلے <O> <OR> سے <O> <SE> زیادہ <O> <ADV> غیرمحفوظ <O> <ADJ> ہو <VB> <O> گئ <O> <AA> - <O> <SM>

In Above Example following tag set is used for chunking task:

B: means Beginning of a Noun Phrase. It is the starting boundary of a noun phrase chunk.

I: means Inside of a Noun Phrase. This tag is used to elaborate a token as inside of the noun phrase.

O: means outside of Noun Phrase. This tag is used to elaborate the tokens which are not part of noun phrase chunks.

4 Literature Review

Abney (1991) introduced a new approach to parsing. He divided the parsing task into chunker and attacher. He mentioned that when we read, we read chunk by chunk. He introduced this natural phenomenon in machine world. The task of chunker was to convert sentences into non-overlapping phrases and the attacher was to combine these chunks in such a way that we would be able to get complete parses of the sentences. After Abney, much of work has been done on chunking which is mentioned in this section.

4.1 Methods of Chunking

Different techniques are implemented for chunking in different languages. Review of these techniques is given as:

1. Rule based Chunking
2. Corpus based Chunking
3. Hybrid Approach for Chunking

4.1.1 Rule based Chunking

Grover (2007) introduces rule based chunking using XML. The concern of this work is to develop a chunker which is reusable and easily configurable for any tag set. As CoNLL[2] data is used which is based on newspaper data and system is trained on this data he intended to use another data for this system. Results show that the machine learning systems out-perform such a rule based system but only when trained and tested on a domain specific data. Whenever the domain will be changed the machine learning systems may require retraining for the new domain. The XML based system outperforms when data from different sources is collected. He reported 89.1% Precision[3] and 88.57% Recall for Noun Group and 88.10% Precision and 91.86% Recall for Verb Group for English.

Ramshaw et al (1995) have proposed chunking as a tagging task. They used IOB tags for this purpose. They used B for beginning of chunk, I for mentioning the word token inside the chunk and O to demonstrate a word token as outside chunk. Their work initiated a new idea and a lot of later research on chunking. They used Brill's Transformation Based Learning Mechanism (TMBL) for text chunking. Previously this technique was used for part of speech tagging and disambiguation. The entire learning process is based on template rules. The first step is derivation

[2] Data provided for Conference on Computational Natural Language Learning (CoNLL 2000) shared task in year 2000

[3] Precision and Recall are illustrated in Section 6.1 (Results and Discussion).

of rules, second is scoring of rules, and third is selection of one rule with maximal positive effect. This process is iterative. They checked the candidate rules using this process to select all the rules which have maximum positive effect. Overall this approach achieves Recall and Precision of about 92% for baseNPs and 88% for partitioning Noun and Verb types.

4.1.2 Corpus based Chunking

Chen (1993) proposed a probabilistic chunker based on idea of Abney (1991) that when human being reads a sentence, the process of reading is on chunk by chunk basis. Experiment was conducted using three phases: training (extraction of bi-gram data from corpus), testing (tagging of raw data and output data) and evaluation (comparison of chunked data with corpus to report correct rate). Training of chunker is done by using Susanne Corpus, a modified version of Brown[4] Corpus containing 1 million words of English text. The evaluation is on the basis of outside and inside tests. Preliminary results showed that more than 98% was chunk correct rate and 94% sentence correct rate in outside test, and 99% chunk correct rate and 97% sentence correct rate in inside test.

Singh (2001) presented HMM based chunk tagger for Hindi. He divided shallow parsing into two main tasks: one was identification of chunk boundaries and the other was labeling of chunks with their syntactic boundaries. He used different schemes of tagging which were 2-tag scheme, 3-tag scheme and 4-tag scheme. He used different input tokens in their experiment which were words only, POS tags only, Word_POS tag (Word followed by POS tag) and POS_Word tag (POS tag followed by word). The annotated data set contains Hindi text of 200,000 words. Out of total annotated data, 20,000 words were used for testing, 20,000 words were kept for parameter tuning, and 150,000 words were used to train different HMM representations. The chunker was tested on 20,000 words of testing data and 92% precision with 100% recall achieved for chunk boundaries. He concluded that the machine learning technique is more suitable because of robustness.

Su (2001) observed the systems built using HMM based machine learning strategies outperform the rule based systems. He used HMM based chunk tagger in text chunking on the basis of ranks. This was observed that rank based HMM chunk taggers outperform even simple HMM based systems. The system was evaluated on MUC-6[5] and MUC-7[6] and the results of F-measure are 96.6 and 94.1 for both the evaluation systems for English named entities.

---

[4] The Brown University Standard Corpus of Present-Day American English (or just Brown Corpus) was compiled by Henry Kucera and W. Nelson Francis at Brown University, Providence, RI as a general corpus (text collection) in the field of corpus linguistics. (http://en.wikipedia.org/wiki/Brown_Corpus Reference cited on 23/07/09)

[5] MUC-6 is the sixth in a series of Message Understanding Conferences, held in November 1995.

Chen et al (1994) used probabilistic technique for chunking task. Previously Abney's motivated partial parsers by an intuition; when you read a sentence, read it chunk by chunk. They used Bi-gram model of HMM. Using this model both Recall and Precision were 95%.

Veenstra et al (1998) reported feasibility of different variants of memory based learning technique for fast chunking. The Dataset was based on 50,000 test and 200,000 train items. Benefits of such a technique are more visible in applications like Information Retrieval, Text Mining and Parsing. Memory based learning is based on examples. These examples are presented in the form of feature vectors with associated class labels. Examples (cases) are presented to classifier in incremental fashion and then added to memory as base cases for comparisons. A distance metric is a measurement to determine the distance between the class label of base cases and test cases. The algorithm which determines the distance is called IB1. It works in a manner if distance is 0 it means the trained class label is applicable on the test case and not applicable on the other hand in the case of 1 distance. An ambiguity is generated when there are more trained or stored cases which have zero distance with the test case. For this purpose a variant of this algorithm known as TiMBL is used which is an extension of IB1 algorithm. If a test case is associated with more than one class of training cases, TiMBL decides the class on the basis of frequency. Another algorithm IGTree is also evaluated in his paper. It is basically combination of IB1 and TiMBL; one for converting the base cases into the tree form, and the other for retrieval of classification information from these trees. Number of levels of a tree is equal to the number of nodes. In this tree features are stored in the form of nodes and in decreasing priority order i.e. the most important feature is at the root node and the next important at other level and so on. Non terminal nodes contain the information about default classes and leaf node contains unique class label. If first feature of test and base case is matched then it checks for next and so on. When the leaf node reaches the unique label of base case is assigned to test case. If the matching at any node is failed the default class label of previous node is assigned to that test case. The data set taken from parsed data of Ramshaw (1995) in the Penn Treebank corpus of Wall Street Journal text for training and testing. The collection was 47,377 for test cases and 203,751 for train cases. They reported that this method performs better as compared to transformation based learning of Ramshaw (1995). He reported the accuracy of 97.2% with 94.3% Recall and 89.0% Precision for NP Chunking.

Daelemans et al (1999) used memory based learning for shallow parsing in which POS tagging, Chunking, and identification of syntactic relations formulated as memory modules. Information extraction and summary generation use shallow parser as a main component. Shallow parsers were

---

[6] MUC-7 is the seventh in the series of Message Understanding Conference Evaluations, held in April 1998.

involved in discovering the main parts of sentences and their heads and syntactic relationships. The unique property of memory based learning approach was that they are lazy learners; all other statistical and machine learning methods are eager learners. Lazy learner techniques provide high accuracy as compared to the eager learner. Lazy learner technique keeps all data available even exceptions which sometimes are productive. Their paper provides empirical evidences for evaluation of memory based learning. The software used for memory based learning is TiMBL which is part of MBL software package. In IB1-IG, the distance between test item and memory item is defined on the basis of match and mismatch. Using IGTree a decision tree is obtained with features as tests). The empirical evaluation is divided into two experiments: one is evaluation of memory based NP and VP chunking, and the other is memory based subject/ object detection. The tag set used by NP and VP memory based chunker is {I_NP (Inside a baseNP), O (outside a baseNP or baseVP), B_NP (Begins a new baseNP in a sequence of baseNPs), I_VP (Inside a baseVP), B_VP (Begins a new baseVP in a sequence of baseVPs)}. The result of chunking experiment showed that accurate chunking is achievable for 94% F-measure value.

Shen (2003) gave a new idea for tagging the data; instead of using POS tagging a new form of tagging named as supertagging was used for detecting the Noun chunks. Supertags were used to expose more syntactic dependencies which are not available with simple POS tags. Such type of tagging is used only for Noun chunks and it was observed that by using this method of tagging about 1% absolute improvement in the F-Score is obtained (from 92.03% to 92.95%). Encoding of much more information than POS tagging was elaborated by Supertags that is why these were used as pre-parsing tool. Time Complexity of Supertags was linear as that of POS tags. On Data of Penn Treebank the Supertags achieved 92.41% accuracy. Supertags are trained on trigram models.

Pammi (2007) implemented decision trees for chunking and POS tagging for Indian Languages (Hindi, Bengali, and Telugu). He used an indirect way to build POS tagger without morphological analyzer using sub-words. Insufficient amount of training data, inherent POS ambiguities and unknown words are some problems faced during POS tagging. To resolve these problems subword like syllable, phonemes and onset vowel coda schemes are used. Rule based systems are not best for Indian languages because of excessive exceptions; his work used decision forests to solve exception problems in POS Tagging and chunking. Manual Annotated data was selected for experiments having 20000 words for each language. Five types of feature sets were selected for POS Tagging. Two-tag scheme was used for chunking in his paper; features used for chunking were also of two levels. At first two previous and then two next words were seen. He used a recursive partitioning algorithms which divides each parent node into left and right child nodes by posing YES-NO questions. The nodes at upper level have unique features but as the levels increase in the tree the nodes become more homogeneous. A stop parameter refers to the minimum number of samples required for training set data. It is observed that low stop value results into an over trained model. The feature in the tree which is predicted as an output of tree is called Predictee. A decision forest contains many decision trees. Each tree has own methodology to take decision. Each tree gives its observation say X to its corresponding forest. Then by voting method, the forest decides which output was favored by more votes. Then forest announces its decision to the corresponding feature list. The feature list receives decisions from multiple forests to use them as votes to decide the class of the word. For selection of dataset a random sample was taken which was 2/3 of the original data and the remaining is called out of bag data. Then it uses the bagging process in which the selection for each feature list was performed with replacement. He reported 69.92% accuracy for Hindi, 70.99% for Bengali and 74.74% for Telugu using decision forests.

4.1.3 Hybrid approach of Chunking

Schmid et al(2000) presents a noun chunker based on head-lexicalized probabilistic grammar. Such types of chunkers have many applications like Term Extraction and Index Terms for information retrieval. In their work, probabilistic noun parser was used to get noun chunks. The language used was German. There are some rules used, which provide robustness to process arbitrary input. They conducted two experiments with different strategies. In both experiments, 1 million training words are provided from corpus of relative clauses, 1 million of verb final clauses and 2 million words of consecutive text. Data was taken from Huge German Corpus (HGC). The respective precision and recall values were 93.06% and 92.19%. The results explain that untrained version of grammar is improved using rules frequencies of trained grammar. The unlexicalised training itself is sufficient to extract nouns instead of combination of lexicalized and unlexicalised version. Identification of syntactic category through Noun chunker results in 83% Precision and 84% Recall.

Park et al (2003) in their paper described a new approach of chunking using Korean language. The hybrid approach is used. Initially, the rule based chunking is done. Memory based learning technique is used for the correction of errors, which were exceptions to rules. Machine based learning methods are considered best for English language but for the languages which are free word order or partially free word order such techniques are not successful. English has different grammatical relations like positions and other determiners which tell about the boundary of chunks, but in free word order languages such a facility is not available. So the free word order languages are difficult to handle during chunking using machine learning. Post-positions are helpful in free order languages while chunking. Korean and Japanese are examples of partially free word order languages. Their work describes a new methodology which is basically hybrid of both rule based and memory based learning techniques. At First rule based approach is used to detect the most of the chunks and then evaluated against the hand crafted rules and then identified the misinterpreted rules and managed into a file called error file. This file is then given to memory based learning system along with correct rules to learn on exceptional rules as information to correct errors introduced by the rule based systems. The main role of memory based learning method in this system is to determine the context for exceptions of rules. The Four basic phrases of Korean language are detected, namely, Noun Phrases (NP), Verb Phrases (VP), Adverb Phrases (ADVP) and Independent Phrases (IP). Each phrase can have two types of chunk tags: B-XP and I-XP. The chunk tag O is used to identify phrases which are not part of any chunk. Using only rules gives 97.99% accuracy and 91.87 of F-Score. Here F-Score is low rather it is important than accuracy. The hybrid approach shows 94.21 F-Score on the average, which is 2.34 score improvement over rules-only technique, 1.67 over support vector machine and 2.83 over memory based learning. This result was even better than reported for English language.

4.2 Tools for Chunking

Voutelainen (1993) explains a tool for detecting noun phrases, named NPTool. It is a modular system for morpho-syntactic analysis. Tool consist of two NP parsers one is NP-friendly and the other is NP-hostile parser. NP Hostile parser is hostile to noun phrase readings while NP Friendly parser is hostile to non noun phrase readings.Match of output of both NP Friendly Parser and NP Hostile Parser conducted and all those noun phrases considered as candidate which are present in output of both the parsers and labeled OK. By using this tool extraction of not only noun phrases can be done but with some improvement extraction of every type of phrases can be done. Analysis of 20,000 words has been done to evaluate this tool, a Recall of 98.5% to 100%, with a Precision of 95% to 98% were achieved.

4.3 SNoW based Chunking tag set comparison

Munoz et al (1999) compares two ways of shallow based pattern learning; one is called Open/Close and the other is called Inside/Out predictors. The learning architecture in this paper is known as SNoW (Sparse Network of Winnows) which is a sparse network of linear functions over predefined or incrementally learned features and is domain dependent. Two different instantiation of this paradigm are studied on two different shallow parsing tasks that are baseNP (baseNP are non recursive NPs) and Subject Verb phrases (SV phrases-phrases starts with subject of the sentence and ends with verb). First instantiation of paradigm decides about the word using predictors whether it is interior of a phrase or not, and then group all the interiors in the form of phrases also known as IOB tagging {I,O,B}.Inside/Out method consists of two predictors. The first predictor takes POS tags (represents the local context of each word) as input after feature extraction. This predictor outputs the IOB boundaries along with POS tags and is presented to the second predictor which takes input in the form of IOB tags which describes the local context of word using neighboring words. The second predictor then outputs its prediction in the form of phrases. In Open/Close Predictor boundaries are determined on the basis of Open bracket and Close bracket, open bracket demonstrates start of a phrase (marked before first word of phrase) and close bracket (marked after the last word of phrases) demonstrates the end of the phrase. Two predictors SNoW Open predictor and SNoW Close predict are used in a competing manner. The target features of both the predictors are compared (Yes bracket Vs No bracket) to get confidence level. It was evaluated that the Open/Close method has better performance than that of Inside/Out method.

5 Current Work

5.1 Motivation

Basili et al (1999) realized the need of chunking in terms of high processing speed and low costs in the design and maintenance of grammars. According to him the chunking improves throughput in comparison of full parsers. Grover (2007) considers chunking useful for Named Entity Recognition.

Thus chunking is a technique to reduce cost of full parsing, it also trims down the search space. Chen et al (1994) considers chunking as an important concept used in the linguistics because complete parsing is not always required. Complete parsing is difficult to achieve because neither syntax analysis nor semantic analysis solely can provide it.

The motivation for selection of only noun phrase chunking was empirical. In other languages, most of the work is present for noun phrases. In this work, the whole corpus is analyzed, and it is observed that around 60% words of corpus are noun phrases or part of noun phrases and the remaining phrases collectively constitute 40% of corpus. So, it is believed that in contrast to other phrases chunking task for noun phrases itself counts more in benefits of chunking.

It is also beneficial where full tree parsers are partially required or not required at all like Named Entity Recognition (NER), Information Retrieval (IR), Question Answer Applications (QA), Machine Translation (MT), Speech Synthesis and Recognition, and Index Term Generation (ITG).

5.2 Problem Statement

Das (2004) illustrated:

Indo-Aryan languages being relatively free word ordered are difficult to tackle using a generative grammar approach. Moreover, unavailability of chunked corpora precludes the use of available statistical approaches.

Then chunking is a task to build a corpus with proper identification of chunks of different types. Chunking task was made easy by Ramshaw et al (1995). They converted the chunking problem into a tagging problem by introducing chunk tags; therefore the problem can be defined as under:

"Given a sentence of Urdu language along with POS tags of tokens, generate Noun Phrase Chunk tags for the sentence."

The solution for this problem is development of a process for Urdu NP chunking, and investigation of different methods for best candidate with respect to Urdu language.

5.3 Scope

The Scope of this work is limited to investigation of best methodology for Urdu NP chunking in terms of accuracy. Different experiments were conducted based on a combination of statistical and rule based chunking. This hybrid approach finds the best candidate method on the basis of accuracy. Marker based chunking is used, based on "Marker Hypothesis" of Green (1979) for marking the noun phrases. The freely available and/ or open source tagging tools are used to investigate hypotheses of this work.

5.4 Methodology

This section introduces methodology which provides basis for overall model of the system. Statistical NP chunking essentials are elaborated in subsequent subsections.

5.4.1 Computational Model

In this work hybrid approach based on Statistical chunking and then Rules based chunking is used. First POS annotated corpus is prepared for statistical model and then after error analysis hand crafted rules are extracted to implement for better accuracy. POS tags are Input of the system and IOB tags are the output. T is a sequence of n tags from t_1 to t_n and C is a sequence of c_1 to c_n chunk tags. So, the problem is to get best chunk tag sequence (C) provided that POS tag sequence (T) is already known. The probabilistic model for this problem is as under:

$$\hat{C} = \arg\max_{C} P(C \mid T)$$

Using Bayes' rule it can be written as:

$$\hat{C} = \arg\max_{C} \frac{P(T \mid C)\, P(C)}{P(T)}$$

Since we are maximizing C so the denominator will remain constant so

$$\arg\max_{C} P(T \mid C)\, P(C)$$

Using Markov assumption, the whole Chunk tag sequence is estimated using Trigrams, and likelihood is also simplified such that a POS tag t_i depends only on corresponding Chunk tag c_i. Hence,

$$\text{Emission Probabilities} = P(t_i \mid c_i) \qquad \text{(I)}$$

$$\text{State Transition Probabilities} = P(c_i \mid c_{i-2}, c_{i-1}) \qquad \text{(II)}$$

By Combining (I) and (II)

$$\arg\max_{C} \prod_{i=1}^{i=n} P(t_i \mid c_i) \quad P(c_i \mid c_{i-2} c_{i-1})$$

For obtaining probability of $P(t_i \mid c_i)$ following equation is used:

$$P(t_i \mid c_i) = \frac{\text{Count of } (t_i, c_i)}{\text{Count of } c_i}$$

For obtaining Trigram probability following equation is used:

$$P(c_i \mid c_{i-1}\ c_{i-2}) = \frac{\text{Count of } (c_{i-2}, c_{i-1}, c_i)}{\text{Count of } c_{i-2}\ c_{i-1}}$$

The Optimal sequence of chunk tags is found using Viterbi algorithm which uses parameters of HMM for fast and better execution.

5.4.2 Architecture

This sub-section elaborates overall architecture of the system. POS annotated corpus of 101428 words is acquired and the data of 91428 words is prepared for training and the remaining 10000 words are kept for testing the model. The whole corpus is then manually chunk tagged. The Training data is then presented to TnT Tagger which generates Uni-gram, Bi-gram and Tri-gram counts and stores these counts to be used at the time testing. Testing POS only data of 10000 tokens properly formatted as required by the tagger is presented to the "tnt.exe" utility of the tagger to get appropriate chunk tags. Tagger outputs the data with appropriate chunk tags using HMM model. Data generated by the tagger is then compared with manually chunk tagged data. The Accuracies are recorded and then the output of the tagger is analyzed and hand crafted rules (post processing) are extracted after this analysis. The sequence of firing the rules is developed carefully to avoid bleeding and creeping. After getting suitable sequence of rules, these rules are applied on the output of tagger one by one and accuracy of each rule is maintained for measuring the effectiveness of rules. Figure 1 describes the architecture of system.

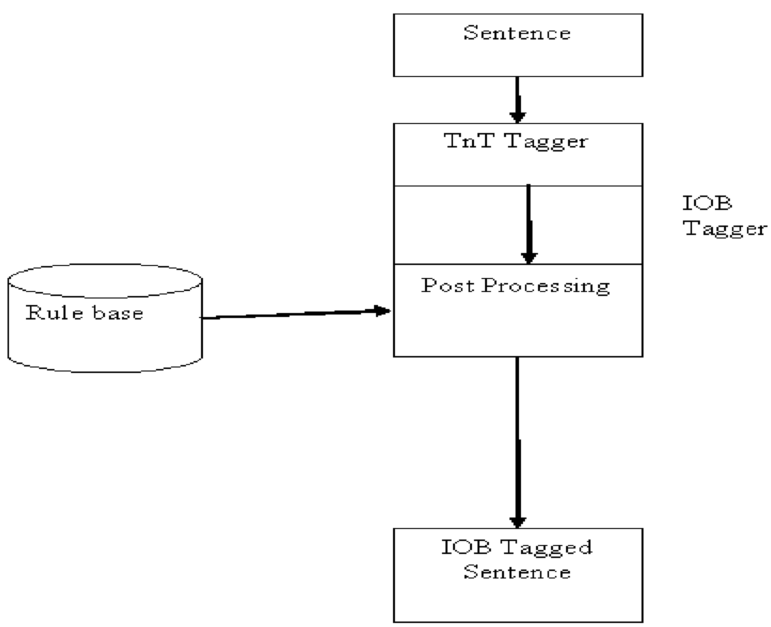

Figure 1: Architecture of the System

5.4.3 Tagger

The tagger used for this work is TnT Tagger developed by Brants (2000). This tagger is a HMM based statistical POS Tagger. It is simple and very efficient POS tagger. It uses linear interpolation model of smoothing.

TnT tagger has different utilities like tnt-para.exe, tnt.exe and tnt-diff.exe. The utility tnt-para.exe generates the n-gram counts of training data. The tool tnt.exe is the main utility that uses Viterbi algorithm and annotates the input and generates an output file. Tnt-diff.exe is used to compare automated output with the manually annotated test corpus, and provides accuracy.

All the experiments are executed using its default option of second order HMMs (Trigram Model).

5.4.4 Preparation of Data

The POS annotated Corpus used is taken from CRULP (Center for Research in Urdu Language Processing). The chunk tagger uses POS tags for marking the NP chunk boundaries. So Chunk tagger is directly dependent on correctness of POS tags. For the purpose of getting maximum benefit out of chunk tagger, errors found in POS tags during IOB tagging are removed from the corpus. The Issues, ambiguities, and their solutions are discussed next.

5.4.4.1 Revision of Data

The study of original data shows that there is a need of revision with respect to the requirements of current work. Following are the observations and their accustomed resolution:

Some Ambiguities are found in POS tags. Some words are marked as personal demonstrators <PD> but their contextual information told that those are personal pronouns. Some examples also exist for other pronouns. In some readings demonstratives are marked as pronouns but in the context those are found as demonstrative. Following are Examples:

1- حالانکہ <ADV> **اس <PD>** سے <SE> بھی <I> چمڑی <NN> ادھڑ <NN> جاتی <VB> ہے <TA>

2- **یہ <PP> تمام <Q> باتیں <NN>** ابھی <AP> تک <P> واضح <ADJ> نہیں <NEG> - <SM>

In sentence 1 of above example “اس” is replacement of a noun so it must be personal pronoun (PP) instead of personal demonstrative (PD). In sentence 2 of above example یہ demonstrates تمام باتیں and behaves like a demonstrator (PD) but marked as personal pronoun.

Some words are marked Personal Pronouns instead of Kaf Pronoun, though it’s not a big problem considering POS only but from training point of view it will decrease count of kaf Pronouns and increase personal pronouns count, which may affect the learning pattern. For example:

اٹلی <PN> یا <SC> جاپان <PN> **کوئی <PP>** بھی <I> ملک <NN>

In above example “کوئی” is marked as personal pronoun (PP) but actually it is kaf pronoun (KP) according to tag set of Sajjad (2007).

Some words are marked subordinate conjunction at some places and marked as nouns and adverb at others though used in same context. For example:

1- زور <NN> دے <VB> کر <KER> دبائیں <VB> **تاکہ <NN>** پانی <NN> باہر <NN> آ <VB> جائے <AA> - <SM>

2- بنایا <VB> جائے <AA> **تاکہ <NN>** گندم <PN> کے <P> ربیع <PN>

3- استعمال <NN> کریں <VB> **تاکہ <SC>** پانی <NN> کم <ADJ> سے <SE>

The occurrences of Date that behave as noun are tailored to fit the need of NP chunking (*See Appendix A*). Following is an example of date tag:

اپریل <PN> 1972 <DATE> کو <P> ملک <NN> میں <P> عبوری <ADJ> آئین <NN> نافذ <NN>

Some instances were present in the annotated corpus in which the same word was tagged as proper noun (PN) and at another place same word was marked as adjective (ADJ), both having same type of context. Following is an example such inconsistencies:

1- پاکستان<PN>میں<P>اقتدار<ADJ>اعلی<ADJ>**اللہ<PN>تعالی<ADJ>**کی<PK>ذات<NN>ہے<VB>-

2- حکومت<NN>**اللہ<PN>تعالی<ADJ>**اور<CC>عوام<NN>کے<PK>سامنے<NN>جوابدہ<NN>ہے<VB>-

3- علامہ<PN>اقبال<PN>**اللہ<PN>تعالی<PN>**سے<SE>دعا<NN>کرتے<VB>تھے<TA>کہ<SC>وہ
<PP>ملت<NN>اسلامیہ<NN>کے<PK>نوجوانوں<NN>کو<P>سحر<NN>سے<SE>مالامال<ADJ>کر
<VB>دے<AA> اور<CC>انہیں<PP>بصیرت<NN>سے<SE>نوازے<VB>-<SM>

4- **اللہ <PN> تعالی <PN>** نے <P> ارشاد <NN> فرمایا <VB> زانی <ADJ> مرد <NN> اور <CC> زانیہ
<ADJ>عورت <NN> کو <P> 100 <CA> کوڑے <NN> مارو <VB> - <SM>

After manual tagging the training data is prepared to present to the tagger, because the tagger accepts data in certain format so it is necessary to convert data into that format so that it would be able to use in the process.

5.4.4.2 Identification of Boundaries for Noun Phrases

While manual chunk tagging the issues of consistency were a real challenge. For Example:

1- **اس <PP> <B> طرح <NN> <B>** اذیتپسند <ADJ> <B> عناصر <NN><I> محکمے <NN> <B> میں <P>
<O>شامل <NN> <B> ہو <VB><O> جاتے <AA> <O> ہیں <TA> <O>

2- **اس <PP> <B> طرح <NN> <I>** اذیتپسند <ADJ> <B> عناصر <NN><I> محکمے <NN> <B> میں <P>

<O>شامل <NN> <B> ہو <VB><O> جاتے <AA> <O> ہیں <TA>   <O>

Above mentioned two readings of noun phrase chunk have different boundaries and both are correct. First one is correct using linguists point of view having two noun phrases but the second one seems correct having only one noun phrase because of daily life usage (Abney's approach[7]). For the sake of consistency linguistic approach was considered for every such case. For the purpose of consistency, a document was developed having all the decisions of ambiguous readings (See Appendix A).

Following is an example of such ambiguities:

1- ہمارے <G> معاشرے <NN> میں <P> آئین <NN> کی <P>**دفعہ <NN> 302 <CA>** کا <P> سب <Q> سے <SE>زیادہ <ADV> غلط <ADJ> استعمال <NN> ہوتا <VB> ہے <TA>

2- اس <PD> **لئے <NN> 80 <CA>** فیصد <ADV> بچوں <NN> کو <P>تربیت <NN> اور <CC> دوڑ <NN> کے <P> دوران <NN> شدید <ADJ>چوٹیں <NN> بھی <I> آتی <VB> ہیں <TA> **-** <SM>

In above example cardinal (CA) has two different versions. In version 1, cardinal is part of the noun phrase having preceding noun and in version 2, cardinal is not part of the noun phrase of preceding noun. The scenario of cardinal for both versions is same, but behavior in both versions is different. If during training, system learns readings of version 1, then readings of version 2 will also be handled with same behavior which will be treated as error of the system. We need to take decision to resolve such ambiguities. Terms of reference document is maintained having decisions to resolve many such ambiguities *(See Appendix A)*.

Marker base chunking approach is used in this work based on "Marker Hypothesis" of Green (1979). Marking the chunk boundaries using syntactic markers is very useful for Noun phrases marking. Such markers are of different types included Genitive, Dative etc. For detail and examples of markers see sub-section 2.3.2.

5.5 Experimentation

A series of experiments are conducted using different implementation techniques to get maximum accuracy for chunking. These Experiments are divided into two phases. In first phase statistical tagger is used to get IOB chunked tags output and the accuracy is obtained using difference of manual IOB tags and automated IOB tags and in second phase using analysis of the difference, the hand crafted rules are devised for each experiment and then implementation of all these rules one by one for individual accuracies of rules is done. The outline of experiments is as follows:

---

[7] Abney (1991) coined the term chunks as "when we read, we read chunk by chunk"

1. Base Experiment using Basic Methodology
    a. Right to Left Training and Testing (Natural direction of Urdu)
    b. Left to Right Training and Testing
2. Extended Experiment using Transformation of All POS
3. Extended Experiment using Transformation of only Nouns

5.5.1 First Phase of Experiments (Statistical Method Implementation)

First phase of experimentation is basically implementation of statistical computational model. In following subsections statistical methodology of all the experiments is discussed.

5.5.1.1 Experiment 1: Base Experiment Using Basic Methodology

In this experiment computational model is trained on POS tags. Given the sequence of POS tags the system outputs the sequence of IOB tags. This experiment is divided into two sub experiments; one is implementation of computational model on right to left direction of corpus data which means that sentence markers are processed at the end of the sentence, second is left to right which means that sentence markers are processed first and so on *(See Appendix C)*. After execution of model from right to left and left to right, a comparison is made.

5.5.1.2 Experiment 2: Extended Experiment Using Transformation of All POS

This experiment is also an extension of base experiment using POS in combination with IOB. In it, IOB tagset is changed to POS_IOB tag set. This method is used by Molina et al (2002) for English and reported best accuracy. In this experiment, method of Molina et al (2002) is tailored. By combining POS with IOB tags in training set and in testing given POS tags POS_IOB tags are obtained from the tagger.

The transformation is executed by concatenating POS tag sequence "T" and the chunk tag sequence "C" to form the sequence such that each term is "$t_i\_c_i$".

Then POS sequence and $t_i\_c_i$ chunk sequence are presented to tagger for training, and given POS sequence of test corpus to tagger and in return tagger outputs $t_i\_c_i$ chunk sequence for POS sequence of test corpus (*See Appendix C*). Then IOB tags are extracted from the output of tagger and compared with same manual IOB tagged testing data, accuracy is recorded.

5.5.1.3 Experiment 3: Extended Experiment Using Transformation of Nouns Only

This experiment is conducted after an observation that some readings of nouns are so ambiguous that even manual analysis cannot detect proper boundaries. For example:

ان <PP> میں <P> سابق <ADJ> ممبر <NN> بورڈ <PN> آف <PN> ریونیو <PN> میاں <PRT> فیض
<PN> کریم <PN> قریشی <PN> سابق <ADJ> ایم <PN> ایس <PN> ایز <PN> ملک <PN> غلام <PN>
محمد <PN> مرتضی <PN> کھر <PN> میاں <PRT> غلام <PN> عباس <PN> قریشی <PN> بریگیڈئیر
<PRT> ضمیر <PN> احمد <PN> خان <PN> سردار <PN> عبدالقیوم <PN> خان <PN> جتوئی <PN> ملک
<PN> سلطان <PN> محمد <PN> ہنجرا <PN> کیپٹن <PRT> خالد <PN> احمد <PN> گورمانی <PN> رانا
<PN> محبوب <PN> اختر <PN> کے <P> نام <NN> قابل <ADJ> ذکر <NN> ہیں <VB>

It is considered that by combining POS with IOB tags in training set and in testing given POS tags of NN and PN all other tags are kept intact. NN_IOB or PN_IOB tags are obtained from the tagger along with IOB of other POS tags (*See Appendix C*). Then IOB tags are extracted from the output of tagger and compared with same manual IOB tagged testing data, accuracy is recorded.

5.5.2 Second Phase of Experiments (Implementation of Rules)

The last phase of the experiment is extraction of rules after analysis of difference between Manual IOB tagged data and IOB tagged out put of Tagger. Then these rules are applied one by one and the accuracy is recorded each time to check effectiveness of every rule.

When dry run of some of examples using computational model was executed, it was observed that system cannot identify some pattern due to ambiguities. The need of rules is evolved by observing the errors. It is an assumption that wrong pattern learning will diminish the accuracy. To obtain high accuracy hybrid approach based on statistical and rule based is used.

Following are some readings found during dry run of the system.

Table 3: Dry Run of Statistical Model

Word Tokens	POS Tags	Manual Tags	Dry Run of Computational Model
اس	PP	B	B
پر	P	O	O
کم	**ADJ**	**B**	**B**

Word Tokens	POS Tags	Manual Tags	Dry Run of Computational Model
از	**CC**	**I**	**O**
کم	**ADJ**	**I**	**B**
پبلک	**NN**	**I**	**I**
مقامات	**NN**	**I**	**I**
پر	P	O	O
قدغن	NN	B	B
لگانا	VB	O	O
وقت	NN	B	B
کی	P	O	O
بنیادی	ADJ	B	B
ضرورت	NN	I	I
ہے	VB	O	O
-	SM	O	O

In above table a noun phrase is marked bold. In this phrase a coordinate conjunction is present between two adjectives and second adjective is followed by a noun. Such a construction is a noun phrase as mentioned by manual chunk tags, but system dry run could not find this pattern. Rule 1 is evolved after observing this pattern *(See Appendix B)*. Following reading is also an example of same phenomenon:

Table 4: Rule 1 Example in Dry Run

Word Tokens	POS Tags	Manual Tags	Dry Run of Computational Model
جن	REP	B	B
میں	P	O	O
بیمار	**ADJ**	**B**	**B**
اور	**CC**	**I**	**O**
لاغر	**ADJ**	**I**	**B**
جانور	**NN**	**I**	**I**

Word Tokens	POS Tags	Manual Tags	Dry Run of Computational Model
بھی	I	O	O
شامل	NN	B	B
ہوتے	VB	O	O
ہیں	TA	O	O
-	SM	O	O

Following example illustrates the need of Rule 2 *(See Appendix B)*

Table 5: Example of Rule 2 in Dry Run

Word Tokens	POS Tags	Manual Tags	Dry Run of Computational Model
یقینا	ADV	O	O
تجاوزات	NN	B	B
کا	P	O	O
خاتمہ	NN	B	B
ممکن	**ADJ**	**O**	**B**
ہو	VB	O	O
سکتا	AA	O	O
ہے	TA	O	O
-	SM	O	O

Adjective in above table is marked outside as per *Appendix A*, but it is marked outside using dry run of computational model of the system. Such readings can be corrected using rules.

6 Results and Discussion

6.1 Results

For the purpose of testing 10,000 words were used for each experiment. Initially statistical model was applied to all experiments then a generic rule set (*see Appendix B*) of 23 rules was devised for these experiments after analysis of automated output of all experiments; these rules were then applied to stochastic output of all experiments. Before presenting the result of experiments, it is considered necessary to introduce the reader with evaluation methods of the results. Following are Evaluation methods used in this work.

6.1.1 Overall Accuracy of Experiments

Over all accuracy of each experiment is calculated using matched tags of manual annotated testing data and automated annotated testing data. It is the ratio between correct tags and total tags. The formula for overall accuracy of Experiment is given below:

$$\text{Accuracy}\,(\%) = \frac{\text{Correct automated Tags}}{\text{Total Tags Generated by Tagger}} * 100$$

6.1.2 Precision

The precision is accuracy of target set which is different for each of B, I and O tags used in this work and is calculated by using following equation:

$$\text{Precision}\,(\%) = \frac{\text{Correct automated Target Tags}}{\text{Total Target Tags Generated by Tagger}} * 100$$

Lager (1995) elaborated that less than 100% precision means that the system found something which is not part of the correct result.

6.1.3 Recall

The Recall is overall coverage of the tagger. Recall is also different for each target tag.
Following is the formula to get Recall for a particular target tag.

$$\text{Recall}\,(\%) = \frac{\text{Correct automated Target Tags}}{\text{Total Tags Generated by Tagger}} * 100$$

Lager (1995) described that less than 100% recall means that the system missed some desired things which were part of the correct result set.
The results are obtained after executing all experiments mentioned in the methodology and are discussed below one by one.

6.1.4 Experiment 1: Base Experiment Using Basic Methodology

Base Experiment is conducted using different direction training and testing. Right to left means sentence marker is at the end of the sentence and left to right means sentence marker is at the start of sentence as mentioned in methodology. Following are results of experiments of both directions one by one along with comparison.

6.1.4.1 Right to Left Training and Testing (Natural Direction of Urdu)

This experiment is conducted using training and testing data in right to left direction which means sentence marker is at the end of the sentence. This direction is natural direction of Urdu language. First stochastic model is executed on testing data to obtain accuracy. Then Precision and Recall for I, O and B tags were calculated separately. The overall accuracy of experiment was 90.93% with 90.10% precision and 83.65% recall for B tag of chunking, 72.10% precision and 90.39% recall for I, and 99.23% precision and 96.22% recall for O.

By applying rules in a sequence on output of statistical tagger we obtained overall accuracy of 93.87%. The Precision and Recall for B tag were 90.81% and 85.44% for I tag those were 74.96 and 94.53, and for O Precision and Recall were 99.62 and 99.60. For illustration of rule's participation in accuracy of this experiment (*see Appendix D)*.

The comparison of accuracy, precision and recall before and after rule execution is given in the Table 6.

Table 6: Overall Accuracy, Precision and Recall Before and After Implementation of Rules (Right Left Direction Experiment)

Type of Measure	Statistical Method	Application of Rules	Improvement
Accuracy (%)	90.93	93.87	2.94
Precision for B Tag(%)	90.10	96.81	6.71
Recall for B Tag (%)	83.65	85.44	1.79
Precision for I Tag(%)	72.10	74.96	2.86
Recall for I Tag (%)	90.39	94.53	4.14
Precision for O Tag(%)	99.23	99.62	0.39
Recall for O Tag (%)	96.22	99.60	3.38

6.1.4.2 Left to Right Training and Testing

This experiment is conducted using training and testing data in left to right direction which means sentence marker is at the header of the sentence. First stochastic model is executed on testing data to obtain accuracy. Then Precision and Recall for I, O and B tags were calculated separately. The overall accuracy of experiment was 90.86% with 90.23% precision and 83.57% recall for B tag of chunking, 71.84% precision and 90.13% recall for I, and 99.08% precision and 96.22% recall for O.

By applying rules in a sequence on output of statistical tagger we obtained overall accuracy of 93.79%. The Precision and Recall for B tag were 96.59% and 85.41% for I tag those were 74.91 and 94.27, and for O Precision and Recall were 99.60 and 99.53. For illustration of rule's participation in accuracy of this experiment (*see Appendix D)*.

The comparison of accuracy, Precision and Recall before and after rules execution is given in the Table 7.

Table 7: Overall Accuracy, Precision and Recall Before and After Implementation of Rules (Left to Right Direction Experiment)

Type of Measure	**Statistical Method**	**Application of Rules**	**Improvement**
Accuracy (%)	90.86	93.79	2.93
Precision for B Tag (%)	90.23	96.59	6.36
Recall for B Tag (%)	83.57	85.41	1.84
Precision for I Tag (%)	71.84	74.91	3.07
Recall for I Tag (%)	90.13	94.27	4.14
Precision for O Tag (%)	99.08	99.60	0.52
Recall for O Tag (%)	96.22	99.53	3.31

Comparison of both left to right and right to left overall accuracies, Precisions and Recalls elaborate that there is no significant difference in both approaches.

Following table shows error analysis of both approaches:

Table 8: Error Analysis of Left to Right and Right to Left Approach

Errors LTR	Errors RTL	Input	Firing of Rules	Output	Errors RTL	Difference RTL	Errors LTR	Difference LTR
914	907	Statistical Input (I1)	Rule 1A	O1	899	8	905	9
905	899	O1	Rule 1B	O2	891	8	898	7
898	891	O2	Rule 2	O3	859	32	866	32
866	859	O3	Rule 3	O4	858	1	865	1
865	858	O4	Rule 4	O5	773	85	779	86
779	773	O5	Rule 5	O6	773	0	779	0
779	773	O6	Rule 6	O7	755	18	765	14
765	755	O7	Rule 7A	O8	755	0	765	0
765	755	O8	Rule 7B	O9	755	0	765	0
765	755	O9	Rule 8	O10	752	3	762	3
762	752	O10	Rule 9	O11	734	18	745	17
745	734	O11	Rule 10	O12	730	4	740	5
740	730	O12	Rule 11	O13	725	5	735	5
735	725	O13	Rule 12	O14	723	2	733	2
733	723	O14	Rule 13	O15	722	1	732	1
732	722	O15	Rule 14	O16	714	8	724	8
724	714	O16	Rule 15A	O17	714	0	724	0
724	714	O17	Rule 15B	O18	711	3	721	3
721	711	O18	Rule 15C	O19	711	0	721	0
721	711	O19	Rule 16A	O20	667	44	677	44
677	667	O20	Rule 16B	O21	667	0	677	0
677	667	O21	Rule 17A	O22	646	21	656	21
656	646	O22	Rule 17B	O23	643	3	650	6
650	643	O23	Rule 18A	O24	640	3	647	3
647	640	O24	Rule 18B	O25	640	0	647	0
647	640	O25	R19A	O26	640	0	647	0
647	640	O26	Rule 19B	O27	630	10	637	10
637	630	O27	Rule 20	O28	626	4	633	4

Errors LTR	Errors RTL	Input	Firing of Rules	Output	Errors RTL	Difference RTL	Errors LTR	Difference LTR
633	626	O28	Rule 21	O29	619	7	625	8
625	619	O29	Rule 22	O30	616	3	621	4
621	619	O30	Rule 23	O31	613	6	621	0

It is observed that almost all errors were same in both approaches, except particles were marked inside phrases six times in right to left approach but none is marked in left to right approach.

6.1.5 Experiment 2: Extended Experiment using Transformation of All POS

This experiment is conducted using a new set consisting of POS_IOB as output set and POS were input of the system. First stochastic model is executed on testing data to obtain maximum accuracy out of it. Then Precision and Recall for I, O and B tags were calculated separately. The overall accuracy of experiment was 97.28% with 96.05% precision and 96.35% recall for B tag of chunking, 91.88% precision and 92.23% recall for I, and 99.92% precision and 99.58% recall for O.

By applying rules in a sequence on output of statistical tagger we obtained overall accuracy of 97.52%. The Precision and Recall for B tag were 96.50% and 96.52%, for I tag those were 92.33 and 92.68, and for O Precision and Recall were 99.90 and 99.76. For illustration of rule's participation in accuracy of this experiment (*see Appendix D)*.

The comparison of accuracy, Precision and Recall before and after rules execution is given in the Table 9.

Table 9: Overall Accuracy, Precision and Recall Before and After Implementation of Rules (Extended Experiment with transformation of All POS)

Type of Measure	Statistical Method	Application of Rules	Improvement
Accuracy (%)	97.28	97.52	0.24
Precision for B Tag (%)	96.05	96.50	0.45
Recall for B Tag (%)	96.35	96.52	0.17
Precision for I Tag (%)	91.88	92.33	0.45

Type of Measure	Statistical Method	Application of Rules	Improvement
Recall for I Tag (%)	92.23	92.68	0.45
Precision for O Tag (%)	99.92	99.90	-0.02
Recall for O Tag (%)	99.58	99.76	0.18

6.1.6 Experiment 3: Extended Experiment using Transformation of Nouns Only

This experiment is conducted using training and testing data of base experiment with transformation of only nouns is done. First stochastic model is executed on testing data to obtain accuracy. Then Precision and Recall for I, O and B tags were calculated separately. The overall accuracy of experiment was 92.30% with 90.40% precision and 94.46% recall for B tag of chunking, 86.23% precision and 85.68% recall for I, and 99.90% precision and 96.95% recall for O.

By applying rules in a sequence on output of statistical tagger we obtained overall accuracy of 96.31%. The Precision and Recall for B tag were 93.64% and 96.50%, for I tag those were 91.09 and 86.57, and for O Precision and Recall were 99.84 and 99.27. For illustration of rule's participation in accuracy of this experiment (*see Appendix D)*.

The comparison of accuracy, Precision and Recall before and after rules execution is given in the Table 10.

Table 10: Overall Accuracy, Precision and Recall Before and After Implementation of Rules (Extended Experiment with transformation of only Nouns)

Type of Measure	Statistical Method	Application of Rules	Improvement
Accuracy (%)	92.30	96.31	4.01
Precision for B Tag(%)	90.40	93.64	3.24
Recall for B Tag (%)	94.46	96.50	2.04
Precision for I Tag(%)	86.23	91.09	4.86

Type of Measure	Statistical Method	Application of Rules	Improvement
Recall for I Tag (%)	85.68	86.57	0.89
Precision for O Tag(%)	99.90	99.84	-0.06
Recall for O Tag (%)	96.95	99.27	2.32

The comparison of overall accuracy of all the experiments with statistical methodology and rule based implementation is described in Table 11:

Table 11: Overall accuracy comparison of all experiments with statistical and rule based implementation

S#	Method	Overall Accuracy of Statistical Method	Rules implementation	Improvement
1a	Experiment # 1A: Base Experiment (Right to Left Direction)	90.93	93.87	2.94
1b	Experiment # 1B: Base Experiment (Left to Right Direction)	90.86	93.79	2.93
2	Experiment # 2: Extended Experiment with transformation of All POS	97.28	97.52	0.24
3	Experiment # 3: Extended Experiment with transformation of only Nouns	92.30	96.31	4.01

6.2 Discussion

This study was planned to perform chunking task on Urdu language and established a system for chunk tagging with maximum accuracy. Another motivation of this work was to compare different experiments using hybrid approach for comparison of different methodologies in terms of accuracy for Urdu language. The intention to conduct experiments using different schemes was to mark the factors which were important for producing high accuracy. The investigation of factors detrimental to accuracy was also under consideration. Some observations are made after analysis of results.

An important observation is about Experiment 1: base experiment in which the tagger was given same training and test corpus once in right to left direction and once in left to right direction to find any difference between both directions implementation. Almost same accuracies were obtained in each direction even after rule implementation minor difference found which is ignorable. The fact was also noted that precision and recall for both directions were also almost same (See Table 6 and Table 7). It was decided that if non-overlapping difference between both approaches will be significant then operations of union, intersection, AND, and OR will be used which one will be suitable to achieve high accuracy. It was observed that no significant non-overlapping difference between both approaches exist, so only right to left direction was followed in later experiments.

The base experiment was supported on POS tags as input set of the system and IOB tag set as output of the system. It was observed that the system could not learn many patterns correctly. Some examples are mentioned in 5.5.2. Base experiment was analyzed and observed that ambiguities evolved due to small output tag set. Overall accuracy obtained in this experiment was 90.93. Precision of B tags in this experiment was 90.10 which were improved by 6.71 using rules. It means most errors found were basically of marking start boundary of noun phrases. Precision for I and O for both directions shows that there is not significant improvement in contrast to precision of B tags after implementation of rules. The major participation of rules in this experiment was correction of tag B marked wrongly I or O by the statistical system.

Another sequence of input and output tag set was executed using same statistical model in which POS tags were merged with output tag set called extended experiment (experiment 2). It was observed that Experiment 2: Extended Experiment using transformation of All POS outperformed all other experiments with the accuracy of 97.28% which improves only 0.24 after implementation of rules on it and reached to 97.52. In analysis of this methodology of the experiment, it was found that using this method, the number of chunk tags increased to more than 100 tags because in this method we combine both the POS and IOB in training and then only POS tags were presented to the tagger for testing data. By combining 40 plus POS tags with three tags of Chunking i.e. I, O and B makes overall count of chunk tag to more than 100, which reduces the ambiguities of the tagger while tagging the test sentence using corpus of 100,000 words. Because processing the test data having count of only B, I and O generates ambiguities but having count of NN_B, NN_I, NN_O, PN_B, PN_I, PN_O and so on (*See Appendix C*), was straight forward for tagger while marking the chunk tags of test corpus. Precision and recall for B tag were 96.05 and 96.35, for I tag 91.88 and 92.23, and for O tag precision and recall were 99.92 and 99.58. The precisions shows that most of the ambiguities found in this method by the statistical system were of I tag. This shows that this system successfully marked the word tokens which were beginning of noun phrases or outside of noun phrases. It could not mark I tag with high accuracy, which means the most ambiguities it found belongs to adjacent nouns. Such adjacent nouns are difficult to mark even manually because normally people do not use commas (phrase markers) to mark different phrases. For example:

اس <PD> موقع <NN> پر <P> صوبائی <ADJ> وزیر <NN> کے <P> ساتھ <NN> سید <PN> ریاض
<PN> بخاری <PN> رانا <PN> ابرار <PN> احمد <PN> امان <PN> اللہ <PN> خان <PN> خلیل <PN>
الرحمان <PN> قریشی <PN> ڈاکٹر <PRT> محمود <PN> خان <PN> یوسف <PN> زئی <PN> حاجی
<PRT> اللہ <PN> بخش <PN> حاجی <PRT> عبدالکریم <PN> ابراہیم <PN> راشد <PN> شیخ <PN>
نصر <PN> اللہ <PN> ناصر <PN> ارشاد <PN> النبی <PN> خان <PN> محمد <PN> نواز <PN> شاہد
<PN> میاں <PRT> اسحاق <PN> فرید <PN> ملک <PN> عبدالحمید <PN> گجر <PN> اور <CC> میاں
<PRT> ثناء <PN> اللہ <PN> موجود <NN> تھے <VB> ۔ <SM>

After comparison of all the experiment using same test corpus and all other conditions kept same it was observed that Experiment 2: Extended Experiment using transformation of All POS outperformed all other experiments with the accuracy of 97.28% which improves only 0.24 after implementation of rules on it and reached to 97.52. In analysis, methodology of this experiment was found best. It is considered that using this method, the number of chunk tags increased to more than 100 tags because in this method we combine both the POS and IOB in training and then only POS tags were presented to the tagger for testing data. By combining 40 plus POS tags with three tags of Chunking i.e. I, O and B makes overall count of chunk tag to more than 100, which reduces the ambiguities of the tagger while tagging the test data. Because processing the test data having count of only B, I and O generates ambiguities but having count of NN_B, NN_I, NN_O, PN_B, PN_I, PN_O and so on (*See Appendix C*), was straight forward for tagger while marking the chunk tags of test corpus. After implementation of rules the accuracy of this method is increased to 97.52%, the analysis about remaining error percentage (2.48%) is made after observing the test corpus. It was revealed that about 40% errors were those which are ambiguous also in manual chunking including consecutive names. Other instances are complex predicate of nouns and pronouns. For example:

بستی <NN> کے <P> مکین <NN> محمد <PN> سلیم <PN> ڈاکٹر <PRT> محمد <PN>
عرفان <PN> محمد <PN> شاکر <PN> محمد <PN> عمران <PN> کامران <PN> مغل <PN>جاوید <PN> مغل
<PN> چودھری <PN> محمد <PN> اشرف <PN> محمد <PN> بشیر <PN>احمد <PN> نے <P> اعلی <ADJ>
حکام <NN> سے <SE> سوئی <PN> گیس <NN>کی <P> فراہمی <NN> کا <P> مطالبہ <NN> کیا <VB> ہے
<TA> - <SM>

Around 15% errors were those which are due to Zair-e-Izaffat which was unhandled in this work. Around 15% errors were induced due to such instances where CA is included in the noun phrase in some places but not in others and we have to select one option. Almost same number of instances was present in the test corpus. Above analysis about experiment 2 is confirmed by observing the base method in which the same HMM system but with different tag set rather ambiguous one, after implementation of rules we got 2.94% improvement but in the case of experiment 2 we obtained only 0.24% improvement in accuracy which clearly shows that probabilistic method couldn't outperform because of ambiguity in only three tags of chunk tag set in contrast with above 100 plus chunk tags of tag set of experiment 2.

It is to note that in experiment 3, concatenation of POS tags of nouns only with chunk tags generates a new output set which produced better results. Though it couldn't out-perform extended experiment (experiment 2) with all POS but it produced better results than base experiment. In this set all other POS are kept intact but only POS of nouns and chunk tags were merged. Statistical tagger was 92.30 % accurate before implementation of rules, which means 1.3 % improvement in accuracy was obtained using this approach. It means, in base experiment the statistical model couldn't mark consecutive noun phrases due to small tag set but as tag set was changed in this experiment, targeting only nouns showed 1.3% improvement in accuracy. An important fact is that after implementation of rules, this method generates 96.31 % accurate tags. This shows 4.01 % improvement after implementation of rules. It means enriching chunk tags of only nouns with terminals information (POS information) makes the tagger to generate errors which can be easily detected by our rule set. Following table is an illustration for comparison of base experiment and extended experiment with combination of noun part of speech only with chunk tags:

Table 12: Comparison of Base Experiment with Extended Experiment with Nouns only

Type of Metrics	Experiment Detail	Statistical Model (Results)	After Rule Implementation (Results)	Improvement
Accuracy (%)	Experiment 1: Base experiment	90.93	93.87	2.94
	Experiment 3: extended experiment using POS of nouns information in chunk tag set	92.30	96.31	4.01

It can be easily seen in above table that experiment 3 out-performs the experiment 1 only using part of speech (POS) information of nouns in chunk tag set.

In this work 97.52 % overall accuracy was achieved for Urdu NP chunking task. This accuracy is mentionable with comparison to different techniques of chunking used for other languages. Following is list of results for chunking task for other languages:

- Chen (1993) reported 98 % chunk correct rate, 94 % sentence correct rate in inside test, and 99 % chunk correct rate and 97 % sentence correct rate in inside test using 1 million words. These results were reported using English language corpus using probabilistic chunker
- Ramshaw et al (1995) reported 92 % precision and recall for baseNPs for English using transformational based learning for corpus of 250,000 words
- Veenstra et al (1998) reported accuracy of 97.2 % with 94.3 % recall. They reported 89.0 % precision for NP chunking. They used memory based learning techniques for English language using corpus of 250,000 words
- Schmid (2000) reported 93.60 % precision and 92.10 % recall using hybrid approach for German grammar for noun phrases using corpus of 1 million words
- Singh (2001) reported 92.63 % precision for chunk boundary identification task for Hindi language using 200,000 words corpus
- Park et al (2003) in their work reported 97.99 % accuracy and 91.87 F-score using only rules. Then using memory based system, they improved 2.3 points F-score to 94.21. The

work was done using Korean language considering four phrases (NP, VP, ADVP, IP) of this language using corpus of 321328 words

- Pammi (2007) reported 69.92 % accuracy for Hindi, 70.99 % for Bengali and 74.74 % for Telugu using decision forests using corpus 25000 words of each language

7 Conclusion and Future Work

7.1 Conclusion

In this work different experiments were conducted using different input and output tag set schemes but with same methodology. The hybrid approach is used, which is combination of statistical and rule based methods. It is observed that high accuracy is extremely influenced by input and output tag sets. More rich out put tag set with POS information produces more accurate results. The overall accuracy of 97.52 % is achieved using the IOB output tag set rich in part of speech information using hybrid approach. It is also observed that output (chunk) tag set having more than 100 tags out-performs in terms of accuracy, precision and recall with corpus of 100,000 word tokens. So, the tag set of three tags (I, O and B) must be modified to a large tag set to get maximum accuracy.

It is also concluded that direction of sentences (Left to Right or Right to Left) has no effect on overall accuracy. The non-overlapping difference of both the directions is ignorable.

7.2 Directions for Future Work

The cases of Zair-e-izaffat were not handled in this work and it is an observation that such cases can improve accuracy of chunk tagger to significant extent. In future work such cases would be handled. This work is done using Tri-gram model of HMM. It is considered that chunking task must be performed by Bi-gram, Uni-gram and Tetra-gram to have comparison that which n-gram suits best for the chunking task.

In this work “Marker Hypothesis” introduced by Green (1979) was used, in which some markers like genitive were excluded from the phrases to mark the boundaries of noun phrases but actually they were part of the phrase. In future work, the chunking task can perform without using this hypothesis.

The next task would be development of a shallow parser to form noun phrases using this work and tags used in this work, so that the noun phrases can be used in full parsing.

Other techniques like Support Vector Machines (SVM), Memory based Chunking, Decision Trees and Decision forests would be investigated in future work for accuracy.

Chunking for other phrases of Urdu like verb phrases and case phrases etc. will be next milestone so that a Treebank can be built using chunking. Such a Treebank will be helpful in development of other applications for Urdu.

Singh (2001) reported 92.63 % precision for chunk boundary identification task for Hindi language using 200,000 words corpus. One possible reason of low accuracy for Hindi might be the fact that in Hindi the case marker is written as part of the noun/ pronoun it is marking. Approach in this work may be used for Hindi language after detaching the case marker from the word to investigate the improvement for that language.

References

Abney S., *Parsing by Chunks*, *Principle based Parsing,* Kluwer Academic Publishers, Dordrecht, 1991.

Brants Thorsten, *TnT: a statistical part-of-speech tagger*, in Proceedings of the sixth conference on Applied natural language processing Seattle, Washington, Pages: 224 – 231, 2000

Chen Huang-hua and Chen Hsin-His , *Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation*. in Proceedings of the 32nd annual meeting on Association for Computational Linguistics Las Cruces, New Mexico Pages: 234 – 41, 1994.

Chen Kuang-hua and Chen Hsin-hsi, A Probabilistic Chunker, in Proceedings of ROCLING VI, 1993.

Croft William, *Typology and universals*, 2nd edition. Cambridge: Cambridge University Press, 2003.

Daelemans Walter, Buchhlolz Sabine and Veenstra Jorn, *Memory-Based Shallow Parsing*, in proceesings of EMNLP/ VLC-99, Pages 239-246, University of Maryland, USA, June 1999.

Dalal Aniket, Nagaraj Kumar, Sawant Uma and Shelke Sandeep , *Hindi Part-of-Speech Tagging and Chunking : A Maximum Entropy Approach*, In Proceedings of the NLPAI Machine Learning Contest 2006 NLPAI, 2006.

Das Dipanjan and Choudhury Monojit, *A Valency Theoretic Approach to Shallow Parsing of Free Word Order Languages,* Presented at the Student Paper Contest at ICON-KBCS. Hyderabad, India, Dec 2004.

Green T., *The necessity of syntax markers: two experiments with artificial language,* Journal of Verbal Learning and Behaviour 18:481-496, 1979.

Grover Claire & Tobil Richard, *Rule Based Chunking and Reusability*, in Proceedings of the Fifth International Conference on Language Resources, 2007.

Haq M. Abdul., *اردو صرف و نحو*, Anjuman-e-Taraqqi Urdu (Hind), 1987.

Hardie A., *Developing a tag-set for automated part-of-speech tagging in Urdu*, In Proceedings of the Corpus Linguistics 2003 conference, UCREL Technical Papers Volume 16, Department of Linguistics, Lancaster University, UK 2003.

Lager T., *A Logical Approach to Computational Corpus Linguistics*, A Doctoral Dissertation, Department of Linguistics, Goteborg University, Sweden, 1971.

Molina Antonio and Pla Ferren, *Shallow Parsing using Specialized HMMs*, Journal of Machine Learning Research 2 (2002) Pages: 595-613, 2002.

Muaz A., Ali A. and Hussain S., *Analysis and Development of Urdu POS Tagged Corpora*, In the Proceedings of the 7th Workshop on Asian Language Resources, IJCNLP'09, Suntec City, Singapore, 2009.

Munoz Marcia, Puyakanok Vasin, Roth Dan and Zimak Dav, *A Learning Approach to Shallow Parsing*, Technical Report: UIUCDCS-R-99-2087, University of Illinois at Urbana-Champaign Champaign, IL, USA, 1999.

Pammi Sathish Chandra and Prahallad Kishore , *POS Tagging and Chunking using Decision Forests*, in Proceedings of Workshop on Shallow Parsing in South Asian Languages at IJCAI, 2007.

Park Seong-Bae and Zhang Byoung-Tak, *Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning*, Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, Pages: 497 – 504, 2003.

Platts John T., *A Grammar of the Hindustani or Urdu Language*, London, 1909.

Quirk Randolph, Svartvik J. and Leech G, "A Comprehensive Grammar of the English Language", Longman Group Limited, England, 1985.

Ramshaw Lance A. and Marcus Mitchell P., *Text chunking using transformation based learning*, in proceedings of the third ACL workshop on Very Large Corpora, Somerset, NJ., pp. 82-94, 1995.

Rao Delip and Yarowsky David, *Part of speech tagging and shallow parsing of Indian Languages,* in Proceedings of the workshop on Shallow Parsing for South Asian Languages, at the International Joint Conference in Artificial Intelligence (IJCAI), 2007.

Robins R. H., *General Linguistics*, 4th ed. London: Longman, 1989.

Sajjad H., *Statistical Part of Speech Tagger for Urdu*, Master of Science Thesis, Department of Computer Sciences, National University of Computer and Emerging Sciences, Lahore, Pakistan, 2007.

Schmid Helmut and Walde Sabine Schulte im, *Robust German Noun Chunking with Probabilistic Context-Free Grammar*, in Proceedings of COLING 2000, 2000.

Shen Libsin and Joshi Aravind K., *A SNoW based Supertagger with Application to NP Chunking*, in Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, Pages: 505 – 512, 2003.

Singh Akshay, Bendre Sushma and Sangal Rajeev 2001, *HMM Based Chunker for Hindi,* in Proceedings of Posters, Intl. Joint Conf. on Natural Language Processing (IJCNLP): 2001.

Stav Adi., *Shallow Parsing*, Seminar in Natural Language Processing and Computational Linguistics, June 17th, 2006.

Su GuoDong Zhou Jian, *Named Entity Recognition using an HMM-based Chunk Tagger*, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics , Pages: 473 – 480, 2001.

Thomson A. J. and Martinet A.V., *A Practical English Grammar*, Oxford University Press, 1986.

Veenstran Jorn and Buchholz Sabine, *Fast NP Chunking Using Memory-Based Learning Techniques*, in Proceedings of BENELEARN'98, 1998.

Voutilainen Atro, *NPTool, A detector of English Noun Phrases*, Workshop on Very Large Corpora: Academic and Industrial Perspectives: 1993.

Appendices

Appendix A: Terms of Reference

Following are decisions taken while manual preparation of corpus:

1. Adjectives alone will be marked outside (O) of noun phrases.
2. Numerals alone will be tagged outside (O).
3. Date shows behavior same like nouns and tagged DATE in annotated corpus used in this work. For maintaining training data for learning this tag is replaced with noun tag (NN).
4. All case markers will be marked outside (O).
5. Coordinate conjunction (CC) will be marked outside (O) if it is present between nouns. Coordinate conjunction between adjective will be considered inside (I) the phrase if followed by noun.
6. If adjective (ADJ) is present after noun and is not followed by the noun. Such adjective will be considered outside to avoid excessive exceptions for computational model.
7. Zair-e-izafat will not be treated specially between adjectives and nouns.
8. Numeral after nouns not followed by noun will be treated as outside (O) noun phrase.
9. Pronouns will be marked as stand alone noun phrase.
10. Units (U) will be treated as nouns though their tag will not be upgraded to noun.
11. Intensifiers (I) will be considered outside (O) noun phrases.
12. Relative pronoun (REP) will be marked as standalone noun phrase (NP).
13. If سے (SE) tag is present between two adjectives to show range, then first adjective will be marked outside (O) and the second adjective is followed by noun will be marked beginning (B).

Appendix B: Rule Set of Experiments

1. If coordinate conjunction is present between two adjectives then followed by noun is marked outside by system.
    a. Mark such coordinate conjunction (CC) inside
    b. Also mark adjective after coordinate conjunction (CC) as inside (I).

   Following is an example of error and then correction by rule.

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
کم	ADJ	B	B
از	CC	O	I
کم	ADJ	B	I
پبلک	NN	I	I
مقامات	NN	I	I
پر	P	O	O
قدغن	NN	B	B
لگانا	VB	O	O
وقت	NN	B	B
کی	P	O	O
بنیادی	ADJ	B	B
ضرورت	NN	I	I
ہے	VB	O	O
-	SM	O	O

2. If adjective not followed by noun but has preceding noun is marked inside by the system then mark it outside.

For example:

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
ضلعی	ADJ	B	B
ناظم	NN	I	I
چودھری	PN	B	B

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
طارق	PN	I	I
بشیر	PN	I	I
چیمہ	PN	I	I
نے	P	O	O
ایک	CA	B	B
پریس	NN	I	I
کانفرنس	NN	I	I
کے	P	O	O
دوران	NN	B	B
بتایا	VB	O	O
تھا	TA	O	O
کہ	SC	O	O
یہ	PD	B	B
منصوبہ	NN	I	I
مکمل	ADJ	I	O
ہونے	VB	O	O
کی	P	O	O
مدت	NN	B	B
دسمبر	PN	B	B
2004ء	NN	I	I
تک	P	O	O
ہے	VB	O	O

3. If adjective has proceeding adjective which is not followed by noun then mark both adjectives outside (O). For example:

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
شہر	NN	B	B
کو	P	O	O
صاف	ADJ	B	O

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
ستھرا	ADJ	I	O
رکھنا	VB	O	O
صرف	ADV	O	O
شہریوں	NN	B	B
کی	P	O	O
ذمہداری	NN	B	B
ہے	VB	O	O
بلکہ	SC	O	O
صحتمند	ADJ	B	B
معاشرے	NN	I	I
کے	P	O	O
لئے	NN	B	B
انتہائی	ADV	O	O
ضروری	ADJ	B	B
ہے	VB	O	O

4. If stand alone adjectives which don't have adjacent noun are marked Beginning (B), then mark such adjectives as Outside (O). Following is an example of such an error of tagger and correction by the rule.

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
شہر	NN	B	B
کو	P	O	O
صاف	ADJ	B	O
ستھرا	ADJ	I	O
رکھنا	VB	O	O
صرف	ADV	O	O
شہریوں	NN	B	B
کی	P	O	O
ذمہداری	NN	B	B
ہے	VB	O	O

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
بلکہ	SC	O	O
صحتمند	ADJ	B	B
معاشرے	NN	I	I
کے	P	O	O
لئے	NN	B	B
انتہائی	ADV	O	O
ضروری	ADJ	B	O
ہے	VB	O	O

5. If stand alone Ordinals (OR) which don't have adjacent noun are marked Beginning (B), then mark such Ordinals as Outside (O). Following is an example of such an error of tagger and correction by the rule.

Word Tokens	POS Tags	Chunk Tags Generated by Statistical Tagger	After implementation of this Rule
ان	PP	B	B
سے	SE	O	O
پہلے	OR	B	O
بل	PN	B	B
کلنٹن	PN	I	I
کی	P	O	O
خارجہ	ADJ	B	B
پالیسی	NN	I	I
کا	P	O	O
اہم	ADJ	B	B
محور	NN	I	I
مسئلہ	NN	B	B
فلسطین	PN	I	I
تھا	VB	O	O
۔	SM	O	O

6. If stand alone Cardinals (CA) which don't have adjacent noun are marked Beginning (B) or Inside (I), then mark such Cardinals as Outside (O). Following is an example of such an error of tagger and correction by the rule.

Word Tokens	**POS tags**	**Chunk tags Generated by Statistical Tagger**	**After Implementation of this rule**
غزہ	PN	B	B
کی	P	O	O
پٹی	NN	B	B
کے	P	O	O
تین	CA	B	O
چوتھائی	FR	I	I
اور	CC	O	O
مغربی	ADJ	B	B
کنارے	NN	I	I
کے	P	O	O
40	CA	B	O
فیصد	ADV	O	O
حصے	NN	B	B
پر	P	O	O
یہ	PD	B	B
ریاست	NN	I	I
قائم	NN	I	I
ہوگی	VB	O	O
۔	SM	O	O

7. If adjacent same nouns are marked as two different noun phrases then mark adjacent same Nouns like جگہ جگہ as same phrase.

For example:

Word Tokens	**POS tags**	**Chunk tags Generated by Statistical Tagger**	**After Implementation of this rule**
جبکہ	ADV	O	O
چوک	NN	B	B
فوارہ	PN	I	I

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
میلاد	PN	B	B
چوک	NN	I	I
سرائیکی	PN	B	B
چوک	NN	I	I
پر	P	O	O
بھی	I	O	O
جگہ	NN	B	B
جگہ	NN	B	I
تجاوزات	NN	I	I
نظر	NN	B	B
آتے	VB	O	O
ہیں	TA	O	O
-	SM	O	O

8. If another noun is present after two adjacent same nouns, and is marked Inside (I) then mark such a noun as Beginning (B) of new phrase. Following example explains error of system and correct by the rule:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
جبکہ	ADV	O	O
چوک	NN	B	B
فوارہ	PN	I	I
میلاد	PN	B	B
چوک	NN	I	I
سرائیکی	PN	B	B
چوک	NN	I	I
پر	P	O	O
بھی	I	O	O
جگہ	NN	B	B
جگہ	NN	I	I
تجاوزات	NN	I	I

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
نظر	NN	B	B
آتے	VB	O	O
ہیں	TA	O	O
۔	SM	O	O

9. A Cardinal (CA) which is not followed by adjective or noun is marked Beginning (B) or Inside (I) by system, then mark such Cardinal (CA) as Outside (O). Illustration of error and correction by rule is given below:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
صدر	NN	B	B
بش	PN	I	I
نے	P	O	O
یہ	PP	B	B
بھی	I	O	O
کہا	VB	O	O
کہ	SC	O	O
ان	PP	B	B
کے	PK	O	O
اتحادیوں	NN	B	B
کی	PK	O	O
تعداد	**NN**	**B**	**B**
30	**CA**	**B**	**O**
ہے	**VB**	**O**	**O**
۔	SM	O	O

10. A Cardinal (CA) is followed by adjective and adjective is followed by noun if marked Outside (O) or Inside (I) by system, them mark Cardinals as Beginning (B). For Example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
اس	PP	B	B
کے	PK	O	O
لئے	NN	B	B
ایک	CA	O	B
بھرپور	ADJ	B	B
ایکشن	NN	I	I
کی	PK	O	O
ضرورت	NN	B	B
تھی	VB	O	O
۔	SM	O	O

11. A Cardinal (CA) is preceded by adjective and also followed by noun if marked Outside (O) or Inside (B) by system, them mark Cardinals as Beginning (I). For Example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
گزشتہ	ADJ	B	B
دو	CA	B	I
سال	NN	I	I
سے	SE	O	O
سرکاری	ADJ	B	B
عمارات	NN	I	I
اور	CC	O	O
کالونیوں	NN	B	B
کی	PK	O	O
مرمت	NN	B	B
کے	PK	O	O
لئے	NN	B	B
ایک	CA	B	B
کوڑی	NN	I	I
بھی	I	O	O

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
فراہم	NN	B	B
نہیں	NEG	O	O
کی	VB	O	O
گئی	AA	O	O

12. If a Pre-title (PRT) is followed by another PRT then second will be marked Inside (I). For Example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
سابق	ADJ	B	B
گورنر	NN	I	I
لیفٹیننٹ	PRT	B	B
جنرل	PRT	B	I
محمد	PN	I	I
اقبال	PN	I	I
خان	PN	I	I
کے	PK	O	O
بعد	NN	B	B
کسی	KD	B	B
ادارے	NN	I	I
نے	P	O	O
اس	PD	B	B
اڈے	NN	I	I
کی	PK	O	O
بہتری	NN	B	B
کی	PK	O	O
جانب	NN	B	B
کوئی	PD	B	B
توجہ	NN	I	I
نہیں	NEG	O	O

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
دی	VB	O	O
۔	SM	O	O

13. A cardinal is followed by Fraction (FR) which is followed by noun. If such a fraction is marked Outside (O) or Inside (B) by the system then mark Fraction (FR) as Inside (I). For example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
اس	PD	B	B
آمدنی	NN	I	I
کا	PK	O	O
ایک	CA	B	B
چوتھائی	FR	B	I
حصہ	NN	I	I
اس	PP	B	B
کی	PK	O	O
تعمیر	NN	B	B
و	CC	O	O
ترقی	NN	B	B
پر	P	O	O
ضرور	ADV	O	O
خرچ	NN	B	B
ہونا	VB	O	O
چاہیے	AA	O	O

14. If quantifier (Q) is not followed by Noun or Adjective and is marked Beginning (B) or Inside (I) by the system then it must be marked Outside (O). Illustration of this rule is as under:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
خاوند	NN	B	B
کے	PK	O	O

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
قتل	NN	B	B
کے	PK	O	O
بعد	NN	B	B
ہمارے	G	B	B
پاس	NN	I	I
دکھوں	NN	I	I
کے	PK	O	O
علاوہ	NN	B	B
کچھ	Q	B	O
نہیں	NEG	O	O
۔	SM	O	O

15. A genitive is succeeded by adjective which is followed by noun. If such adjective and noun are marked Outside (O) or Beginning (B) by the system, then mark such adjective and noun as Inside (I) of genitive phrase. Following is an example of error and correction by using rule:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
ڈاکٹروں	NN	B	B
نے	P	O	O
اپنے	GR	B	B
پرائیویٹ	ADJ	B	I
کلینک	NN	I	I
سجا	VB	O	O
لئے	AA	O	O
ہیں	TA	O	O
۔	SM	O	O

16. All the pronouns are marked stand alone noun phrase. If Tagger could not follow this pattern then mark all pronouns as beginning tag (B). To mark it as stand alone noun phrase, ensure that proceeding token is not marked Inside (I). Example of error and correction is given below:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
مگر	SC	O	O
مجھے	PP	B	B
امید	NN	I	B
ہے	VB	O	O

17. A Cardinal is followed by Cardinal which is followed by Noun. If the second cardinal and Noun are not marked Inside (I) by the system then mark them with Inside tag (I). For example

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
قصاب	NN	B	B
ایک	CA	O	B
دو	CA	B	I
جانور	NN	I	I
مذبحہخانہ	NN	I	I
میں	P	O	O
ذبح	NN	B	B
کرتے	VB	O	O
ہیں	TA	O	O
اور	CC	O	O
باقی	Q	B	B
جانوروں	NN	I	I
کو	P	O	O
اپنے	GR	B	B
گھروں	NN	I	I
میں	P	O	O
ذبح	NN	B	B
کر	VB	O	O
لیتے	AA	O	O
ہیں	TA	O	O

18. Adjective is followed by adjective and then noun then second adjective and noun will be marked as Inside (I) and first adjective will be marked as B. If tagger could not produce this output, then use this rule to correct the tags produced by tagger. Illustration of error and correction using this rule is given below:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
موٹر	NN	B	B
سائیکلوں	NN	I	I
وغیرہ	NN	I	I
کی	PK	O	O
لمبی	ADJ	B	B
لمبی	ADJ	B	I
قطاریں	NN	I	I
نظر	NN	B	B
آتی	VB	O	O
ہیں	TA	O	O
۔	SM	O	O

19. Cardinal is followed by Adjective and then Noun. Such Adjective and Noun will be marked inside (I) and the Cardinal (CA) will be marked Beginning (B). If tagger could not produce this pattern then by using rule correct the tagger output. For example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
شہر	NN	B	B
میں	P	O	O
مزید	ADJ	B	B
تین	CA	O	O
چار	CA	B	B
نئے	ADJ	B	I
مذبحہخانہ	NN	I	I
قائم	NN	I	I
کئے	VB	O	O
جائیں	AA	O	O

20. If relative pronoun (REP) is marked Beginning (B) and proceeding token as Inside (I) by the system, then mark such a proceeding token Beginning (B) if it is beginning of a noun phrase or Outside (O) otherwise. For example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
جو	REP	B	B
حادثے	NN	I	B
کا	PK	O	O
سبب	NN	B	B
بن	VB	O	O
سکتے	AA	O	O
ہیں	TA	O	O
۔	SM	O	O

21. A demonstrative is followed by adjective (ADJ) then noun or by noun (NN/ PN). Such an adjective and noun is marked Outside (O) or Beginning (B) by the system then mark both inside (I). For example.

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
پنجاب	PN	B	B
حکومت	NN	I	I
ہر	ADJ	B	B
سال	NN	I	I
ان	PD	B	B
سرکاری	ADJ	B	I
عمارات	NN	I	I
کی	PK	O	O
مرمت	NN	B	B
و	CC	O	O
دیکھ	NN	B	B
بھال	NN	I	I
کے	PK	O	O

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
لئے	NN	B	B
باقاعدگی	NN	B	B
سے	SE	O	O
2	CA	B	B
کروڑ	CA	I	I
روپے	NN	I	I
فراہم	NN	B	B
کرتی	VB	O	O
تھی	TA	O	O
-	SM	O	O

22. If Adjective is immediately followed by Noun (NN/ PN) and is marked Outside (O) by tagger then mark it Beginning (B). For example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
ہر	ADJ	B	B
روز	NN	I	I
بڑے	ADJ	O	B
واقعات	NN	B	B
رونما	NN	I	I
ہوتے	VB	O	O
ہیں	TA	O	O
لیکن	SC	O	O
پولیس	NN	B	B
بےبس	ADJ	B	B
نظر	NN	I	I
آتی	VB	O	O
ہے	TA	O	O
-	SM	O	O

23. If Particle is marked Beginning (B) by tagger, then mark it Outside (O). For Example:

Word Tokens	POS tags	Chunk tags Generated by Statistical Tagger	After Implementation of this rule
ان	PP	B	B
کے	P	B	O
حریف	ADJ	B	B
ڈیمو	PN	I	I
کریٹک	PN	I	I
پارٹی	PN	I	I
کے	P	B	O
مضبوط	ADJ	B	B
امیدوار	NN	I	I
جان	PN	B	B
کیری	PN	I	I
کو	P	B	O
بیس	CA	B	B
ریاستوں	NN	I	I
میں	P	B	O
کامیابی	NN	B	B
ہوئی	VB	O	O
ہے	TA	O	O
-	SM	O	O

Appendix C: Tag Sequence Examples of Experiments

Following tables illustrate the training data tag sequence of each experiment.

Training Tag sequence of Experiment 1A: Base Experiment using Basic Methodology Right to Left Direction (Sample Data)

Column 2 and Column 3 were presented to Statistical tagger as training data while training. After Training only Column 2 was given to tagger as Testing data and the output of the tagger was Column 3.

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
پولیس	NN	B
کے	P	O
ہاتھوں	NN	B
ظلم	NN	B
و	CC	O
زیادتی	NN	B
کی	P	O
خبروں	NN	B
نے	P	O
لوگوں	NN	B
کا	P	O
اعتماد	NN	B
مجروح	NN	B
کیا	VB	O
ہے	TA	O
۔	SM	O
پولیس	NN	B
کو	P	O
لامحدود	ADJ	B
اختیارات	NN	I
حاصل	NN	B

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
ہیں	VB	O
۔	SM	O
جنہیں	REP	B
وہ	PD	B
تفتیش	NN	B
کے	P	O
دوران	NN	B
بےجا	ADJ	B
استعمال	NN	I
کرتی	VB	O
ہے	TA	O
۔	SM	O
پولیس	NN	B
کے	P	O
ہاتھوں	NN	B
خواتین	NN	B
کی	P	O
تذلیل	NN	B
کا	P	O
خبریں	NN	B
عموما	ADV	O
اخبارات	NN	B
کی	P	O
زینت	NN	B
بنتی	VB	O
رہتی	AA	O
ہیں	TA	O
۔	SM	O
قانون	NN	B
کے	P	O

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
محافظوں	NN	B
کی	P	O
ان	PD	B
حرکتوں	NN	I
سے	SE	O
پولیس	NN	B
کے	P	O
محکمے	NN	B
کی	P	O
بدنامی	NN	B
ہوتی	VB	O
ہے	TA	O
بلکہ	SC	O
لوگوں	NN	B
کا	P	O
اعتماد	NN	B
بھی	I	O
مجروح	NN	B
ہوتا	VB	O
ہے	TA	O
-	SM	O

Training Tag sequence of Experiment 1B: Base Experiment using Basic Methodology Left to Right Direction (Sample Data)

Column 2 and Column 3 were presented to Statistical tagger as training data while training. After Training only Column 2 was given to tagger as Testing data and the output of the tagger was Column 3.

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
۔	SM	O
ہے	TA	O
ہوتا	VB	O
مجروح	NN	B
بھی	I	O
اعتماد	NN	B
کا	P	O
لوگوں	NN	B
بلکہ	SC	O
ہے	TA	O
ہوتی	VB	O
بدنامی	NN	B
کی	P	O
محکمے	NN	B
کے	P	O
پولیس	NN	B
سے	SE	O
حرکتوں	NN	I
ان	PD	B
کی	P	O
محافظوں	NN	B
کے	P	O

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
قانون	NN	B
-	SM	O
ہیں	TA	O
رہتی	AA	O
بنتی	VB	O
زینت	NN	B
کی	P	O
اخبارات	NN	B
عموما	ADV	O
خبریں	NN	B
کا	P	O
تذلیل	NN	B
کی	P	O
خواتین	NN	B
ہاتھوں	NN	B
کے	P	O
پولیس	NN	B
-	SM	O
ہے	TA	O
کرتی	VB	O
استعمال	NN	I
بےجا	ADJ	B

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
دوران	NN	B
کے	P	O
تفتیش	NN	B
وہ	PD	B
جنہیں	REP	B
۔	SM	O
ہیں	VB	O
حاصل	NN	B
اختیارات	NN	I
لامحدود	ADJ	B
کو	P	O
پولیس	NN	B
۔	SM	O
ہے	TA	O
کیا	VB	O
مجروح	NN	B
اعتماد	NN	B
کا	P	O
لوگوں	NN	B
نے	P	O
خبروں	NN	B
کی	P	O

1	2	3
Word Tokens	**POS Tags**	**Chunk Tags**
زیادتی	NN	B
و	CC	O
ظلم	NN	B
ہاتھوں	NN	B
کے	P	O
پولیس	NN	B

Tag Sequence of Experiment 2: Extended Experiment using Transformation of All POS (Sample Data)

Column 2 and Column 3 were presented to Statistical tagger as training data while training. After Training only Column 2 was given to tagger as Testing data and the output of the tagger was Column 3 which then split into POS tags and Chunk tags and the Chunk tags were compared with Column 4 (Manually Marked) for evaluation.

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Both POS Tags and Chunk Tags**	**Chunk Tags**
پولیس	NN	NN_B	B
کے	P	P_O	O
ہاتھوں	NN	NN_B	B
ظلم	NN	NN_B	B
و	CC	CC_O	O
زیادتی	NN	NN_B	B
کی	P	P_O	O
خبروں	NN	NN_B	B
نے	P	P_O	O

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Both POS Tags and Chunk Tags**	**Chunk Tags**
لوگوں	NN	NN_B	B
کا	P	P_O	O
اعتماد	NN	NN_B	B
مجروح	NN	NN_B	B
کیا	VB	VB_O	O
ہے	TA	TA_O	O
۔	SM	SM_O	O
پولیس	NN	NN_B	B
کو	P	P_O	O
لامحدود	ADJ	ADJ_B	B
اختیارات	NN	NN_I	I
حاصل	NN	NN_B	B
ہیں	VB	VB_O	O
۔	SM	SM_O	O
جنہیں	REP	REP_B	B
وہ	PD	PD_B	B
تفتیش	NN	NN_B	B
کے	P	P_O	O
دوران	NN	NN_B	B
بےجا	ADJ	ADJ_B	B
استعمال	NN	NN_I	I
کرتی	VB	VB_O	O
ہے	TA	TA_O	O
۔	SM	SM_O	O
پولیس	NN	NN_B	B
کے	P	P_O	O
ہاتھوں	NN	NN_B	B
خواتین	NN	NN_B	B

1	2	3	4
Word Tokens	POS Tags	Combination of Both POS Tags and Chunk Tags	Chunk Tags
کی	P	P_O	O
تذلیل	NN	NN_B	B
کا	P	P_O	O
خبریں	NN	NN_B	B
عموما	ADV	ADV_O	O
اخبارات	NN	NN_B	B
کی	P	P_O	O
زینت	NN	NN_B	B
بنتی	VB	VB_O	O
رہتی	AA	AA_O	O
ہیں	TA	TA_O	O
-	SM	SM_O	O
قانون	NN	NN_B	B
کے	P	P_O	O
محافظوں	NN	NN_B	B
کی	P	P_O	O
ان	PD	PD_B	B
حرکتوں	NN	NN_I	I
سے	SE	SE_O	O
پولیس	NN	NN_B	B
کے	P	P_O	O
محکمے	NN	NN_B	B
کی	P	P_O	O
بدنامی	NN	NN_B	B
ہوتی	VB	VB_O	O
ہے	TA	TA_O	O
بلکہ	SC	SC_O	O
لوگوں	NN	NN_B	B

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Both POS Tags and Chunk Tags**	**Chunk Tags**
کا	P	P_O	O
اعتماد	NN	NN_B	B
بھی	I	I_O	O
مجروح	NN	NN_B	B
ہوتا	VB	VB_O	O
ہے	TA	TA_O	O
۔	SM	SM_O	O

Tag Sequence of Experiment 3: Extended Experiment using Transformation of Nouns Only (Sample Data)

Column 2 and Column 3 were presented to Statistical tagger as training data while training. After Training only Column 2 was given to tagger as Testing data and the output of the tagger was Column 3. Chunk Tags then separated from tagger's output and compared with column 4 (Manually Marked Chunk Tags) to get results of evaluation metrics.

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Nouns with Chunk Tags**	**Chunk Tags**
پولیس	NN	NN_B	B
کے	P	O	O
ہاتھوں	NN	NN_B	B
ظلم	NN	NN_B	B
و	CC	O	O
زیادتی	NN	NN_B	B
کی	P	O	O
خبروں	NN	NN_B	B
نے	P	O	O

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Nouns with Chunk Tags**	**Chunk Tags**
لوگوں	NN	NN_B	B
کا	P	O	O
اعتماد	NN	NN_B	B
مجروح	NN	NN_B	B
کیا	VB	O	O
ہے	TA	O	O
۔	SM	O	O
پولیس	NN	NN_B	B
کو	P	O	O
لامحدود	ADJ	B	B
اختیارات	NN	NN_I	I
حاصل	NN	NN_B	B
ہیں	VB	O	O
۔	SM	O	O
جنہیں	REP	B	B
وہ	PD	B	B
تفتیش	NN	NN_B	B
کے	P	O	O
دوران	NN	NN_B	B
بےجا	ADJ	B	B
استعمال	NN	NN_I	I
کرتی	VB	O	O
ہے	TA	O	O
۔	SM	O	O
پولیس	NN	NN_B	B
کے	P	O	O
ہاتھوں	NN	NN_B	B
خواتین	NN	NN_B	B
کی	P	O	O

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Nouns with Chunk Tags**	**Chunk Tags**
تذلیل	NN	NN_B	B
کا	P	O	O
خبریں	NN	NN_B	B
عموما	ADV	O	O
اخبارات	NN	NN_B	B
کی	P	O	O
زینت	NN	NN_B	B
بنتی	VB	O	O
رہتی	AA	O	O
ہیں	TA	O	O
-	SM	O	O
قانون	NN	NN_B	B
کے	P	O	O
محافظوں	NN	NN_B	B
کی	P	O	O
ان	PD	B	B
حرکتوں	NN	NN_I	I
سے	SE	O	O
پولیس	NN	NN_B	B
کے	P	O	O
محکمے	NN	NN_B	B
کی	P	O	O
بدنامی	NN	NN_B	B
ہوتی	VB	O	O
ہے	TA	O	O
بلکہ	SC	O	O
لوگوں	NN	NN_B	B
کا	P	O	O
اعتماد	NN	NN_B	B

1	2	3	4
Word Tokens	**POS Tags**	**Combination of Nouns with Chunk Tags**	**Chunk Tags**
بھی	I	O	O
مجروح	NN	NN_B	B
ہوتا	VB	O	O
ہے	TA	O	O
-	SM	O	O

Appendix D: Results for rule implementation in experiments

In this Appendix effect of rules on each experiment is discussed in detail.

Experiment 1: Base Experiment using basic methodology

Following table explains the role of individual rules of experiment 1 A (Right to left direction execution) in over all accuracy.

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
907	Statistical Input (I1)	Rule 1A	O1	899	8.9918	91.0082
899	O1	Rule 1B	O2	891	8.9118	91.08822
891	O2	Rule 2	O3	859	8.5917	91.40828
859	O3	Rule 3	O4	858	8.5817	91.41828
858	O4	Rule 4	O5	773	7.7315	92.26845
773	O5	Rule 5	O6	773	7.7315	92.26845
773	O6	Rule 6	O7	755	7.5515	92.44849
755	O7	Rule 7A	O8	755	7.5515	92.44849
755	O8	Rule 7B	O9	755	7.5515	92.44849
755	O9	Rule 8	O10	752	7.5215	92.4785
752	O10	Rule 9	O11	734	7.3415	92.65853
734	O11	Rule 10	O12	730	7.3015	92.69854
730	O12	Rule 11	O13	725	7.2515	92.74855
725	O13	Rule 12	O14	723	7.2314	92.76855
723	O14	Rule 13	O15	722	7.2214	92.77856
722	O15	Rule 14	O16	714	7.1414	92.85857
714	O16	Rule 15A	O17	714	7.1414	92.85857
714	O17	Rule 15B	O18	711	7.1114	92.88858
711	O18	Rule 15C	O19	711	7.1114	92.88858
711	O19	Rule 16A	O20	667	6.6713	93.32867
667	O20	Rule 16B	O21	667	6.6713	93.32867
667	O21	Rule 17A	O22	646	6.4613	93.53871

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
646	O22	Rule 17B	O23	643	6.4313	93.56871
643	O23	Rule 18A	O24	640	6.4013	93.59872
640	O24	Rule 18B	O25	640	6.4013	93.59872
640	O25	R19A	O26	640	6.4013	93.59872
640	O26	Rule 19B	O27	630	6.3013	93.69874
630	O27	Rule 20	O28	626	6.2613	93.73875
626	O28	Rule 21	O29	619	6.1912	93.80876
619	O29	Rule 22	O30	616	6.1612	93.83877
619	O30	Rule 23	O31	613	6.1312	93.86877

Following table explains the role of individual rules of experiment 1 B (Left to right direction execution) in over all accuracy.

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
914	SI1	Rule 1A	O1	905	9.0518	90.94819
905	O1	Rule 1B	O2	898	8.9818	91.0182
898	O2	Rule 2	O3	866	8.6617	91.33827
866	O3	Rule 3	O4	865	8.6517	91.34827
865	O4	Rule 4	O5	779	7.7916	92.20844
779	O5	Rule 5	O6	779	7.7916	92.20844
779	O6	Rule 6	O7	765	7.6515	92.34847
765	O7	Rule 7A	O8	765	7.6515	92.34847
765	O8	Rule 7B	O9	765	7.6515	92.34847
765	O9	Rule 8	O10	762	7.6215	92.37848
762	O10	Rule 9	O11	745	7.4515	92.54851
745	O11	Rule 10	O12	740	7.4015	92.59852
740	O12	Rule 11	O13	735	7.3515	92.64853
735	O13	Rule 12	O14	733	7.3315	92.66853
733	O14	Rule 13	O15	732	7.3215	92.67854
732	O15	Rule 14	O16	724	7.2414	92.75855

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
724	O16	Rule 15A	O17	724	7.2414	92.75855
724	O17	Rule 15B	O18	721	7.2114	92.78856
721	O18	Rule 15C	O19	721	7.2114	92.78856
721	O19	Rule 16A	O20	677	6.7714	93.22865
677	O20	Rule 16B	O21	677	6.7714	93.22865
677	O21	Rule 17A	O22	656	6.5613	93.43869
656	O22	Rule 17B	O23	650	6.5013	93.4987
650	O23	Rule 18A	O24	647	6.4713	93.52871
647	O24	Rule 18B	O25	647	6.4713	93.52871
647	O25	R19A	O26	647	6.4713	93.52871
647	O26	Rule 19B	O27	637	6.3713	93.62873
637	O27	Rule 20	O28	633	6.3313	93.66873
633	O28	Rule 21	O29	625	6.2513	93.74875
625	O29	Rule 22	O30	621	6.2112	93.78876
621	O30	Rule 23	O31	621	6.2112	93.78876

Experiment 2: Extended Experiment using Transformation of All POS

Following table explains the role of individual rules of experiment 3 in over all accuracy of experiment.

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
271	SI1	Normalization	SI2	270	2.7005	97.29946
270	SI2	Rule 1A	O1	270	2.7005	97.29946
270	O1	Rule 1B	O2	270	2.7005	97.29946
270	O2	Rule 2	O3	270	2.7005	97.29946
270	O3	Rule 3	O4	270	2.7005	97.29946
270	O4	Rule 4	O5	264	2.6405	97.35947
264	O5	Rule 5	O6	264	2.6405	97.35947

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
264	O6	Rule 6	O7	263	2.6305	97.36947
263	O7	Rule 7A	O8	254	2.5405	97.45949
254	O8	Rule 7B	O9	254	2.5405	97.45949
254	O9	Rule 8	O10	252	2.5205	97.4795
252	O10	Rule 9	O11	251	2.5105	97.4895
251	O11	Rule 10	O12	251	2.5105	97.4895
251	O12	Rule 11	O13	251	2.5105	97.4895
251	O13	Rule 12	O14	251	2.5105	97.4895
251	O14	Rule 13	O15	251	2.5105	97.4895
251	O15	Rule 14	O16	251	2.5105	97.4895
251	O16	Rule 15A	O17	251	2.5105	97.4895
251	O17	Rule 15B	O18	251	2.5105	97.4895
251	O18	Rule 15C	O19	251	2.5105	97.4895
251	O19	Rule 16A	O20	251	2.5105	97.4895
251	O20	Rule 16B	O21	252	2.5205	97.4795
251	O20	Rule 17A	O22	251	2.5105	97.4895
251	O22	Rule 17B	O23	253	2.5305	97.46949
251	O22	Rule 18A	O24	255	2.5505	97.44949
251	O22	Rule 18B	O25	251	2.5105	97.4895
251	O22	R19A	O26	252	2.5205	97.4795
251	O22	Rule 19B	O27	251	2.5105	97.4895
251	O27	Rule 20	O28	251	2.5105	97.4895
251	O28	Rule 21	O29	251	2.5105	97.4895
251	O29	Rule 22	O30	251	2.5105	97.4895
251	O30	Rule 23	O31	248	2.4805	97.5195

Experiment 3: Extended Experiment using Transformation of POS Only

Following table explains the role of individual rules of experiment 4 in over all accuracy of experiment.

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
570	SI1	Normalization	SI2	569	5.6911	94.30886
569	SI2	Rule 1A	O1	569	5.6911	94.30886
569	O1	Rule 1B	O2	569	5.6911	94.30886
569	O2	Rule 2	O3	568	5.6811	94.31886
568	O3	Rule 3	O4	568	5.6811	94.31886
568	O4	Rule 4	O5	485	4.851	95.14903
485	O5	Rule 5	O6	485	4.851	95.14903
485	O6	Rule 6	O7	468	4.6809	95.31906
468	O7	Rule 7A	O8	453	4.5309	95.46909
453	O8	Rule 7B	O9	453	4.5309	95.46909
453	O9	Rule 8	O10	457	4.5709	95.42909
453	O9	Rule 9	O11	450	4.5009	95.4991
453	O11	Rule 10	O12	449	4.4909	95.5091
449	O12	Rule 11	O13	449	4.4909	95.5091
449	O13	Rule 12	O14	449	4.4909	95.5091
449	O14	Rule 13	O15	449	4.4909	95.5091
449	O15	Rule 14	O16	444	4.4409	95.55911
444	O16	Rule 15A	O17	444	4.4409	95.55911
444	O17	Rule 15B	O18	444	4.4409	95.55911
444	O18	Rule 15C	O19	444	4.4409	95.55911
444	O19	Rule 16A	O20	378	3.7808	96.21924
378	O20	Rule 16B	O21	378	3.7808	96.21924
378	O21	Rule 17A	O22	378	3.7808	96.21924
378	O22	Rule 17B	O23	380	3.8008	96.19924
378	O22	Rule 18A	O24	382	3.8208	96.17924
378	O22	Rule 18B	O25	378	3.7808	96.21924

Errors	Input	Firing of Rules	Output	Errors	Error %	Accuracy %
378	O25	R19A	O26	379	3.7908	96.20924
378	O25	Rule 19B	O27	378	3.7808	96.21924
378	O27	Rule 20	O28	373	3.7307	96.26925
373	O28	Rule 21	O29	372	3.7207	96.27926
372	O29	Rule 22	O30	372	3.7207	96.27926
372	O30	Rule 23	O31	369	3.6907	96.30926