Two Pass Parsing Implementation for an Urdu Grammar Checker

Hammad Kabir , Shanza Nayyer , Jahangir Zaman , and Dr. Sarmad Hussain
Center for Research in Urdu Language Processing (CRULP),
National University of Computer and Emerging Sciences (NUCES), Lahore, Pakistan
832@nu.edu.pk, shanza_nayyer@hotmail.com, 806@nu.edu.pk, sarmad.hussain@nu.edu.pk

Abstract: *The research brings forward a computational model for developing a Grammar Checker for Urdu. The model uses the proposed two pass parsing approach for sentence analysis. Two pass parsing approach is basically introduced to reduce the redundancy in the phrase structure grammar rules developed for sentence analysis. Initially some base Phrase Structure Grammar (PSG) Rules are used to parse the sentence. In case of failure, Movement Rules are applied and sentence is reparsed. The model checks the grammatical and structural mistakes in declarative sentences.*

Keywords: *Urdu syntax, Phrase Structure Grammar Rules, Grammar checker, Natural Language Parsing, Transformation, Computational model*

1. INTRODUCTION

This paper proposes a computational model for checking the grammatical syntactic level mistakes in Urdu declarative sentences. The research brings forward a two pass parsing approach used by the computational model.

In two pass parsing approach, a sentence is first parsed on basic PSG rules and upon failure, Movement Rules are applied to convert it to a desired correct form. After conversion the sentence is reparsed to check for errors.

The computational model first identifies grammatical and sentence structure mistakes in an Urdu input. If the sentence is erroneous, then all possible suggestions for correction are given. Grammatical error is the gender, number or case disagreement between two Parts of Speech(POS) e.g. "اچھی" and "لڑکا" have gender disagreement i.e. "اچھی" is a feminine form of adjective and "لڑکا" is a masculine form of noun. The appropriate correction for this gender mismatch is "اچھا", which is the masculine form of adjective. Syntactic error is the wrong sentence structure error e.g. "کرسی اچھی پر لڑکا بیٹھا" is structurally wrong.. The suggestion will be to correct the placement of adjective "اچھی".

The paper also discusses the architecture of a grammar checker, which includes Tokenzier, POS Tagger, Parser and Error Correction and Suggestion (ECSM) modules. All these modules are explained in detail with the help of a case study.

2. SYNTACTIC ANALYSIS OF URDU

2.1. Urdu Syntax

In an effort to check the grammaticality of a sentence, it is important to know the syntactic behavior of the language. The scope of this study is limited to declarative sentences. Declarative sentence is the normal sentence showing positive tone. Sentences showing command, request or interrogation are not catered. The order of language is also determined by the sequence of subject, object and verb (SOV) in simple declarative sentences [3].

Urdu language is comparatively more complex than some other languages. Following are some complexities encountered in doing syntactic analysis of Urdu.

2.1.1. Parts of Speech carrying gender

Urdu exhibits gender agreement e.g. the word "کتاب" "has gender 'Masculine' and number 'Singular'. On the contrary, in English, noun only contains number e.g. the same word in English, 'book' has only got the number as 'Singular'. Similarly degree, adjective, possessive, nouns, main verbs, auxiliary verbs, all agree gender property in Urdu [3].

Noun phrase (NP) is always the main constituent of the sentence. In case of gender/ number disagreement between noun phrase and other constituents, noun phrase is given more weight-age and other constituents are changed accordingly. For example if native speakers of Urdu are asked to correct the sentence "□□ اا□□بي□□لڑکی ", they'll intuitively suggest "□□ بي□□لڑکی ای□□" instead of " □□ اا□□بي□□لڑکا".

2.1.2. Word Ambiguity

Some words in Urdu are ambiguous as they can be assigned more than one part of speech e.g. "کھانا"can be used both as noun and verb.

2.1.3. Multiple structures for the same sentence

In Urdu some POS has the flexibility to occur at varied locations thus creating multiple structures for the sentence having exactly the same POS e.g. displacement of adjective " اچھا " in the sentence "□□لڑکا اچھا□□" to make the sentence "□□اچھا□□لڑکا " makes a perfectly grammatical sentence. But this shift of adjective after the noun has caused a change in the structure of sentence thus making multiple structures for the same sentence.

2.1.4. Two forms of pronoun

Pronoun has two forms which altogether changes the complexion of the sentence. The pronoun form used for the sake of respect changes the gender to plural form e.g. "تم بيﷺ جاؤ " Vs. " "آپ بيﷺ جائيﷺ".

2.1.5. Direct Object (DO) and Indirect Object (IO) movement

In Urdu it is not possible to identify swapping of locations between DO and IO, at syntactic level. One needs semantic information to check agreement in this case e.g. in case of "ﷺ پيتا دودﷺ لﷺکا", if "لﷺکا" and "دودﷺ " swap locations and become "ﷺ پيتا لﷺکا دودﷺ", it is not possible to identify that "لﷺکا" is a DO at syntactic level. Semantic information for words "لﷺکا" and "دودﷺ" is required in this case.

2.2.6 Variation with respect to Case marker

Noun in Urdu is sometimes followed by a Case marker (e.g. ﷺمي،کو،نﷺ) which acts as a connector between noun and other parts of speech. The addition of Case marker to noun adds to the variation of gender and number to noun. Hence noun form with Case marker is different from the one without it e.g. the word "لﷺکا" has the following variation.

Table 1. Different forms of word **"لﷺکا"**

Gender-Number/ Case marker	Masculine-Singular	Masculine-Plural	Feminine-Singular	Feminine-Plural
Without Case marker	لﷺکا	لﷺکﷺ	لﷺکی	لﷺکیﷺ
With Case marker	لﷺکﷺ	لﷺکوﷺ	لﷺکی	لﷺکیوﷺ

2.2.7 Varied forms of number

In Urdu, one singular form of a word might have two words mapping to its plural form e.g. the plural forms of word "کتاب" are "کتب " and "کتابيﷺ ".

2.2. Grammatical Errors

The grammatical errors to be checked for syntactic level mistakes are gender, number and case disagreement.

2.2.1. Gender Disagreement

It is the mismatch of gender between two words. Gender value can be Masculine (M), Feminine (F) and Neutral (N). Neutral gender is used in case when the particular POS is used for both masculine and feminine form e.g. the preposition "ﷺمي" is used with both M and F form of noun, hence its gender is N..

2.2.2. Number Disagreement

This disagreement occurs when the numbers of two words are not the same. Number can be Singular (S), Plural (P) or Neutral (N). Neutral is assigned to those POS which occurs both as singular and plural e.g. noun "شﷺر".

2.2.3. Case Disagreement

Case marker is a special POS which only occurs with noun as a connector. There are particular forms of a noun which can occur with Case marker, see table 1. If a noun form, which is not permissible, occurs with Case marker, case mismatch error is identified e.g. "لﷺکا نﷺ" has case disagreement, as "لﷺکا" and "نﷺ" cannot come together [3].

Based on the above information, following grammatical errors have been found in Urdu syntax.

2.2.4. Noun and Adjective Disagreement

Given below are some disagreements that occur between noun and Adjective.

Number Disagreement

Noun with Case marker: "اچﷺ لﷺکﷺ کو"
Noun without Case marker: "اچﷺ لﷺکﷺ"

Gender Disagreement

Noun with Case marker: "اچﷺ لﷺکی کو"
Noun without Case marker: "اچﷺ لﷺکی"

2.2.5. Noun and Case marker Disagreement

Given below is a disagreement that occurs between noun and Case marker.

Case Disagreement "لﷺکا کو"

2.2.6. Noun and Quantifier Disagreement

Here are some disagreements that occur between noun and Quantifier.

Number Disagreement

Noun with Case marker: "ایک لﷺکا اور لﷺکی کو"
Noun without Case marker: "دو لﷺکا"

2.2.7. Noun and Possessive Disagreement

Here are some disagreements that occur between noun and possessive.

Number Disagreement

Noun occurring before Possessive: "لﷺکاکی گاﷺی"
Noun occurring after Possessive: "لﷺکﷺ کا جوتﷺ"

Gender Disagreement

Noun occurring after Possessive: "لﷺکﷺ کا گاﷺی"

2.2.8. Adjective and Degree Disagreement

Given below are some disagreements that occur between Adjective and Degree.

Number Disagreement "اتنا اچھا لڑکا"
Gender Disagreement "اتنی اچھا لڑکا"

2.2.9. Disagreement between two Adjectives

Following are the disagreements between two adjectives.

Number Disagreement

Noun without Case marker: "اچھا اور برا لڑکا"
Noun with Case marker: "اچھا اور برا لڑکا کو"

Gender Disagreement

Noun without Case marker: "اچھا اور برا لڑکی"
Noun with Case marker: "اچھا اور برا لڑکا کو"

2.2.10. Disagreement between noun and verb

Following are the disagreements between noun and verb; verb having main and auxiliary part.

Number Disagreement

Noun with Case marker: "لڑکی نے چائے پیا ہو گا"
Noun without Case marker:"لڑکا کرسی پر بیٹھا ہو گا"

Gender Disagreement:

Noun with Case marker "لڑکی نے چائے پیا ہو گا"
Noun without Case marker "لڑکا کرسی پر بیٹھی ہو گی"

2.3. Structural Errors

Structural errors occur when the words are not in the desired sequence or there is some constituent missing [4]. There are numerous possibilities for structural errors in Urdu, some such errors are discusses below:

2.3.1. Verb Phrase(VP) missing

A declarative sentence with out a VP is incorrect. For example, **"لڑکا اچھا ایک"** has no VP.

2.3.2. Main Verb missing after Case marker

If a sentence contains a Case marker then it must also contain main verb e.g. "ایک لڑکے نے"has a Case marker but there is no main verb.

2.3.3. Misplaced Adjective Phrase (AP)

Sometimes adjectives are not occurring at the desired place. This generates a structural error where sequence of words is invalid e.g. in " لڑکا کرسی اچھی پر بیٹھا", adjective "اچھی" should be placed before "کرسی ".

2.3.4. Noun missing

In a sentence Case marker should not be followed immediately by a possessive. If such a situation occurs then it implies that there is a noun missing between Case marker and possessive. For example in the sentence, " لڑکے نے کی کرسی"a noun should be placed between Case marker "نے"and possessive "کی".

3. MODELED PHRASE STRUCTURE GRAMMAR RULES

Structure analysis of a sentence can be done using various methods. The simplest and most commonly used is the method of phrase structure analysis [1].

Phrases structure analysis results in development of Phrase Structure Grammar (PSG) Rules, which are used for parsing a sentence. Right hand side of the rules consists of one or more terminals or non-terminals but left hand side is always a non-terminal [1].

Every sentence should finally reduce to sentence (S). S should always divide into NP and IP. NP is the noun phrase while IP is the inflection phrase which contains Verb Phrase (containing main verb as head) and auxiliary verbs.

Following are some basic PSG grammar rules of Urdu syntax.

Non-Terminal rules

S	→	NP	IP
NP	→	N	
IP	→	VP	I
I	→	av	I
VP	→	NP	VP
VP	→	PP	VP
VP	→	V	

Terminal Rules

V	→	vv
N	→	nn
P	→	p
I	→	null

If the sentence "لڑکا اچھا کرسی پر بیٹھا ہے" is parsed on the above grammar, following phrasal division will be made:

S[IP[I[ہے] VP[VP[بیٹھا] PP[کرسی پر]]] NP[لڑکا اچھا]]

4. TWO PASS PARSING

4.1. Why Two Pass Parsing?

It is impossible to reckon the number of sentences in any language by simply listing the sentence structure of each and every one of them. Some generalization is needed to cover the syntax of a language in reasonable depth. The major problem with phrase structure analysis is that it doesn't give any generalization. Even two sentences with exactly the same constituents but a little varied structure is analyzed using separate rules (as mentioned above in 2.2.3).

Hence generalization in phrase structure analysis is achieved by introducing movement approach. Movement helps in reducing the number of phrase structure rules needed to represent the same sentence with a different sentence structure. These additions of movements

introduce a second pass, in which the restructured sentence is parsed again.

Another way to achieve generalization is transformation approach. But unlike phrase structure analysis, it is computationally more complex. Besides this, transformation approach requires semantic information but our research is basically done at syntactic level. Movement approach helps to reduce grammar rules just like transformational approach without going into semantic details of the sentence.

4.2. Generalization through movement approach

Movement approach is introduced to add generalization to phrase structure analysis. Grammar rules have been made for the assumed base sentence structures and all other sentences are derived from the base structure through movement. Hence if a sentence is not in its base form, the parsing in first pass, which is done on base structure grammar rules, fails. Then the movement rules are applied to restructure the sentence into its base form for parsing in second pass. If none of the transformation rules are applicable to the structure, then this means that the sentence structure is wrong and hence it is an erroneous entry [2].

PSG rules actually form the base structure. Hence if any sentence digresses from this base structure, it is not parsed. As shown in the PSG rules, parsing is successful only when the constituent finally ends up in sentence (S). S can be formed only if the eventual constituents left to be reduced are NP and IP.

This leads to applying movements to convert the sentence to its base form. Following are the two movements identified till yet.

4.3. PP Movement

If the final constituents left to be reduced include PP as the first constituent, then the sentence is never reduced to S. Hence movement is required. The movement rule replaces PP with the first NP it encounters in the sentence at the same level. If the following sentence is parsed on base structure PSG rules, parsing initially fails and PP movements are applied to the sentence.

[پا ہا ہا] [پانی] [میز پر] [گلاس میں]

IP NP PP PP

Transformed sentence given below is reparsed on base structure PSG rules.

[پا ہا ہا] [گلاس میں] [میز پر] [پانی]

IP PP PP NP

4.4. AP Movement

AP movement is applied when the parsing fails because an AP constituent is eventually left to be reduced with no NP following it. The following sentence is not initially parsed. AP movement is done to transform it into base structure and then parsed again in second pass.

[ہا ہا] [اچھا ہا] [کالا]

IP AP NP

The transformed sentence given below has NP preceded by AP.

[ہا ہا] [کالا] [اچھا ہا]

IP NP AP

5. GRAMMAR CHECKER

5.1. Architectural Diagram

The computational model for Grammar Checker consists of four major modules, as shown in Figure 1. A brief description of each one of these modules is given below [5].

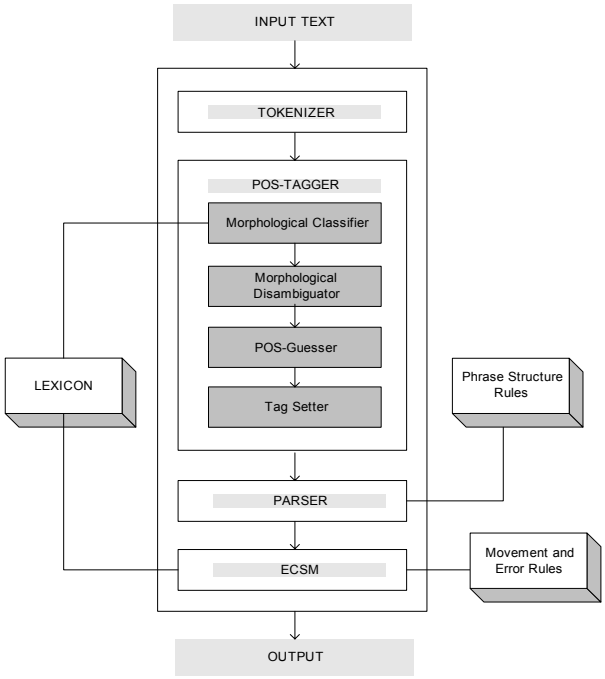

Figure 1. Architecture Diagram of Grammar Checker

5.1.1. Tokenizer

This module identifies the word and sentence boundaries. It takes in the input sentence and then generates tokens accordingly. Only word tokens generated by Tokenizer are then passed on to POS-Tagger.

5.1.2. POS-Tagger

POS-Tagger assigns part of speech tags (POS-Tags) to words, reflecting their syntactic category. It consults lexicon and gathers morpho-syntactic information for each tokeznied word. Then based on the gathered morpho-syntactic information it assigns appropriate tags

to each token. POS-Tagger consists of following sub-modules:

5.1.2.1. Morphological Classifier

It classifies word-tokens with sets of morpho-syntactic features. This is implemented by lexicon lookup. Words are listed in the lexicon with their morpho-syntactic features and the lookup retrieves all possible readings for a given word. The morphological classifier retrieves for a word its possible POS and other related morpho-syntactic features such as number, case, gender, etc. For example the information returned against the word "کتاب " is shown in table 2.

Table 2. Information returned by Classifier

Lexeme	کتاب
Root	کتاب
Affix	null
Part of Speech	noun
Gender	Masculine
Number	Singular

There is a possibility that there are words in the input text that have more than one set of morpho-syntactic information attached to them. This creates serious ambiguity which is resolved by passing such words to the Morphological Disambiguator.

5.1.2.2. Morphological Disambiguator

The core functionality of this module is to remove the ambiguities that arise when there is more than one set of morpho-syntactic information retrieved for a word e.g. the word can be used both as a noun and a verb. In such cases the Morphological Disambiguator selects the best and the most probable morpho-syntactic information of input word. For example, the tokenized word passed on to the Morphological Classifier was "گا ". In Urdu the word "گا" is used both as a noun (for cow) and as a verb (for singing), so Morphological Classifier will return two sets of morpho-syntactic information. Now it is responsibility of Morphological Disambiguator to select one of them keeping in view the context of the input sentence. The other morpho-syntactic information is discarded.

This removal of ambiguity can be done by using numerous approaches e.g. Rule based system, Neural Networks, Stochastic and probabilistic techniques etc.

5.1.2.3. POS Guesser

POS (Part of Speech) Guesser, as it is obvious from the name, is used to guess a POS for words that are not known to the lexicon. This phase will be skipped if there are no unknown words in the tokenized text. Sometimes there are proper nouns in the input text which are usually not stored in lexicon. In case of such scenarios it is the responsibility of the POS Guesser to try and find out the POS. For example, in sentence " اسلم بی " lexicon will return morpho-syntactic information for " بی " and "", but the word "اسلم" is unknown to the lexicon as it is name of a person. Here POS Guesser can guess the POS for "اسلم" by analyzing the words occurring before

and after it. In above example it might consider that verb "بی" needs a subject, which is necessarily a noun. So it will assign "noun" as POS to "اسلم". More sophisticated Guesser can also make decisions even about gender and number quite accurately.

POS Guesser now passes on the tokenized text, along with the morpho-syntactic information attached with each token, to Tag Setter.

5.1.2.4. Tag Setter

After removing, maximum possible ambiguities and guessing unknown words the tokenized text along with their morpho-syntactic information is passed to Tag Setter. Each tokenized word is now assigned an appropriate POS-tag based on its morpho-syntactic information.

For example, input text was:

"ل کا گانا گا گا".

The tagged text will be:

گا/av گا/mv گانا/nn کا ل/nn

5.1.3. Parser

POS-Tags assigned by the pervious modules will now be fed to the parser as terminal nodes for the parse tree. This module parses the POS-tags on the PSG rules loaded from the file. It will generate a correct parse tree if there were no structural errors in the input sentence. Grammatical errors cannot be checked using the grammar alone. For identifying the grammatical error, control is given to the next module which is Error Correction and Suggestion Module (ECSM). In case of a unsuccessful parsing control is also passed to ECSM to carry out movements or to suggest correction.

5.1.4. Error Checking and Suggestion Module

The Error Checking and Suggestion Module (ECSM) works side by side with the Parser. The basic purpose of this module is to check errors and give suggestions for corrections. Parser passes control to this module in three scenarios:

1. *Checking Grammatical Error:* In this scenario parser passes the control whenever a phrase structure rule is fired that also demands firing of an agreement error rule. Error rules are fired in order to check agreements between two words/phrases. If an agreement error rule is fired successfully then this means that there exists a disagreement. In such cases the ECSM consults the lexicon to suggest a correction for the error. For example the sentence is "ل کا بی ای". Here there is a gender disagreement between masculine noun (subject) "ل کا" and feminine verb "بی ای". The ECSM will consult the lexicon and will look

for a masculine verb having the same root as "بی□□ی". The suggested correction from the lexicon will be the verb "بی□□ا".

2. *Movements for Second Pass:* In this scenario control is passed to ECSM by the parser when parsing fails on a specific sentence and movement rules can be applied on it. After applying the movement rules the transformed sentence (base form sentence) is again fed to the parser for second pass. For example the input sentence "بی□□ا کا□ل پر کرسی" fails to parse on the first pass. Then the Prepositional Phrase (PP) movement rule is applied to it and the sentence will be transformed into "بی□□ا کا□ل پر کرسی" and parsed again.

3. *Structural Error:* In this scenario control is passed whenever there is a structural error in a sentence and no movement rules could be applied. This indicates that the sentence was structurally incorrect. For example "بی□□ا تا□ کا□ل" has structural error. So the ECSM will give "Auxiliary Verb not in Place" as suggestion to correct this structural error.

6. CASE STUDY

In this section we will take an example sentence and will pass it through all the phases of Grammar Checker.

Input Sentence:

"□□□ اتی کا□انا اسلم می□ کمر□"

6.1. Tokenizer

The input sentence will be tokenized and each word will be uniquely identified. In all, Tokenizer will generate seven tokens. Six tokens will be for words and one token for sentence delimiter.

6.2. POS-Tagger

The tokenized text will be passed to the POS-Tagger which will assign POS-Tags depending upon the morpho-syntactic information for each word.

6.2.1. Morphological Classifier

Morphological classifier will pass the tokenized text to Lexicon, in order to get morpho-syntactic information for each word. Result set for each word will be as under:

Table 3. Information returned for word "کمر□"

Lexeme	کمر□
Root	کمر
Affix	□
Part of Speech	Noun
Gender	Masculine
Number	Singular

Table 4. Information returned for word "می□"

Lexeme	می□
Root	می□
Affix	Null
Part of Speech	Preposition

No morpho-syntactic information is available for "اسلم", this is an unknown word.

Two sets of morpho-syntactic information will be available for "کا□انا", as it is both a noun and a verb. The sets returned will be:

Table 5. Information returned for word "کا□انا" as Verb

Lexeme	کا□انا
Root	کا□
Affix	نا
Part of Speech	Main Verb
Gender	Masculine
Number	Singular

Table 6. Information returned for word "کا□انا" as Noun

Lexeme	کا□انا
Root	کا□
Affix	نا
Part of Speech	Noun
Gender	Masculine
Number	Singular

Table 7. Information returned for word "کا□اتی"

Lexeme	کا□اتی
Root	کا□
Affix	تی
Part of Speech	Main Verb
Gender	Feminine
Number	Neutral

Table 8 . Information returned for word "□□"

Lexeme	□□
Root	□□
Affix	Null
Part of Speech	Auxiliary Verb
Gender	Neutral
Number	Singular

6.2.2. Morphological Disambiguator

The information extracted by Morphological Classifier is ambiguous as there is more than one set of morpho-syntactic information for the word "کا□انا", one as a noun and other as a verb. Morphological Disambiguator shall now select one set of morpho-syntactic information based on the context in which the word is used.

Morphological Disambiguator shall retain "کھانا" as a noun because it is immediately followed by a main verb "کھاتی", and in Urdu no two main verbs can occur simultaneously. The morpho-syntactic information of "کھانا" as verb will be discarded.

6.2.3. POS Guesser

Morphological Classifier was unable to find the morpho-syntactic information for Word "اسلم", as it does not exist in lexicon.

It is the task of POS Guesser to make a guess about the Part-of-Speech for unknown word "اسلم". POS-Guesser will mark it as a noun (masculine, singular) because of the context in which the word "اسلم" occurred i.e. it appeared after a preposition and before a noun.

6.2.4. Tag Setter

The input text will be finally tagged using the morpho-syntactic information available for each word after removal of ambiguities and unknown words. The tagged text will be as under:

کمر /nn میں /pp اسلم/nn کھانا /nn کھاتی/mv /av

These tags will now be fed to parser for parsing.

6.3. Parser

The Tags passed by POS-Tagger will be treated as terminal nodes for the parser. The input for parser will be "nn pp nn nn mv av". PSG rules that shall be used for parsing are explained in section 3.

6.3.1. Sequence of Rules for Input Sentence

The sequence in which the sentence will be parsed is as under (bottom-up parsing is being done over here):

nn pp nn nn mv av→N pp N N mv av→NP pp NP NP mv av→NP P NP NP mv av→PP NP NP mv av →PP NP NP V av→PP NP NP VP av → PP NP VP av→PP VP av→VP av-> VP I→ IP→ **ERROR**

The Parse structure generated for this input will be:

[[[کمر]NP میں]PP [[اسلم]NP [[کھانا]NP

[[ ]VP [VP[VP[VP[کھاتی]IP]I]S

Parsing fails here, as we have no Phrase Structure rule that reduces {IP}. This Error is passed on to the ECSM for further processing.

6.4. Error Checking and Suggestion Module

ECSM targets such problems i.e. structural errors, by first applying movement rules on parse structure, and in case no movement rules are applied, it fires structural error rules to find structural errors.

For the above case, movement rule for PP is fired successfully i.e. swap the places of PP and NP. So "اسلم" and "کمر میں" are swapped. The new transformed input text for second pass is now:

"کمر میں اسلم کھانا کھاتی "

6.5. Second Pass

As tagging has already been done, so the new sequence of tags is now passed on to parser.

6.5.1. Parser (2nd Pass)

The new sequence of terminals for parser is "nn nn pp nn mv av".

nn nn pp nn mv av→N N pp N mv av→NP NP pp NP mv av→NP NP P NP mv av→ NP PP NP mv av →NP PP NP V av→NP PP NP VP av → NP PP VP av →NP VP av-> NP VP I→ NP IP→ **S**

The parse structure generated for this input will be as under:

[[[کمر]NP میں]PP [[اسلم]NP [[کھانا]NP

[[ ]VP [VP[VP[VP[کھاتی]IP]I]S

The control will be passed to ECSM because a PSG rule (S→NP IP) requires firing of an agreement error rule for checking grammatical error.

6.5.2. Error Checking and Suggestion Module (ECSM) (2nd Pass)

ECSM will fire agreement error rules associated with:

S → NP IP

6.5.2.1. An Agreement Error Rule

The rule is supposed to check the agreement between NP and IP, in fact the noun (subject) with the main and auxiliary verbs. Here we see that noun (subject) "اسلم" is singular and masculine, where as main and auxiliary verb "کھاتی " as a whole is singular and feminine [5]. The steps followed will be as under:

1. *Number Agreement:* Compare the number of both noun and verbs. Here we see that there is no number disagreement because both are singular.

2. *Gender Agreement:* Compare the genders. Here we see that there is gender disagreement, noun is masculine where as main verb is feminine.

6.5.2.2. Suggestion for Correction

ECSM will consult lexicon in order to find a correction for the above gender disagreement. The decision for suggestion will be based on intuition of an Urdu native speaker, as a native speaker will intuitively find a correction for verb instead of a noun.

The query for lexicon will be to find a main verb agreeing with the noun. Hence query will be [Root = کھا

, Number = Singular, Gender = Masculine]. As a result lexicon will return "کاتا" as suggested main verb, to replace "کاتی".

6.5.2.3. Final Output

To show the final output the movement rules applied on the sentence will be reversed (as they were applied only to parse successfully). The final suggested sentence will be:

"کمرہ میں اسلم کانا کاتا □□□"

7. CONCLUSIONS

Two-pass parsing implementation is a new and unique method to solve complex parsing problems. It allows you to keep the computational Grammar simpler, which at the same time covers maximum range of sentences. It gives you a flavor and functionality of transformations without actually going into the details of Transformational Grammar.

We have successfully implemented this model (excluding the modules of Morphological Disambiguator and POS Guesser). The implemented system is capable of taking a declarative Urdu sentence as input to check its grammaticality, if errors are found it displays suggested corrections for the erroneous sentence (GUI given in Appendix A.2).

This implemented system proves the validity of the two pass parsing approach and the proposed computational model.

ACKNOWLEGEMENTS

Urdu Grammar Checker (UGC) group is really thankful to

- Dr. Anjum.P.Saleemi, for helping us in understanding the complexities of Syntax, especially syntactic aspects of Urdu.
- Mr. Shafiq-ur-Rehman, for helping us out in the computational issues of Natural Language Processing.

REFERENCES

[1] Donna Jo Napoli, *Linguistics An Introduction*, Oxford University Press, 1996.

[2] Liliane Haegeman, *Introduction to Government and Binding Theory*, 2nd edition, BlackWell (Oxford UK and Cambridge USA).

[3] John.T.Platts, *A Grammar of Hindustani or Urdu Language,* Fifth Impression, pages 21-43 and 223-370, London Crosby Lockwood and Son (7 Stationers' Hall Court, Ludgate Hill) 1909.

[4] Looking for Errors: A Declarative Formalism for Resource-Adaptive Language Checking by Andrew Bredenkamp, Berthhold Crysman, Mirela Petera. Pg. 667-673, Proceedings of 2nd International Conference for Language Resources and Evaluation, Athens-Greece, 31st May-2nd June 2000. (FOR ARCH DIAG)

[5] Johan Carlberger, Rickard Domeij, Viggo Kann, Ola Knutsson. "A Swedish Grammar Checker". http://www.amt.nada.kth.se/theory/projects/granska/rapporter/compling20000419.pdf

APPENDIX

A.1 Abbreviation used in the paper

Word	Abbreviation
Sentence	S
Subject Object Verb	SOV
Part Of Speech	POS
Inflectional Phrase	IP
Noun Phrase	NP
Verb Phrase	VP
Preposition Phrase	PP
Adjectival Phrase	AP
Terminal	T
Non terminal	NT
Verb	V
Noun	N (NT), nn (T)
Preposition	P (NT), pp (T)
Auxiliary Verb	av
Main Verb	mv
Masculine	M
Feminine	F
Singular	S
Plural	P
Neutral	N

A.2 GUI of implemented Urdu Grammar Checker