An Approach of Checking Grammar for Telugu Language Compound Sentences

Dr. V Suresh,

Associate Professor,

Department of Information Technology,

Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India

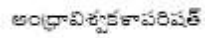
Abstract: For grammar checking of compound sentences, it is necessary to identify the structure of these sentences. The structure of compound sentences can be identified on the basis of number of clauses and types of clauses present in them. A sentence composed of single independent clause is called a simple sentence and a sentence having more than one independent clause is a compound sentence. Once the sentence is identified as compound 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 compound sentences of Telugu language has been done with grammatical error detection and correction. This research work on grammar checking of compound sentences is based on the assumption that the input sentences will be in Telugu script.

Key words : Independent clause, divide and conquer,Adverb clause , noun phrase, verb phrase.

1.INTODUCTION :

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 11^{th} 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.  (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 COMPOUND 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.

2.1.1. Types of grammatical errors in compound sentences:

As the compound sentences are composed of more than one independent clauses, therefore each independent clause undergoes detection of following types of errors:

2.1.1.1. All the Noun phrases joined by conjunctions to form group having same case:-

If in an independent clause, there are two or more than two noun phrases (NP) joined by a conjunction, then all these noun phrases (NP) must have same case.

Consider the following example:

Incorrect example:

అబ్బాయి	మరియు	అమ్మాయిలు	విందులో	పాల్గొన్నారు
Noun Phrase1	Conjunction	Noun Phrase2		

(abbaayi mariyu ammaayilu vindulo paalgonnaadu)

Boy and girls are participated in the function

There are two noun phrases. One is అబ్బాయి (abbaayi) and second is అమ్మాయిలు (ammaayilu) joined by conjunction మరియు (mariyu) అబ్బాయి (abbayi) is the noun phrase in direct case whereas అమ్మాయిలు (ammayilu) is the noun phrase in an oblique case in the above example. As per Telugu grammar, both these noun

phrases should have the same case. The correct sentence should be:

Corrected sentence:

Telugu : అబ్బాయిలు మరియు అమ్మాయిలు విందులో పాల్గొన్నారు.
Roman Transliteration : (abbaayilu mariyu ammaayilu vinduloo palgonnaaru).
English : Boys and girls are participated in the function.

2.1.1.2.

యొక్క (yokka), కలుపుట (kaluputa) should be in Agreement with Noun Phrase next to it in terms of Number and Gender :

If there is యొక్క (yokka) postposition in a sentence, then this postposition should be grammatically in agreement with noun phrase next to this postposition in terms of number and gender. Consider the following example :

Incorrect example:

అబ్బాయి ది పుస్తకం పోయింది
pp noun phrase

(abbaayi thi pusthakam poyimdi)

Boy's book was lost

As shown above, పుస్తకం (pusthakam) is the noun phrase next to the ది (di) postposition. Therefore, as per Telugu grammar, యొక్క (yokka) (postposition) should be grammatically in agreement with its next noun phrase i.e. పుస్తకం (pusthakam). Since పుస్తకం (pusthakam) is feminine noun, so the postposition ది (di) should be in its feminine form and its feminine form is యొక్క (yokka). Therefore, correct sentence should be:

Correct example:

అబ్బాయి యొక్క పుస్తకం పోయింది.

(abbaayi yokka pusthakam poyimdi). Boy's book was lost.

2.2.1.3. Modifier and Noun Agreement

The modifier and noun should be grammatically in agreement in terms of number and gender in a noun phrase.

Consider the following example:

కొంతమంది అబ్బాయిలు ఆటలాడుకున్నారు.
(konthamamdi abbaayilu aatalaadukunnaru)
Some boys were playing.

Incorrect example:

కొంతమంది అబ్బాయి ఆటలాడుకున్నారు
(komthamamdi abbaayi aatalaadukunnaaru)
Some boy were playing

The modifier కొంతమంది (komthamamdi) modifies the noun అబ్బాయి (abbaayi). Therefore, as per Telugu grammar,అబ్బాయి (abbaayi) should be grammatically in agreement in terms of number and gender in the above sentence. Since కొంతమంది (komthamamdi) is singular modifier and అబ్బాయిలు (abbaayilu) is the plural noun, therefore, modifier కొంతమంది (komthamamdi) should be plural in order to be in agreement with plural noun అబ్బాయిలు (abbaayilu). Hence, the correct sentence should be:

Correct example:

కొంతమంది అబ్బాయిలు ఆటలాడుకున్నారు.

(konthamamdi abbaayilu aatalaadukunnaaru).

Some boys were playing.

2.2.1.4. Order of Modifier of Noun Phrase

If a noun has two or more modifiers then the order of modifier is fixed in an independent clause. Generally, numerals precede adjective or verb phrases. Consider the following example:

Incorrect example:

అయిదొవ అందమైన బాలుడు చెప్పాడు

Numeral Adjective

(aidova amdamaina baaludu cheppaadu) Fifth handsome boy said

The order of modifier is not in accordance with the Telugu grammar rule as the numerals అయిదొవ (aidova) succeed adjective అందమైన (amdamaina) instead of preceding it shown in the above example. So, the **correct sentence** should be:

అందమైన అయిదొవ బాలుడు చెప్పాడు.

(amdamaina aidova baaludu cheppaadu). Handsome fifth boy said.

2.2.1.5. Order of Word in Verb Phrase

The words in the verb phrase should follow the specific sequence i.e. main verb should be followed by an operator and an operator should be followed by an auxiliary verb as discussed . Consider the following example:

Incorrect example:

బాలుడు	స్కూలు	కి	తున్నాడు	వెళు
		operator	auxiliary verb	Main verb
(baaludu skoolu ki thunnaadu velu)				Boy is school going

The main verb is following the auxiliary verb which is not in accordance with Telugu grammar as mentioned in chapter 3 under section 3.2.2 in the above example. The correct sequence of words in the verb phrase present in the above example should be:

Main verb + operator + auxiliary verb వెళుతున్నాడు (veluthunnaadu)

Therefore, the correct sentence should be:

Correct example:

బాలుడు స్కూలుకి వెళుతున్నాడు.	
(baludu skooluki veluthunnaadu).	Boy is going to school.

2.2.1.6. Style Error

Some errors occur due to missing a punctuation mark, use of an in-appropriate punctuation mark or by using duplicate words in a sentence. In this research work,three types of style errors have been covered; one is an error due to using an inappropriate punctuation, second error due to use of a duplicate word in a sentence and third is due to missing a punctuation mark. All these errors have been discussed in the following section:

2.2.1.6.1. Error due to using In-Appropriate Punctuation

These type of errors occur due to use of an incorrect punctuation mark or missing punctuation mark in the sentence. Consider the following example:

Incorrect sentence:

అతను ఏ స్కూలుకి వెళ్ళాలి.	
(athanu ye schoolki vellaali.)	In which school he has to go.

Above sentence is an interrogative sentence, but instead of using a question mark (?) the sentence ends with an affirmative i.e. full stop (.). Therefore, it is incorrect use of punctuation mark. The correct sentence should be:

అతను ఏ స్కూలుకి వెళ్ళాలి?	
(athanu ye schoolki vellaali ?)	In which school he have to go?

2.2.1.6.2. Error due to Missing Conjunction or Punctuation Mark

Compound sentences are composed of independent clauses separated by conjunctions.These conjunctions include some punctuation marks like comma (,) or they may be words belonging to coordinate conjunctions word class.

If this conjunction is missing in the sentence, then two clauses will merge into single clause and it becomes difficult to process the compound sentence.

Consider the following example:

Incorrect example:

అమె లేవక మునుపే దొంగతనం జరిగిపోయింది (aame levaka munupe domgathanam jarigipoyimdi) Robbery had happened before she woke up

There are two clauses; one is adverb clause i.e. అమె లేవక మునుపే (aame levaka munupe) and second is independent clause i.e. దొంగతనం జరిగిపోయింది (domgathanam jarigipoyimdi) shown in the above sentence. The two clauses should be separated by comma i.e. there should be a comma (,) after the adverb clause అమె లేవక ముందే (aame levaka mumde). Therefore, this sentence has a missing punctuation mark. The correct sentence should be :

Correct example:

అమె లేవక ముందే, దొంగతనం జరిగిపోయింది. (aame levaka mumde , domgathanam jarigipoyimdi). Robbery had happened , before she woke up.

2.2.1.6.3. Error due to Duplicate Words

While writing a sentence, a word is typed twice. This results in unstructured sentence and it becomes difficult to understand the meaning of the sentence. Consider the following example:

Incorrect example:

అ అమ్మాయి అందంగా అందంగా వుంది (aa ammaayi amdamgaa amdamgaa umdi)

The అందంగా (amdamgaa) is a duplicate word as it has been typed twice in the above sentence. One of these duplicate words should be removed from the sentence. The correct sentence should be :

Correct sentence:

ఆ అమ్మాయి అందంగా వుంది. (aa ammaayi amdamga umdi).

Various types of grammatical mistakes in an independent clause analyzed by researcher's system are listed in Table 1.1. These errors are basically categorized into three classes; first is agreement error; second is postposition related errors and third is style error. First column of the table represents the type number of the error, second column represents the name of the grammatical mistake and third column shows the example containing incorrect and correct sentences related with the corresponding error shown in second column.

Each independent clause is checked for various grammatical errors. All these errors are detected in sequence as mention in Table 1.1.

Table 1.1 : Various Error types handled by the system

Error Type	Type of grammatical mistake	Examples
Type 1	All the Noun phrases joined conjunctions to form group have same Case	**Incorrect :** అబ్బాయి మరియు అమ్మాయిలు విందులో పాల్గొన్నారు (abbaayi mariyu ammaayilu vindulo paalgonnaadu) **Correct :** అబ్బాయిలు మరియు అమ్మాయిలు విందులో పాల్గొన్నారు. (abbaayilu mariyu ammaayilu vindulo palgonnaaru)
Type 2	Style error	**Incorrect :** ఇది నా ఇల్లా (idi naa illaa) **correct :** ఇది నా ఇల్లు. (idi naa illu)
Type 3	Noun phrase must be in oblique form before postposition	**Incorrect :** అబ్బాయి ఇప్పుడు వెళ్ళాలి (abbaayi ippudu vellaali) **correct :** ఆ అబ్బాయే ఇప్పుడు వెళ్ళాలి. (aa abbaaye ippudu vellaali)
Type 4	ది (di) should be replaced by యొక్క (yokka) in agreement with noun phrase in terms of number and gender	**Incorrect :** అబ్బాయిది పుస్తకం పోయింది (abbaayidi pusthakam poimdi) **correct :** అబ్బాయి యొక్క పుస్తకం పోయింది. (abbaayi yokka pusthakam poimdi)

2.2.2. APPROACHES USED FOR GRAMMATICAL ERRORS IN COMPOUND SENTENCES :

Our system checks the errors in the sentence at phrase level, clause level and then at sentence level. For phrase level and clause level, rule based approach has been followed. For sentence level, this rule based approach has been extended to all the clauses of the sentence. The compound sentences are composed of more than one independent clauses. In a compound, sentence the subject and verb agreement takes place at the clause level. Therefore, each independent clause is checked for this subject-verb agreement. The grammar "checking in compound sentences takes place in two phases:

Phase 1:

In this phase, style error at the sentence level is identified and rectified. This is done by analyzing the complete structure of the input compound sentence. As discussed, the compound sentences have fixed patterns of structure. An input compound sentence is matched against these fixed patterns. In order to snatch the input sentence against these patterns, a database containing all the possible patterns of compound sentences has been developed and stored in the form of regular expression. The design of database is shown in below table 1.2:

Table 1.2 : Database Design of Compound Sentence Structure

Field Name	**Description**
Priority	Priority for this structure
Sid	Identifier for this structure
TagSequence	Tags that will form this structure
Comments	Additional information about this sentence

Sample database entries

<SentenceStructure>

<Priority> 1
</Priority><TagSequence>(/IDC\(\w+\)IDC(Comma)(/IDC\(\w+\)IDC/)</TagSequence>

<Sid> l </Sid>

<Comments>compound sentence patteren 1:- (IDC),(IDC)</Comments>

</SentenceStructure>

<SentenceStructure><Priority>2</Priority><TagSequence>(/IDC\(\w+\)IDCO(CJ\w+)(/IDC\(\w+\)IDCO</TagSequence><Sid>2</Sid>

<Comments>compound sentence patteren 2:- (IDC)CJ(IDC)</Comments>

</SentenceStructure>

This database has been used to detect errors in the structure of compound sentences.
TagSequence may be formed of POS tags, phrase tags, and/or phrase group tags.

Algorithm used: To match the structure of compound sentences

Databases used: Compound sentence patterns. Input: Compound sentence

Output: Sentence ID of the matched structure.

1. Get all the structures, having *OnOff*value set to 1, from the compound sentence pattern database sorted by the *Priority* field.
2. Repeat steps 3 to 5 for all the sentences.
3. Get Tag values of all the words in the current sentence and create a TagSequence.
4. Repeat step 5 for all the compound sentence patterns.
5. If there is a pattern having sequence of tags same as created in step 3 and stored in TagSequence then return the Sid of that pattern.

   Else

   Return 0;

Phase 2:

In the second phase, internal structure of compound sentences is analyzed for detection of errors. In the internal structure, errors at phrase level and then at clause level are detected and rectified. In order to perform error detection and correction at phrase and clause levels, each input compound sentence is simplified by splitting the sentence into independent clauses (simple sentences). This simplification of sentences is performed on the basis of clause boundary mark information. To check the grammatical errors in simple sentences or in independent clauses of Punjabi sentences, `government and binding' prevalent in Punjabi sentences has been studied. As per `government and binding' prevalent, there exists a grammatical agreement between various components of a sentence like modifier and noun agreement, subject/object and verb agreement, noun and adjective agreement, noun phrase in oblique form before postposition etc. Also, as per government and binding, all the words present in an independent clause must grammatically agree with the head word of that clause. The head word of the clause is the first phrase head that is present in the noun phrase of the clause. Various types of errors detected and corrected by the system developed by the researcher has been listed in table 1.2. For each error type mentioned in table 1.1, a separate module has been developed. Each module detects and rectifies a specific type of error. During detection of error, all these modules are executed in sequence.

The algorithm used in the grammar checking of compound sentences is as following:

Algorithm:

For error detection in compound sentences:

Databases used: Error type.

In put: incorrect simple sentence (independent clause)

Output: Corrected simple sentence.

1. Get all the error type that have *On Off* value set to 1, from the error type database sorted by the *Priority* field
2. Repeat steps 3 to 5 for current clause.
3. Repeat steps 4 and 5 for all the error types.
4. Call the respective module to perform the correction on the current clause.

5. Output the corrected sentence.

Consider the following incorrect compound sentence:

Incorrect sentence:

రాముడు మంచి బాలురే కాని అతనికి కళాశాలలో ర్యాంక్ రాదు (raamudu mamchi baalure kaani athaniki kalaasalalo ryaamk raadu)

There are two clauses in the compound sentence and each clause contains errors in the above incorrect sentence. The first clause is రాముడు మంచి బాలురే (raamudu mamchi baalure) and it contains noun modifier agreement error as the modifier రాముడు (raamudu) (singular) does not grammatically agree with noun బాలురే (baalure) (plural) in terms of number. Second clause is కాని అతనికి కళాశాలలో ర్యాంక్ రాదు (kaani athaniki kalaasalalo ryaamk raadu) and it contains subject verb agreement error as the object అతనికి (athaniki) (feminine) does not grammatically agree with verb రాదు (raadu) (masculine) in terms of gender. Both these errors are detected and rectified in two steps. In the first step, two clauses are separated from the sentence and in the second step, these clauses are detected for the presence of error. After applying detection and correction on individual clauses, the final updated output given by the researcher's system is:

Corrected sentence:

రాముడు మంచి బాలుడే కాని అతనికి కళాశాలలో ర్యాంక్ రాలేదు. (raamudu mamchi baalude kaani athaniki kalaasalalo ryaamk raaledu).

The complete architecture of the above method with example has been shown in figure 1. l. It is clear from figure 1.1 that the compound sentence is first simplified by splitting it at the conjunction and separating each independent clause. Then each independent clause is passed through grammar checking system where error detection and correction algorithm mention above is applied.

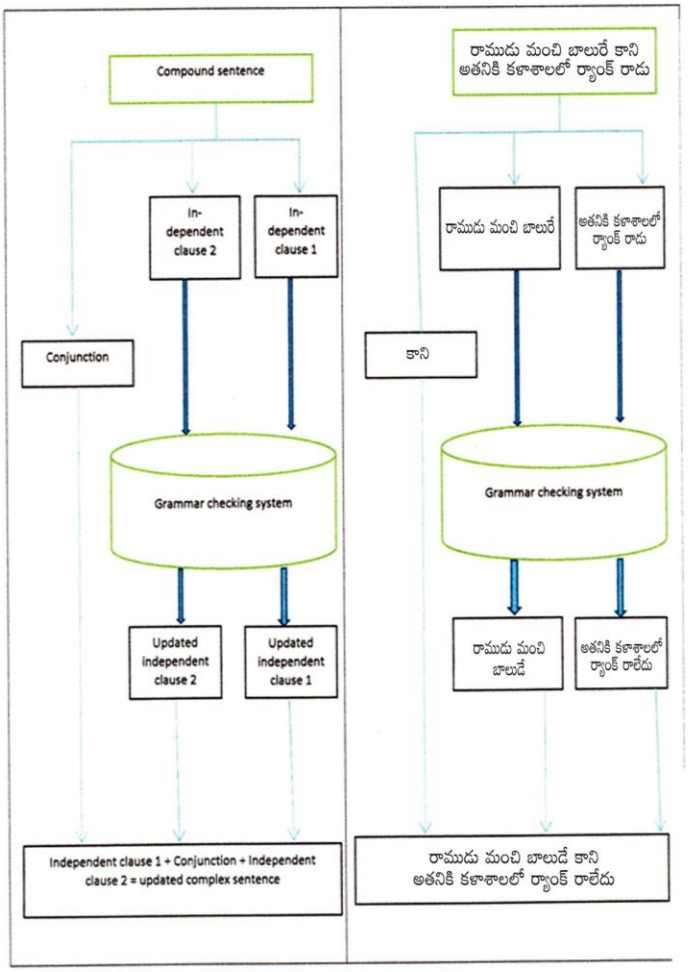


Fig 1.1 : Architecture of grammar checking of compound sentence

3.CONCLUSION

A final activity of Grammar Checking System that includes various algorithms for Grammar Checking of compound sentences has been discussed in this paper Besides these, a complete

architecture of error detection and correction mechanism used in compound sentences have been discussed. Various types of grammatical errors like agreement error, postposition error etc. have been detected and corrected in a compound sentences have been discussed..Checking Grammar categories of all features of Telugu grammar compound sentences discussed in this paper and the basis of various approaches involved in checking grammar were covered.

4.REFERENCES

1. 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).*

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. 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

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