AN APPROACH OF CHECKING GRAMMAR FOR TELUGU LANGUAGE COMPLEX SENTENCES

Dr. V Suresh

Associate Professor, Department of Information Technology, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India

ABSTRACT

For grammar checking of complex sentences, it is necessary to identify the structure of these sentences. The structure of complex sentences can be identified on the basis of number of clauses and types of clauses present in them. If a sentence contains a dependent clause along with independent clause then it is a complex sentence. Once the sentence is identified as complex sentence, the next step is to identify its pattern. After identification of patterns, various clauses present in the sentence are extracted and grammar checking is performed on them. A grammar checking system for complex sentences of Telugu language has been done with grammatical error detection and correction. This research work on grammar checking of complex sentences is based on the assumption that the input sentences will be in Telugu script.

KEYWORDS: Dependent Clause, Divide and Conquer, Adverb Clause, Noun Phrase, Verb Phrase

1. INTRODUCTION

Telugu language belongs to Indo-Aryan family of languages (Dravidian languages). Dravidian languages are Telugu, Tamil, Kannada, Malayalam- Other members that belong to this family are Kannada, Tamil, Malayalam, Hindi, Bengali, Gujarati, and Marathi etc. Telugu is spoken in India,Canada, USA, UK, and other countries with Telugu immigrants. Telugu language is the 8th most widely spoken language in the world, 4th most spoken language in Canada(The Times of India, 14th February, 2008) and the 9th in India with more than 45 million speakers. It is the official language of Telugu states (Andhra Pradesh and Telangana). The first treatise on Telugu grammar, the "Andhra Shabda Chintamani" was written in Sanskrit by Nannaya who was considered as the first poet and translator of Telugu in the 11th century A.D. There was no grammatical work in Telugu prior to Nannayya's "Andhra shabda chintamani". This grammar followed the patterns which existed in grammatical treatises like Astadhvavi and Valmiki vyakaranam but unlike Paninni.Nannayya divided his work into five chapters,covering Samjna, Sandhi, Ajanta, Halanta and Kriya. After Nannayya, Atharvana and Ahobala composed Sutras, Vartika and,Bhashyam. Like Nannayya, they had previously written their works in Sanskrit.

This paper work on grammar checking of compound and complex sentences is based on the assumption that the input sentences will be in Telugu script in Unicode. Thus, the examples given in this work are in Telugu script in Unicode,along with their transliteration in Roman script and translation in English. For inline examples, transliteration will be provided in parentheses and translated text in single quotes. e.g. ఆంధ్రావిశ్వకళాపరిషత్ (andhraviswakalaparishad) 'Andhra University'

2. A GRAMMAR CHECKER

The fundamental task of the grammar checker is to check the internal and external structure of the sentence to detect the grammatical errors and to give a suggestion to rectify these errors. Grammar checking is one of the widely used applications in the field of Natural Language Processing (NLP). A Grammar checker for simple, compound and complex sentences of a language is a system that checks various structural and grammatical errors in a given text based on the available possible patterns of simple, compound and complex sentences and grammatical rules of that particular language, and reports errors. It is something different though that even many of those first language writers will find it hard to list explicitly the grammatical rules involving their writings.

2.1 Grammar Checking of Complex Sentences

Compound sentences are composed of at least two independent clauses joined by coordinate conjunctions, comma or semicolon. For grammar checking of compound sentences, each clause is extracted from the sentence and grammar checking is performed on it. Since there may be two to any number of independent clauses present in compound sentences, therefore, divide and

conquer model can be used for grammar checking of compound sentences. In accordance with divide and conquer, the compound sentence is simplified by splitting it into individual clauses and then each clause undergoes error detection and correction mechanism. In this way, overall grammar checking process for compound sentences takes place in two steps; first step is extracting the independent clauses from the compound sentence and second step is to perform grammar checking on each extracted clause. For extracting the independent clauses from the compound sentence, the clause boundary of these clauses has to be identified.

2.2. Grammar Checking of Complex Sentences
Complex sentences are broadly classified into four categories as discussed
- Complex sentences containing non-finite clause.
- Complex sentences containing relative clause.
- Complex sentences containing adverb clause.
- Complex sentences containing Key Indicator (KI) clause.

In the above mentioned categories, the first one is predicate bound type and next three are non-predicate bound type complex sentences.

2.2.1. Grammar Checking of Complex Sentence Containing Non-Finite Clauses
This type of complex sentence is identified by the presence of non-finite clause which contains a non-finite verb. As discussed in this type of structure, the dependent and independent clauses can be marked using morphological features. In the following section, the procedure of error checking performed in this type of complex sentence is explained.

2.2.1.1. Participle sentences
These sentences contain dependent and independent clauses. The dependent clause has a subject and an incomplete action or condition performed on this subject in the form of a non-finite verb and the independent clause explains the consequence of the action performed on the subject in the dependent clause. Since the dependent clause has only subject and non-finite verb, and non-finite verb does not inflect for any grammatical category, there is no need to check the grammar of the dependent clause.

The independent clause, on the other hand, contains a subject and a predicate and hence, the following agreement must be checked:
- Modifier and noun agreement
- Noun and adjective agreement
- Subject and verb agreement
- Order of modifier of noun in noun phrase
- Order of words in a verb phrase

This sentence is further categorized into following two categories.

2.2.1.1.1. Sentence having separate subject of Dependent and Independent Clauses
In such sentences, both dependent and independent clauses have their own subject

Consider the following example:
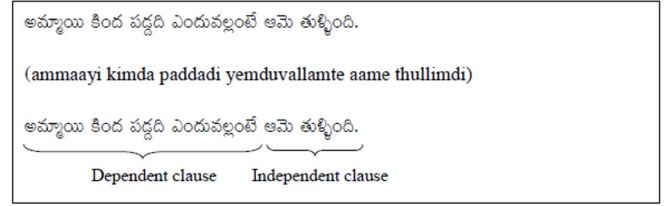
అమ్మాయి (ammaayi) is the subject of dependent clause అమ్మాయి కింద పద్దది (ammaayi kimda paddadi) and ఆమె (aame) is the subject of independent clause ఆమె తుళ్ళింది (aame thullimdi) shown in the above sentence. In this type of sentence, all the components of independent clause must grammatically agree to one another. Therefore, all the grammatical agreements in the independent clause must be checked with respect to its own subject and not with the subject of the dependent clause.

2.2.1.1.2. Sentence having common subject between dependent and Independent Clauses
The sentences falling in this category share a common subject between dependent and dependent clauses. Consider the example:
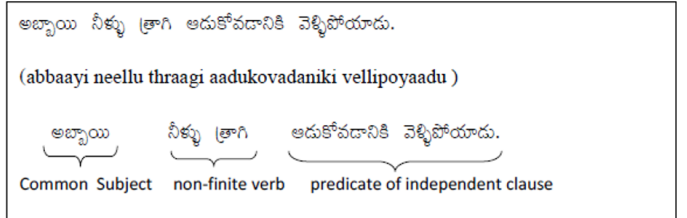
అబ్బాయి (abbaayi) is the common subject shared by both clauses- one is dependent clause i.e. నీళ్ళు త్రాగి (neellu thraagi) and second is independent clause i.e. ఆదుకోవడానికి వెళ్ళిపోయాడు (aadukovadaaniki vellipoyaadu) as shown in the above example. Therefore, dependent and independent clauses generated from the above complex sentence are:

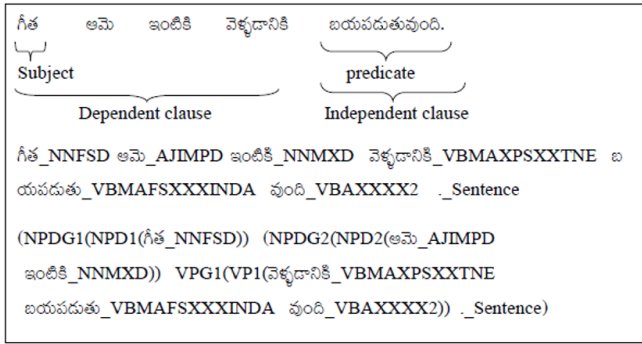**Dependent clause:** అబ్బాయి నీళ్ళు త్రాగి (abbaayi neellu thraagi)

Independent clause: అబ్బాయి ఆదుకోవడానికి వెళ్ళిపోయాడు (abbaayi aadukovadaaniki vellipoyaadu)

Therefore, the predicate of independent clause i.e.ఆదుకోవడానికి వెళ్ళిపోయాడు (aadukovadaaniki vellipoyaadu) must be grammatically in agreement with this common subject i.e. అబ్బాయి (abbaayi).

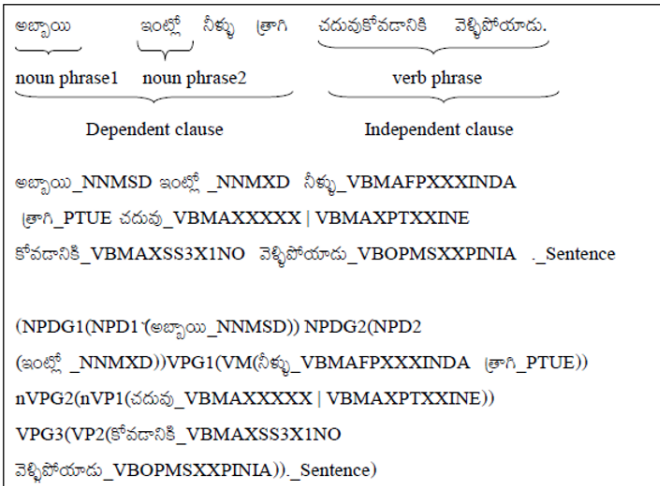**Rules to share the common subject :**
1. If the sentence will share the subject of a dependent clause with an independent clause, otherwise the predicate of the independent clause has a subject i.e. it starts with a noun phrase, then this sentence does not have a common subject.

Consider the following example:

గీత   ఆమె   ఇంటికి   వెళ్ళదానికి   బయపడుతువుంది.
Subject   predicate
Dependent clause   Independent clause

గీత_NNFSD ఆమె_AJIMPD ఇంటికి_NNMXD వెళ్ళదానికి_VBMAXPSXXTNE బ యపడుతు_VBMAFSXXXINDA వుంది_VBAXXXX2 ._Sentence

(NPDG1(NPD1(గీత_NNFSD)) (NPDG2(NPD2(ఆమె_AJIMPD ఇంటికి_NNMXD)) VPG1(VP1(వెళ్ళదానికి_VBMAXPSXXTNE బయపడుతు_VBMAFSXXXINDA వుంది_VBAXXXX2)) ._Sentence)

The predicate of independent clause does not contain a noun phrase and hence, the subject గీత (geetha) of dependent clause will be used as common subject as shown in the above example.

2. If the first noun phrase will act as the subject of independent clause then dependent clause contains two consecutive noun phrases. Consider the following example:

అబ్బాయి   ఇంట్లో నీళ్ళు తాగి చదువుకోవదానికి   వెళ్ళిపోయాదు.
noun phrase1 noun phrase2   verb phrase
Dependent clause   Independent clause

అబ్బాయి_NNMSD ఇంట్లో _NNMXD నీళ్ళు_VBMAFPXXXINDA తాగి_PTUE చదువు_VBMAXXXXX | VBMAXPTXXINE కోవదానికి_VBMAXSS3X1NO వెళ్ళిపోయాదు_VBOPMSXXPINIA ._Sentence

(NPDG1(NPD1`(అబ్బాయి_NNMSD)) NPDG2(NPD2 (ఇంట్లో _NNMXD))VPG1(VM(నీళ్ళు_VBMAFPXXXINDA తాగి_PTUE)) nVPG2(nVP1(చదువు_VBMAXXXXX | VBMAXPTXXINE)) VPG3(VP2(కోవదానికి_VBMAXSS3X1NO వెళ్ళిపోయాదు_VBOPMSXXPINIA)) ._Sentence)

A dependent clause contains two noun phrases; అబ్బాయి (abbaayi) and ఇంట్లో (imtlo) shown in the above sentence. Therefore, first noun phrase i.e. అబ్బాయి (abbaayi) will be the common subject.

The algorithm used for grammar checking of participial type complex sentence is as follows :
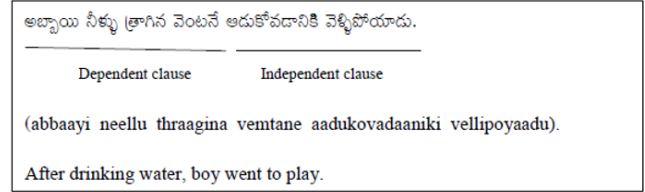
Databases used: Grammar checking options
Input: Tokenized and annotated Telugu sentence
Output: Updated Tokenized and annotated Telugu sentence

1. Using the information of clause boundary, separate the dependent and independent clauses and store them in separate tables.
2. If independent clause contains a subject then go to step 4 else go to step 3.
3. Insert the subject of dependent clause in independent clause
4. Get all the grammar checking options, having the onoff value set to 1, from the Grammar checking options database sorted by the Priority field.
5. Repeat steps 6 to 8 for dependent and independent clause separately.
6. Repeat steps 7 and 8 for all the grammar checking options.

7. Call the respective method to perform the required check on the current clause.
8. Rejoin the dependent and independent clause to construct the sentence again.
9. Output the rectified sentence.

Consider the following example:

అబ్బాయి నీళ్ళు తాగిన వెంటనే ఆదుకోవదానికి వెళ్ళిపోయాదు.
Dependent clause   Independent clause

(abbaayi neellu thraagina vemtane aadukovadaaniki vellipoyaadu).

After drinking water, boy went to play.

The independent clause ఆదుకోవదానికి వెళ్ళిపోయాదు (aadukovadaaniki vellipoyaadu) along with subject of dependent clause అబ్బాయి (abbaayi) will go for error detection and correction shown in the above sentence. The steps followed in grammar checking of this sentence are shown in figure 2.1.
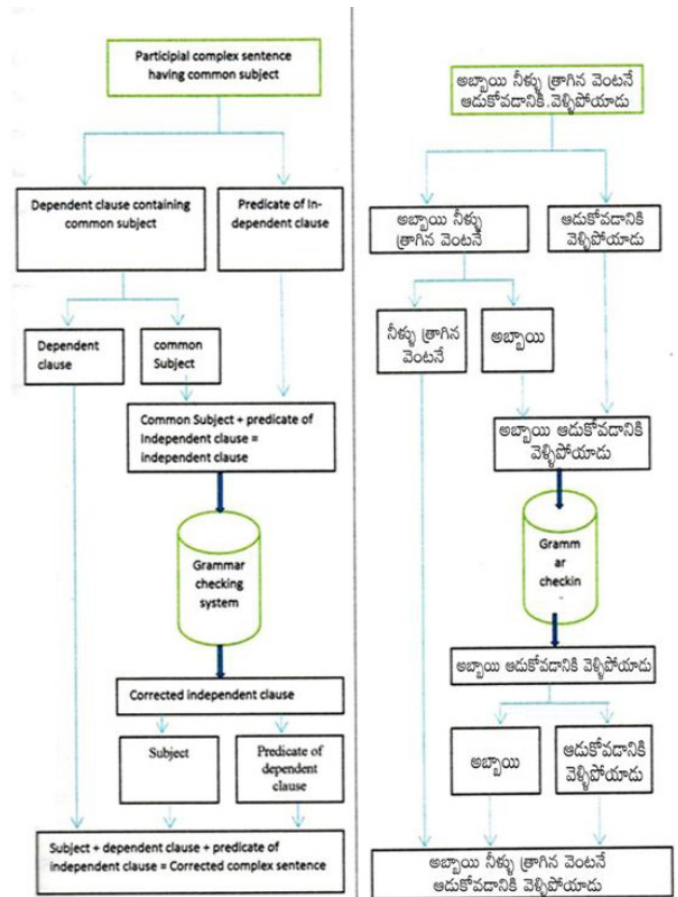

Figure 2.1 : Architecture of Grammar Checker for Participial Type System

As shown in the above figure, first the sentence is split down into two parts at the non-finite verb; one part will contain the common subject of independent clause along with the dependent clause and the second part contains the predicate of independent clause. In the next step, the common subject is copied from the first part and joined to the predicate of the independent clause in the second part. This results in the generation of an independent clause. This independent clause is then checked for different

types of grammatical errors as mentioned in Table 2.1.

2.2.1.2. Grammar Checking of Complex Sentences containing Infinitival Clause

In these types of sentences, a common subject is shared by both dependent and independent clauses. A complex sentence containing infinitival clause can be marked for dependent and independent clauses as discussed in chapter 5. Therefore, this sentence can be simplified by splitting it into dependent and independent clauses by using clause boundary information. Consider the following example:

రవి దొంగతనం చేసాడు కాబట్టి భయపడ్డాడు.

(ravi domgathanam chesaadu kaabatti bhayapaddaadu)

Ravi was afraid because he committed the theft.

After marking the clause boundaries, it is clear that this sentence is composed of two clauses; dependent clause దొంగతనం చేసాడు కాబట్టి (domgathanam chesaadu kabatti) and independent clause భయపడ్డాడు (bhayapaddaadu) with a common subject రవి (ravi) shared by both the clauses. These separate parts of the sentence i.e. common subject, dependent clause and independent clause have been shown below:

రవి	దొంగతనం	చేసాడు	కాబట్టి	భయపడ్డాడు
Common subject		Dependent clause		Independent clause
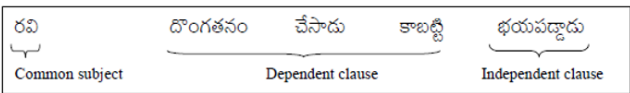
After simplification and sharing the common subject, two clauses of the sentence are:

Dependent clause: రవి దొంగతనం చేసాడు కాబట్టి (ravi domgathanam chesaadu kaabatti),

Independent clause: రవి భయపడ్డాడు (ravi bhayapaddaadu).

The dependent clause does not contain predicate and hence, various agreement related errors checking cannot be performed as discussed earlier. It is the independent clause that participates in the error checking.

Algorithm used :

The algorithm used for grammar checking of infinitival type complex sentence is as follows:

Databases used: Grammar checking options
Input: Tokenized and annotated Telugu sentence
Output: Updated tokenized and annotated Telugu sentence

1. Using the clause boundary mark, simplify the complex sentence by splitting it into dependent and independent clauses and store them in separate databases.
2. Copy the subject of the dependent clause in an independent clause.
3. Get all the grammar checking options, having the onoff value set to 1, from the grammar checking options database sorted by the Priority field.
4. Repeat steps 5 to 6 for dependent and independent clauses separately.
5. Repeat steps 6 for all the grammar checking options.
6. Call the respective method to perform the required check on the current clause.
7. Rejoin the dependent and independent clauses to construct the sentence again.

8. Output the rectified sentence

Consider the following example:

రవి దొంగతనం చేసాడు కాబట్టి భయపడ్డాడు.

(ravi domgathanam chesaadu kabatti bhayapaddaadu)

Ravi was afraid because he committed the theft.

The architecture of the system along with the step involved in grammar checking of above sentence is shown in figure 2.3. As shown in the figure 2.3, the input sentence is divided into dependent and independent clauses on the basis of the clause boundary information. After that the subject of dependent clause i.e. రవి (ravi) is combined with the independent clause i.e. భయపడ్డాడు (bhayapaddaadu) and fed to the grammar checker module. After checking grammar, this subject again combined with dependent clause to re-construct the sentence.
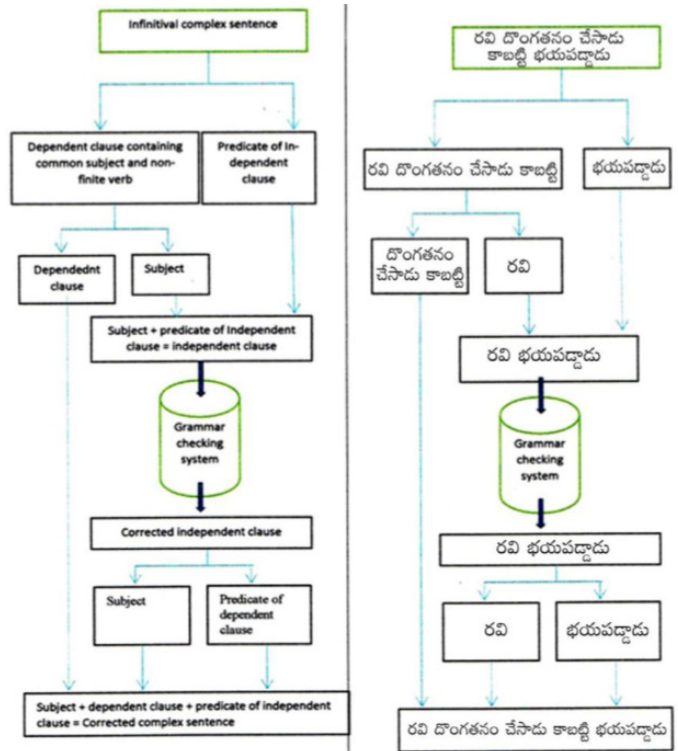

Fig 2.3 : Architecture and working of grammar checker for infinitival type system

2.2.2. Grammar Checking of Complex Sentences Having Sequential Non- Predicate Bound Clauses (Ki Clause)

These types of complex sentences have dependent clauses having ఏమిటంటే (emitamte) conjunction ఏమిటంటే (emitamte) as an essential component. ఏమిటంటే (emitamte) conjunction is used to mark the clause boundary between dependent and independent clauses as discussed in chapter 5 under section 5.4.2.1. If this ఏమిటంటే (emitamte) conjunction is removed from the dependent clause, then this dependent clause behaves like independent clause and the agreement between its various components can be checked. Consider the following example:

అతనికి తెలిసింది ఏమిటంటే ఆ అడవుల్లో ఎక్కువవగా దోమలు వున్నాయని.

(athaniki thelisindi emitamte aa adavullo yekkuvaga domalu unnayani)

He knew that there are a lot of mosquitoes in forests.

After splitting the above sentence into classes, following are obtained :

అతనికి తెలిసింది   ఏమిటంటే ఆ అడవుల్లో ఎక్కువవగా దోమలు వున్నాయని

Independent clause   Dependent clause

After simplification of the above complex sentence, the following clauses are obtained:

Dependent clause: ఏమిటంటే ఆ అడవుల్లో ఎక్కువవగా దోమలు వున్నాయని. (yemitamte aa adavullo yekkuvaga domalu unnayani)

Independent clause: అతనికి తెలిసింది (athaniki thelisindi)

Now if the ఏమిటంటే (emitamte) conjunction is removed from the dependent clause then this dependent clause will become an independent clause i.e. దోమలు వున్నాయని (domalu unnaayani).

The algorithm used for grammar checking of sequential type complex sentence is as follow :
Databases used: Grammar Checking options
Input: Tokenized and annotated Telugu sentence
Output: Updated tokenized and annotated Telugu sentence

1. Using the information of clause boundary, separate the dependent and independent clauses and store them in separate tables.
2. Elinimate the ఏమిటంటే (Emitamte) conjunction from the dependent clause the convert it into independent clause.
3. Get all the grammar checking options having the *OnOff* value set to 1, from the grammar checking options database sorted by the Priority field.
4. Repeat steps 5 to 6 for both independent clauses.
5. Repeat steps 6 for all the grammar checking options.
6. Call the respective method to perform the required check on the current clause.
7. Rejoin both the independent clause using ఏమిటంటే (Emitamte) conjunction to construct a complex sentence again.
8. Output the rectified complex sentence.

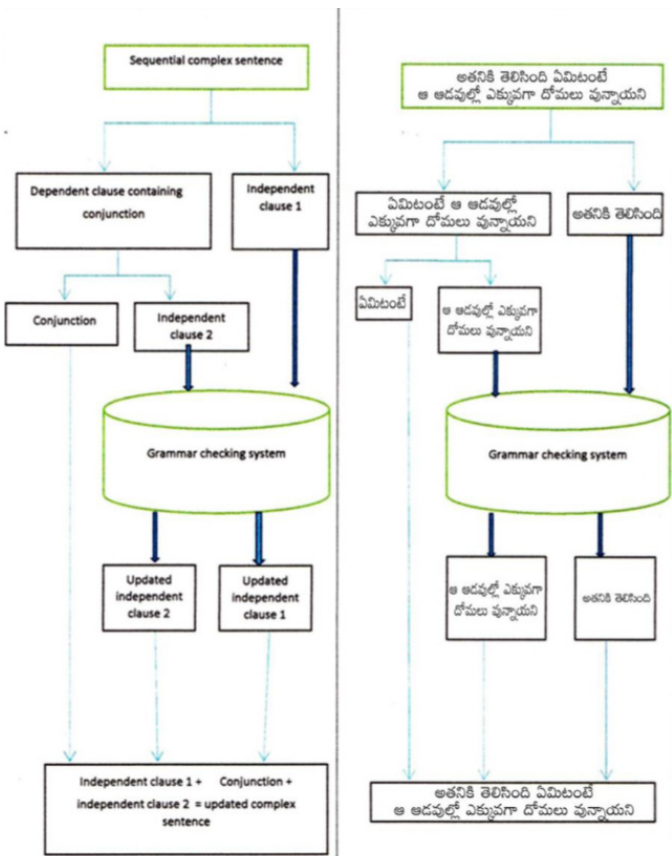The steps followed in grammar checking of this sentence are shown in figure 2.4.

Fig 2.4 : Architecture of grammar checker for sequential type system

2.2.3. Grammar Checking of Complex Sentences Having Nonrelative (Conditional) Type Non-Predicate Bound Clauses (Adverb Clause) :

Non-relative (conditional) clauses have generally an independent clause followed by a dependent clause in a complex sentence. The dependent clause in such sentences starts either with అ (a)-Adverb or తే (te)-Adverb and the conjunction used may be correlated as discussed in chapter 3 under section 3.4.3.2.3. The dependent clause can be converted to an independent clause by removing the adverb component i.e. అ (a)--Adverb or తే (te)-Adverb. Except for some exceptional cases, there is no grammatical agreement between dependent and independent clauses. Consider the following example:

అతను బ్రతికుంటే , ఇచ్చితంగా గొప్ప మనిషి అయ్యుందేవాదు.

(athanu brathikumte, khachithamgaa goppaa manishi ayyumdevaadu).

If he had been alive,he would have definitely become a big man.

After marking the clause boundary, this sentence can be represented as :

అతను బ్రతికుంటే ,   ఇచ్చితంగా గొప్ప మనిషి అయ్యుందేవాదు.

Dependent clause   Independent clause

There are two clauses:

Dependent clause: అతను బ్రతికుంటే (athanu brathikumte)

Independent clause: అతను ఇచ్చితంగా గొప్ప మనిషి అయ్యుందేవాదు (athanu khachithamgaa goppaa manishi ayyumdevaadu) shown in the above sentence.

Various components of independent clause can be directly matched for various grammatical agreements. Now, the dependent clause contains correlative conjunction

అతను–వుంటే (athanu – umte) and this dependent clause can be converted into independent clause after removing this correlated conjunction from the dependent clause. The independent clause obtained after removing the correlative conjunction is:

Independent clause: అతను గొప్ప మనిషి అయ్యుందేవాడు (athanu goppaa manishi ayyumdevaadu).

Now various components of this independent clause are matched for various grammatical agreements.

The algorithm used for grammar checking of non-relative (conditional) type complex sentence is as follow:
Algorithm used: Databases used: Grammar Checking options
Input: Tokenized and annotated Telugu sentence
Output: Updated tokenized and annotated Telugu sentence

1. Using the information of clause boundary, separate the dependent and independent clauses and store them in separate tables.
2. Convert the dependent clause into independent clause by removing the correlated conjunction from dependent clause and call it as clause2.
3. Get all the grammar checking options, having the OnOff value set to 1, from the Grammar checking options database sorted by the Priority field.
4. Repeat steps 5 to 6 for both independent clauses.
5. Repeat steps 6 for all the grammar checking options.
6. Call the respective method to perform the required check on the current independent clause.
7. Reconstruct dependent clause by adding the correlated conjunction in clause2.
8. Reconstruct the complex sentence by joining the dependent clause and independent clause.
9. Output the rectified complex sentence.

ఆ బాలుడు స్కూలుకి వెళ్ళుంటే, అతను పాస్ అయ్యుందేవాడు.

(aa baaludu skooluki vellumte, athanu paass ayyumdevaadu).

If that boy went to school then he might have passed.

There are two clauses; one is ఆ బాలుడు స్కూలుకి వెళ్ళుంటే (aa baaludu skooluki vellumte) and second is అతను పాస్ అయ్యుందేవాడు (athanu paass ayyumdevaadu) shown in the above sentence. After sharing the common subject and removing the conjunction, the clauses will be :

బాలుడు స్కూలుకి వెళ్ళాడు (baaludu skooluki vellaadu)- Dependent Clause

బాలుడు పాస్ అయ్యుందేవాడు (baaludu pass ayyumdevaadu)-Independent Clause

Both these clauses contain subject verb agreement mistake as in both clauses subject బాలుడు (baludu) which is masculine does not grammatically agree with వెళ్ళాడు (vellaadu), అయ్యుందేవాడు (ayyumdevaadu) which is feminine. The complete architecture with example has been shown in fig 2.5.
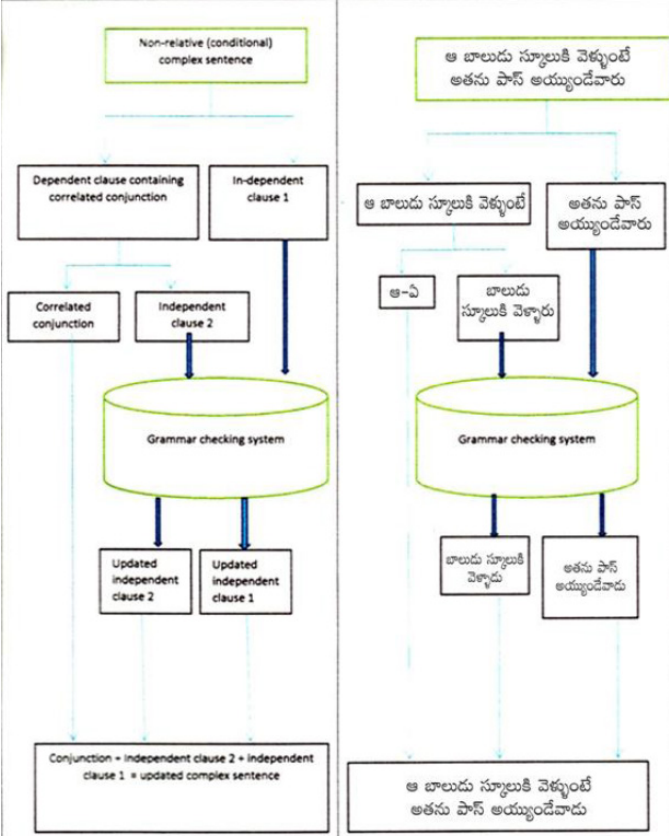

Fig 2.5: Architecture of Grammar Checker for Non-relative (conditional) Type System

3. CONCLUSION

A final activity of Grammar Checking System that includes various algorithms for Grammar Checking of complex sentences has been discussed in this paper. Besides these, a complete architecture of error detection and correction mechanism used in various types of complex sentences containing a non-finite verb, adverb clause and relative clause have been discussed. Various types of grammatical errors like agreement error, postposition error etc. have been detected and corrected in a complex sentences have been discussed..Checking Grammar categories of all features of Telugu grammar complex sentences discussed in this paper and the basis of various approaches involved in checking grammar were covered.

4. REFERENCE

1. Bustamante, F. R., & Le6n, F. S. 1996. GramCheck: A grammar and style checker. In Proceedings of the 16th conference on Computational linguistics-Volume 1. Association for Computational Linguistics. pp. 175-181
2. Bharati, A., Chaitanya, V., Sangal, R., & Ramakrishnamacharyulu, K. V. 1995. Natural language processing: a Paninian perspective. New Delhi: Prentice-Hall of India. pp. 65-106.
3. Bigert, J., Kann, V., Knutsson, O., & Sjobergh, J. 2004. Grammar checking for Swedish second language learners. pp. 33-47.
4. Sanjeev kumar Sharma, G.S Lehel 'Identification of Compound Sentences in Punjabi Language' Research Cell: An International Journal of Engineering Sciences, InauguralIssue2010ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online) Vol. 1, pp. 1-8.
5. Carlberger, J., Domeij, R., Kann, V., & Knutsson, O. 2004. The development and performance of a grammar checker for Swedish: A language engineering perspective. Natural language

engineering, 1(1).

6. Beesley, K. R. 2001. Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001. In ACL Workshop on Arabic Language Processing: Status and Perspective Vol. 1, pp. 1-8.

7. Chidambaram, D. 2005. Processing complex sentences for information extraction. A Thesis Presented in Partial Fulfillment of the Requirements for the Degree Master of Science.

8. Ehsan, N., & Faili, H. 2010. Towards grammar checker development for Persian language. IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), 2010. pp. 1-8

9. Fernandes, E. R., Pires, B., dos Santos, C. N., & Milidiiu, R. L. 2009. Clause identification using entropy guided transformation learning. IEEE 2009 Seventh Brazilian Symposium in Gill, M. S., Lehal, G. S., & Joshi, S. S. 2009. Part of speech tagging for grammar checking of Punjabi. The Linguistic Journal, 4(1), pp. 6-21.

10. Information and Human Language Technology (STIL), pp. 117-124.

11. Hein, A. S. 1998. A Chart-Based Framework for Grammar Checking Initial Studies. In Proc. of 11 th Nordic Conference in Computational Linguistic. pp. 68-80.

12. Jurafsky, Daniel and James H. Martin. 2000. Speech and Language Processing: An Introduction to Natural language Processing, Computational Linguistics, and Speech Recognition. Pearson Education, Delhi, India

13. Kubon, V., & Platek, M. 1994. A grammar based approach to a grammar checking of free word order languages. In Proceedings of the 15th conference on Computational linguistics-Volume 2. Association for Computational Linguistics. pp. 906-910

14. Naber, D. 2003. A rule-based style and grammar checker. Thesis, Technical Faculty, University of Bielefeld, Germany

15. http://simple.wikipedia.org/wiki/Telugu_language

16. http://en.wikipedia.org/wiki/Telugu_grammar