Punjabi Grammar Checker Components, Scope and Techniques

[1]Leekha Jindal, [2]Vijay Rana, [3]Sanjeev Kumar Sharma
[1]Research Scholar, [2,3]Assistant Professor
[1,2]Department of Computer Science and Applications,
Sant Baba Bhag Singh University, Jalandhar
[3]Assistant Professor, Department of Computer Science and Applications
DAV University, Jalandhar

ABSTRACT

In this paper, general overview of one of the major NLP tool i.e. grammar checker is provided. An overview of basic components like pre-processor, morphological analyzer, part of speech tagger, phrase chunker, clause detection, parser and error detection and correction has been provided. Further various techniques used for developing a grammar checker i.e. rule based, syntax based and statistical based techniques are also discussed. In the end of the paper various types of syntactic errors present in the Punjabi language and scope of the Punjabi grammar checker is specified.

KEYWORDS

Punjabi grammar checker, morphological analyzer, POS tagger.

INTRODUCTION TO GRAMMAR CHECKER

Grammar checker can be defined as an automated system (software) that checks the sentence of a given language against the linguistic rules of that language. 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. Primary functional diagram of the grammar checker has been shown in figure1.

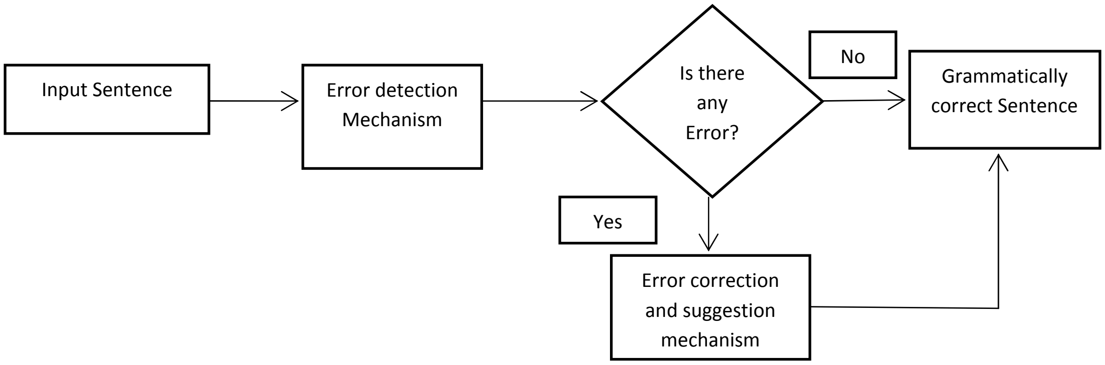
Input Sentence
Error detection Mechanism
Is there any Error?
No
Grammatically correct Sentence
Yes
Error correction and suggestion mechanism

Fig 1 : Simplified functional diagram of grammar checker

As shown in figure 1, error in the input sentence is revealed by using error detection mechanism. If error is found, this error is corrected by using error correction mechanism. Grammar checking is a relatively new field, with earliest systems reported in 1970s. Earliest grammar checkers used to focus more on style and punctuation errors. Later actual grammar checking features were incorporated in these systems. They were marketed as standalone applications until 1992. Then Microsoft introduced grammar checker (for English) in its product, Microsoft Office, in 1992 by licensing an existing grammar checker. WordPerfect also followed Microsoft's move. The incorporation of grammar checkers in these widely used office suites had catastrophic effects on grammar checking research. Due to the dominant position of these word processing systems, other research groups refrained from investing in grammar checking though still some open-source projects are going on.

AUTOMATIC GRAMMAR CHECKING TECHNIQUES

Different techniques are used by different researchers in developing typical grammar checking systems. Some researchers used syntax based approach, some used rule based approach and other followed the statistical based approach. So, these approaches can be categorized into three types; syntax based approach, statistical based approach and rule based approach. These three approaches have been discussed in the following section:

Syntax based approach

In this type of approach, full parsing of the input sentence is performed. A sentence is parsed completely if it follows the grammatical rule. This is one of the best method used for grammar checking if a parser is developed by using all possible rules of the language. However, one problem of this method is that, it only tells about the correctness of the sentence. It does not provide any suggestion for the incorrectness of the sentence. If suggestions are required then new rules have to be implemented. This approach was used by Bernth (1997) [1] for development of EasyEnglish analyzer to check discourse and document level errors, Hein (1998) [8] used this approach for development of ScarCheck (a Swedish grammar checker), Ravin (1998) [10] developed a Text-Critiquing System to check grammar error and style weakness, Young-soog (1998) [11] used this approach for improvement in Korean proof reading system, Martin et al. (1998) [2] for development of Brazilian Portuguese grammar checker and Kabir et al. (2002) [14] for development of Urdu grammar checker.

Statistics based checking

This is a probability based approach. In this approach, a sequence of POS (part-of-speech) tags is generated from an annotated corpus (a corpus in which each word is associated with grammatical information in the form of part of speech tags) and then the frequency of these sequences and hence, the probability of these sequences is calculated. The input text will be considered more incorrect if the tag sequence generated by this input text has lower probability than some threshold. The basic requirement to implement this approach is pre-annotated corpus. The accuracy of this approach depends upon the amount and type of annotated corpora. The corpus should cover all the domains. Since this approach is purely statistical based, so sometimes it gives the unexpected results, and the user will never come to know about this. Another problem with this approach is that the results are very difficult to interpret. The advantage of this approach is that there is no need of the knowledge of language. Thus, this approach can be applied on any language without thorough knowledge of the language. N-gram based statistical technique has been used by Alam et al. (2006)[3] in the development of grammar checker for Bangla and English language, Carlberger et al. (2002,2004) [12-13]for Granska (a Swedish grammar checker), Ehsan and Faili (2010) [5] for Persian language, Temesgen and Assabie (2012)[6] for Amharic language, Kinoshita et al. (2006) [21] for CoGroo (a Brazilian

Portuguese grammar checker) and Verena Henrich and Timo Reuter (2009) [7] proposed a Language Independent Statistical Grammar (LISG) checking system.

Rule based checking

In this approach, rules are developed in the form of patterns. Every input sentence is checked against these patterns. If a sentence matches the pattern, it is grammatically correct otherwise it is incorrect. A thorough knowledge of the language is required for the development of rules. Generally, a linguistic is required to develop the rules. It is very difficult to develop all the possible rules. So, one can say that a rule-based system is never complete. Therefore, there will always be some errors that it fails to detect. The advantage of this approach is that the rules can be turned on and off as per requirement. Another advantage of this approach is that rules can be edited or added in later stage. This approach was used by Schmidt-wigger (1998) [9] for developing a grammar checker for German language. Kann (2002) [4] used hand written error rules in CrossCheck (a Swedish grammar checker), Naber (2003) [15] used purely rule based approach for development of English grammar checker, Rider (2005) [16] used hand constructed error rules in English grammar checker; Faili (2010) [5] used this technique for development of Persian grammar checker; Tesfaye (2011) [17] used this rule-based technique for Afan Oromo grammar checker, Jiang et al. (2011) [18] developed a rule-based Chinese grammar checker, Kasbon et al. (2011) [19] used rules for development of grammar checker of Malay language, Singh and Lehal (2008) [20] used rules for development of Punjabi grammar checker; Bal and Shrestha (2007) used this technique to develop Nepali grammar checker.

Many combinations of these approaches have been used to improve the accuracy of different grammar checkers.

COMPONENTS OF GRAMMAR CHECKER

There are certain components that are part of every grammar checking system. An overview of such components is provided in figure2. These components are usually organized in a sequence. However, depending on the technique being followed, some of these components may be omitted.

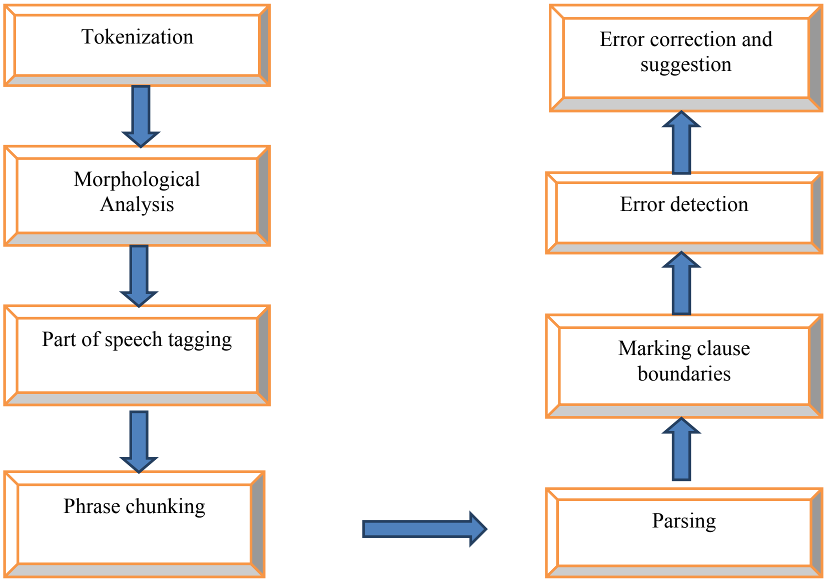

Figure 2: Key components in grammar checking system

Tokenization

A token is the atomic item of the sentence. This includes every word, number, punctuation mark, abbreviation, etc. In tokenization, the input text is first split into sentences and then these sentences are further broken down into tokens. This is the fundamental activity in almost all the grammar checking systems. The pre-processing unit of the grammar checker splits/breaks the input text into sentences by using sentence boundaries. Some of the common sentence boundaries for Punjabi language include question mark (?), sign of exclamation (!) and sentence ended marker (।). A list of sentence boundaries is already defined and fed into the system. After breaking the input text into sentences, each sentence is further divided into smaller parts called tokens by using word boundaries. The only word boundary used is space ( ). Another task that is performed in this activity is filtering of some special expressions. The special expressions include abbreviations, some fixed expressions and phrases, etc.

Morphological analysis:

In this activity, every token separated in previous activity is morphologically analyzed. Two tasks are performed in this activity; one is associating every word/token with its grammatical information and second is to provide the information of the root word for the given word. It is extremely significant for Indian languages as these languages have highly rich system of inflectional morphology. To associate the grammatical information with a word, tags are used. These tags represent the grammatical information of that word/token. Since a word can exist in more than one word class in different contexts, therefore, such word has more than one tag. For such type of words, all possible tags are assigned and appropriate tag is selected by the POS (part-of-speech) tagger in the next step.

Part-of-speech tagging

The main task of POS (Part-of-speech) tagger is to remove the ambiguities generated during morphological analyzer. The ambiguities arise due to assignment of more than one tag to a word. In POS tagging, an appropriate tag is selected out of more than one tag assigned to the word. In grammar checking, POS tagger plays an important role. The accuracy of grammar checker depends upon the accuracy of POS tagger. If an incorrect tag is assigned to the word, the grammar checker can raise a false alarm. Different approaches are used by different researchers for development of POS tagger. These approaches include rule based approach, stochastic based approach, neural network based approach etc.

Phrase chunking

In this activity, words are combined into groups to construct next higher unit called phrase. A phrase can be a noun phrase (NP), a verb phrase (VP), an adjective phrase, an adverb phrase, etc. The phrases have the fixed structure. A noun phrase (NP) can have a noun and one or more adjectives. A verb phrase (VP) can have a main verb, operator verb and an auxiliary verb in sequence. Phrase chunking can only be applied after the POS tagging because to group a sequence of words, we need their grammatical information.

Parsing

This activity is mandatory in syntax based grammar checking system as the sentence is parsed using a parser. The syntactic parsing of the input sentence is performed in this activity. The output of this activity is a syntax tree.

Clause boundary identification

It is the next level activity that is performed after phrase chunking. In this activity, the starting and the end point of the clause are identified and marked. Larger sentences like compound and complex sentences are composed of more than one clause. This activity helps in decomposing the sentence into smaller units and further helps to differentiate the simple, compound and complex sentences.

Error detection system

In this activity, a sentence or clause is checked against the grammatical rules of the language. Various types of errors like style error, agreement mismatch, order of modifier etc. are checked in this activity. In agreement checking, various components of the sentence like noun phrase (NP), verb phrases (VP) etc. are checked for their mutual grammatical agreement with respect to number, gender and case. These include the agreement checking between subject and verb, noun and modifier, between two different clauses, etc.

Error correction and suggestion

Generally this is the last component of a grammar checkers. In this activity, the appropriate action is taken or suggested to rectify the error, and all possible solutions are provided for correcting the error.

ERROR TYPES COVERED

Grammar checkers basically handles syntactic errors. This type of error occurs when the sentence does not follow the grammatical rule of the language in which it has been written. Table 1shows a list of syntactic errors with examples.

Table 1: Various types of syntactic errors

Sr. No.	**Incorrect sentence**	**Error type**
1.	ਦੋ ਮੁੰਡਾ ਸਕੂਲ ਜਾਂਦੇ ਹਨ। (dō muṇḍā sakūl jāndē han.)	Modifier and Noun agreement Error
2.	ਚਾਰ ਬੰਦੇ ਕੰਮ ਕਰ ਰਿਹਾ ਹੈ।(cār bandē kamm kar	Subject Verb agreement Error.

	rihā hai.)	
3.	ਦੋਵੇਂ ਮੁੰਡੇ ਅਮਰੀਕਾ ਜਾਕੇ ਗੋਰਾ ਹੋ ਗਏ।(dōvē ṃmuṇḍē amrīkā jā kē gōrā hō gaē.)	Noun and Adjective agreement Error.
4.	ਵੱਡਾ ਮੇਰਾ ਮੁੰਡਾ ਸ਼ਹਿਰਰਹਿੰਦਾ ਹੈ।	Order of modifier of Noun phrase.
5.	ਮੁੰਡਾ ਘਰ ਸੌਣ ਜਾ ਸੀ ਰਿਹਾ।(muṇḍā ghar sauṇ jā sī rihā.)	Order of word in Verb phrase.
6.	ਜੇ ਮੁੰਡਾ ਸਕੂਲ ਜਾਏਗਾ ਤਾਂ ਪਾਸ ਹੋ ਜਾਏਗੀ। (jē muṇḍā sakūl jāēgā tāṃ pās hō jāēgī.)	Noun phrase and verb phrase agreement between dependent and independent clause.
7.	ਦਵਾਈ ਪੀਂਦਿਆਂ ਹੀ ਮੇਰਾ ਦਿਲ ਘਬਰਾਉਣ ਲਗ ਪਈ ਸੀ।(davāī pīndiāṃ hī mērā dil ghabrāuṇ lag paīsī.)	Noun phrase and verb phrase agreement within independent clause.
8.	ਉਹ ਮੁੰਡਾ ਜਿਹੜ਼ਾ ਮੇਰੇ ਨਾਲ਼ ਪੜ਼੍ਹਦਾ ਸੀ ਬਹੁਤ ਗੋਰੀ ਹੈ। (uh muṇḍā jihḍaā mērē nāla଼ paḍhadā sī bahut gōrī hai)	Noun phrase and adjective agreement between dependent and independent clauses.
9.	ਉਹ ਮੁੰਡਾ ਜਿਹੜਾ ਮੇਰੇ ਨਾਲ ਪੜ੍ਹਦਾ ਸੀ ਅੱਜ ਬਹੁਤ ਸੋਹ ਣੀ ਗਾਉਂਦਾ ਹੈ। (uh muṇḍā jihṛā mērē nāl paṛhdā sī ajj bahut sōhṇī gāundā hai)	Noun phrase and adjective agreement within independent clause.
10.	ਮੇਰਾ ਭਰਾ ਜਿਹੜ਼ਾ ਬਹੁਤ ਵੱਡਾ ਅਫ਼ਸਰ ਹੈ ਅੱਜ ਮੇਰੇ ਘਰ ਆਈ ਸੀ।(mērā bharā jihḍaā bahut vaḍḍā aphsar hai ajj mērē ghar āī sī)	Common subject and verb agreement.

NEED AND SCOPE OF GRAMMAR CHECKER

The most common uses of the grammar checker are in report writing, official document writing where the grammar needs to be correct and in word processing. This can be implemented by adding the grammar checker package to the existing Punjabi word processor like "Akhar". It can be further helpful in machine translation system where the input and output needs to be grammatically correct. Another application of grammar checker could be learning the sentence formation and understanding the structure of compound and complex sentences for first language learners. As Indian languages mostly have common features so it can be extended to develop the grammar checker for compound and complex sentences for other Indian languages like Bengali, Oriya, Gujarati, Hindi, etc.

CONCLUSION AND FUTURE SCOPE

All the mentioned components are not mandatory in every grammar checker. The selection of activities depends upon approach used for grammar checking. For instance, a grammar checker developed to check the grammar of simple sentences may not include clause identification as this activity is for compound and complex sentences. Similarly, if a grammar checker is developed using N-gram technique there is no need of phrase chunker, clause identification and parsing because only POS tagging is sufficient for this. Similarly each technique used for developing the grammar checker has its own advantage and disadvantage. Therefore any technique can be used for development of grammar checker.

REFERENCES

[1]. Bernth, A. 1997. EasyEnglish: a tool for improving document quality. In Proceedings of the fifth conference on Applied natural language processing. Association for Computational Linguistics. pp. 159-165.

[2]. Martins, R. T., Hasegawa, R., Montilha, G., & De Oliveira, O. N. 1998. Linguistic issues in the development of ReGra: A grammar checker for Brazilian Portuguese. Natural Language Engineering, 4(04), pp. 287-307.

[3]. Alam, M. Jahangir, Naushad UzZaman, and Mumit Khan. 2006. N-gram based Statistical Grammar Checker for Bangla and English. In Proc. of ninth International Conference on Computer and Information Technology (ICCIT 2006).

[4]. Bigert, J., Kann, V., Knutsson, O., & Sjöbergh, J. 2004. Grammar checking for Swedish second language learners. pp. 33-47.

[5]. 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

[6]. Temesgen, A., & Assabie, Y. 2013. Development of Amharic Grammar Checker Using Morphological Features of Words and N-Gram Based Probabilistic Methods. IWPT-2013, p. 106.

[7]. Henrich, V. 2009. LISGrammarChecker: Language Independent Statistical Grammar Checking (Doctoral dissertation, Reykjavík University).

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

[9]. Schmidt-Wigger, A. 1998. Grammar and style checking for German. In Proceedings of CLAW (Vol. 98).

[10]. Ravin, Y. 1993. Grammar Errors and Style Weaknesses in a Text-Critiquing System. In Natural Language Processing: The PLNLP Approach. Springer US. pp. 65-76.

[11]. Young-Soog, C. 1998. Improvement of Korean Proofreading System Using Corpus and Collocation Rules. Language, pp. 328-333.

[12]. Carlberger, J., Domeij, R., Kann, V., & Knutsson, O. 2002. A Swedish grammar checker. Submitted to Comp. Linguistics, oktober.

[13]. 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).

[14]. Kabir, H., Nayyer, S., Zaman, J., & Hussain, S. (2002, December). Two Pass Parsing Implementation for an Urdu Grammar Checker. In Multi Topic Conference, 2002. Abstracts. INMIC 2002. International (pp. 51-51). IEEE.

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

[16]. Rider, Z. 2005. Grammar checking using POS tagging and rules matching. In Class of 2005 Senior Conference on Natural Language Processing.

[17]. Tesfaye, D. 2011. A rule-based Afan Oromo Grammar Checker. IJACSA Editorial.

[18]. Jiang, Y., Wang, T., Lin, T., Wang, F., Cheng, W., Liu, X., & Zhang, W. 2012. A rule based Chinese spelling and grammar detection system utility. IEEE International Conference on System Science and Engineering (ICSSE), 2012. pp. 437-440

[19]. Kasbon, R., Amran, N., Mazlan, E., & Mahamad, S. 2011. Malay language sentence checker. World Appl. Sci. J.(Special Issue on Computer Applications and Knowledge Management), 12, pp. 19-25.

[20]. Gill, M. S., & Lehal, G. S. 2008. A grammar checking system for Punjabi. In 22nd International Conference on Computational Linguistics: Demonstration Papers. Association for Computational Linguistics. pp. 149-152.

[21]. Kinoshita, J., Salvador, L. N., & Menezes, C. E. D. 2006. CoGrOO: a Brazilian-Portuguese Grammar checker based on the CETENFOLHA Corpus. In The fifth international conference on Language Resources and Evaluation, LREC.