Syntactic Error Detection System Using HMM

[1]Leekha Jindal,[2] Ravinder Jindal, [3]Sanjeev Kumar Sharma
[1]Research Scholar, [2]Research Scholar, [3]Associate Professor
[1,2] SBBS University, Jalandhar
[3]DAV University, Jalandhar
[1]leekhajindal@gmail.com, [2]rmjindal2002@gmail.com, [3]sanju3916@rediffmail.com

ABSTRACT

Having an error detection and correction system is a fundamental requirement for any word processing application such as MS Word, Applix Word, JWPce, KWord, etc. Despite various efforts to develop such systems using rule-based, statistical-based, and other machine learning approaches, none of them have been satisfactory. The author of this research proposes an algorithm that utilizes the Hidden Markov Model to detect grammatical errors in input sentences. The Viterby algorithm is used to implement the Hidden Markov Model, and an annotated corpus from ILCI is used to calculate the HMM parameters. The results of testing the system on three types of datasets showed an overall precision of 100%, recall of 93.83%, and an f-measure of 96.7. The proposed algorithm has the potential to be used in the development of similar systems for other Indian languages.

Keywords: Grammar checker, syntactic analyzer, error detection, HMM

Introduction to grammar checker:

Automated grammar checking systems are computer programs designed to identify and correct grammatical errors in text. They are widely used for proofreading purposes, such as in newspapers, magazines, novels, reports, theses, and other written materials. These systems use various approaches, including rule-based, statistical-based, machine learning-based, and hybrid methods. Rule-based systems use predefined grammatical rules to detect errors, while statistical-based systems use probabilistic models and annotated corpora to identify errors. Machine learning-based systems use neural networks and other techniques to learn from examples and improve their accuracy over time. Hybrid systems combine multiple approaches to achieve higher precision and recall. Automated grammar checking systems not only detect errors but also provide suggestions for correcting them, which can significantly improve the efficiency and accuracy of writing and translation tasks. Further, a grammar checker is a computer program that automatically checks the grammatical accuracy of typed or input text, ensuring it adheres to the grammatical rules of the language in which it was written. These systems not only detect errors, but also provide suggestions for correcting them, making them useful for proofreading various written works, such as newspapers, magazines, novels, reports, theses, and stories. They also play a vital role in improving the effectiveness of machine translation systems, where both the input and output texts must be grammatically correct. Many automated error detection systems have been developed using different approaches, including phrase-based statistical machine translation, specialized grammar formalisms, machine learning, finite state automata, and syntax-based techniques, among others. Additionally, a combination of these approaches has been used to develop grammar checkers for various languages, including English, Czech, Swedish, Dutch, Bulgarian, Bangla, Punjabi, Amharic, Nepali, and Indonesian. The general architecture of these systems is shown in Figure 1. Furthermore, some studies have explored the use of neural networks, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, for text correction and completion in keyboard decoding and spell and grammar checking for Swedish and English languages. Finally, some researchers have proposed a search engine-based grammar checker that uses the internet as a normative corpus and developed rule-based grammar checkers.

Existing Grammar Checking Approaches:

Automated grammar checking systems have become an essential part of many writing tasks, including academic writing, professional writing, and personal communication. These systems use a variety of approaches, including rule-based, statistical-based, and machine learning-based methods to identify grammatical errors and provide suggestions for correction. The effectiveness of these systems varies depending on the approach used, the language being checked, and the complexity of the grammar rules involved. Despite their limitations, automated grammar checking systems have been shown to significantly improve the efficiency and accuracy of writing tasks. However, it is important to note that these systems should be used as a tool to aid in the writing process, rather than a substitute for human proofreading and editing. Overall, automated grammar checking systems are a valuable tool for improving writing skills and ensuring grammatically correct written communication.

A. Rule base grammar checking:

Rule-based automatic grammar checkers have been widely used for detecting grammatical errors in various languages. These systems work on the basis of a set of predefined rules that are applied to the input text to check its grammatical correctness. The rules are usually derived from the grammar rules of the target language and cover a wide range of grammatical aspects such as verb agreement, tense, prepositions, and word order. One of the advantages of rule-based systems is that they are transparent, as the rules are explicitly defined and can be modified by language experts to improve the system's performance. Moreover, rule-based systems are generally faster and more accurate than statistical-based systems for languages with rich morphology and complex syntax, as they can handle the language's specific grammatical rules. However, the limitations of rule-based systems are also apparent. They rely on the correctness and completeness of the rule set, which is difficult to achieve in practice. Additionally, the rules may not cover all the possible variations of the language and may result in false positives or false negatives. Furthermore, rule-based systems require extensive language expertise and time to develop and maintain, making them expensive and challenging to scale to new languages or domains. Overall, rule-based automatic grammar checkers have been widely used and have shown promising results in detecting grammatical errors. However, they also have limitations that need to be addressed, and further research is required to improve their accuracy, coverage, and scalability. There are numerous languages for which rule-based grammar checkers have been developed. Here is a list of some of the languages:

- English
- French
- German
- Czech
- Swedish
- Danish
- Bulgarian
- Bangla
- Korean
- Punjabi
- Afan Oromo
- Nepali
- Latvian
- Indonesian

This is not an exhaustive list, and there may be other languages for which rule-based grammar checkers have been developed.

B. **Statistics based grammar checker**:

Statistical-based automatic grammar checkers rely on large corpora to identify patterns and probabilities of language use. These systems analyze the text and compare it to a large database of language usage statistics to identify potential grammar errors. These systems are highly accurate and can detect many common grammar mistakes. However, they may struggle with more complex grammar issues or language usage that is less common. In addition, these systems may not provide explanations for the suggested corrections or take into account the context of the sentence. This can lead to suggestions that are not necessarily the best fit for the specific text or writing style. Despite these limitations, statistical-based automatic grammar checkers are widely used and can be highly effective tools for identifying and correcting many types of grammar errors. They are often integrated into other writing tools, such as word processors or writing platforms, making them easily accessible and widely used by writers. Some of the languages for which statistic-based grammar checker systems have been developed are:

- German
- Spanish
- Greek
- Brazilian-Portuguese
- English
- Indonesian

Note: This is not an exhaustive list, and there may be other languages for which statistic-based grammar checker systems have been developed as well.

C. **Syntax based grammar checker**:

Syntax-based automatic grammar checkers use rules of syntax and morphology to detect and correct grammatical errors. They analyze the structure of the sentence, identifying parts of speech and their relationships, and use this information to determine the correctness of the sentence. These systems are designed to detect errors such as subject-verb agreement, tense consistency, and pronoun usage. The advantage of syntax-based systems is that they can handle complex sentences and detect errors that are not easily detected by rule-based or statistical-based systems. They are also able to provide more detailed explanations for the errors, which can be helpful for language learners. However, syntax-based systems may have limitations when it comes to detecting errors in informal or non-standard language usage. They may also struggle with ambiguity and idiomatic expressions, which can lead to false positives or false negatives. Overall, syntax-based systems can be useful tools for improving the grammatical correctness of written language, but they should be used in conjunction with human proofreading to ensure accuracy. Syntax-based grammar checkers have been developed for several languages, including but not limited to:

- English
- German
- Czech
- Bulgarian
- Bangla
- Dutch
- Swedish
- Urdu
- Latvian

These syntax-based grammar checkers use different techniques such as chart-based approach, finite state automata, and syntax-based parsing. The main advantage of syntax-based grammar checkers is that they can detect more complex grammatical errors than rule-based or statistics-based systems. However, they are often more computationally intensive and may require more advanced linguistic knowledge for development.

Machine learning based grammar checker:

Machine learning-based automatic grammar checkers have emerged as a promising approach to improving the accuracy of automated error detection systems. These systems use a variety of machine learning techniques such as neural networks, support vector machines, decision trees, and others to learn from large amounts of text data and develop models that can detect and correct grammatical errors in new text. These models can be trained on large corpora of text in multiple languages, allowing them to be highly adaptable to different writing styles and genres. Machine learning-based grammar checkers have shown promising results in detecting and correcting a wide range of grammatical errors, including spelling, punctuation, sentence structure, and word choice. However, these systems also face some challenges, such as the need for large amounts of training data, the potential for overfitting to the training data, and the difficulty of interpreting and explaining the decisions made by the model. Nonetheless, machine learning-based grammar checkers have the potential to significantly improve the accuracy and effectiveness of automated error detection systems, making them a promising area of research and development. Machine learning-based grammar checkers have been developed for a wide range of languages, including but not limited to: English, French, Spanish, German, Portuguese, Danish, Swedish, Urdu, Korean, Indonesian, Afan Oromo, Nepali, Amharic etc.

This list is not exhaustive and there may be other languages for which machine learning-based grammar checkers have been developed.

Existing system and its shortcomings:

Singh and Lehal (2008) created a Punjabi grammar checker using a rule-based approach. To achieve this, they developed various components for the grammar checker, which include a pre-processing unit, morphological analyzer, Part of Speech (POS) tagger, Phrase Chunker, and an error detection and suggestion system. The morphological analyzer was developed using a full-form lexicon-based approach, while the part of speech tagger was developed using a rule-based technique. The phrase chunker was also developed using a rule-based approach, and finally, for error detection, another rule-based approach was utilized. Some of the major shortcomings (Research gap) of existing system includes checking of simple sentences, morph dependent etc.

Proposed methodology:

As mentioned above, there are lot of drawbacks of the existing system. This is due to rule based approach in which fixed rules are used to develop the system. Therefore only those sentences will be checked properly that fall under the defined rules. But it is not feasible to develop exhaustive number of rules to handle all possible types of situation. By keeping above things in mind, we proposed a statistics based approach in which Hidden Markov Model is used to detect and correct the grammatical errors in a sentence. This model has been successfully implemented in development of various natural language processing applications like part of speech tagger, speech recognition, sentence segmentation, grapheme to phoneme conversion, partial parsing and chunking, named entity recognition and information extraction, spell checker, morphological analyzer, estimating POS tags of Unknown words, error correction and detection systems etc.

Details of the corpus used to Implement Hidden Markov Model:

The implementation of Hidden Markov Model relies on the presence of annotated corpus, with the corpus size being dependent on the number of tags used for annotation. As the number of tags increases, the corpus size also increases to ensure that each tag appears at least once for effective training and to avoid sparseness. A larger corpus results in higher accuracy, but it is important to ensure the accuracy and minimal unknown words in the corpus. Therefore, the author of this research utilized a standard annotated corpus from the ILCI website, which can be accessed at https://www.tdil-dc.in/index.php?option=com_download&task=fsearch&lang=en&limitstart=20&limit=5.

Testing and Results:

The system that was created was evaluated through manual testing on three different types of data sets. The first data set consisted of test papers and notebooks from students in grades 5 to 9 who were learning Punjabi as a second language. The second data set was obtained from ILCI sentences from the Punjabi corpus, with some sentences having errors manually introduced into them. The third data set was composed of test sentences that were manually created. Tables 2.1 and 2.2 present the details of the corpus that was used, as well as the results that were obtained from testing the system on these corpora.

Table 2.1: details of the corpus used for testing the HMM based grammar checker

Type of corpus	Total No. of Input sentences in the corpus	No. of sentences having Syntax error (In-correct sentences)	No. of in-correct sentences identified as incorrect by proposed algorithm	No. of in-correct sentences identified as correct by proposed algorithm
From note book of students learning Punjabi as second language	1000	411	401	10
ILCI sentences	1000	20	17	03
Manually developed corpus	200	200	198	02

Table 2.2: Result obtained after testing the HMM based grammar checker

Actual number of in-correct sentences in the corpus (A)	Correctly identified in-correct sentences (B)	In-correctly identified incorrect sentences (C)	Precision $\frac{\boldsymbol{B+C}}{\boldsymbol{A}}$ X 100	Recall $\frac{\boldsymbol{B}}{\boldsymbol{A}}$ X 100	F-measure $\frac{\boldsymbol{Preciion\ X\ Recall}}{\boldsymbol{Precision+Recall}}$ X 2
411	401	10	100	97.5	98.7
20	17	3	100	85	91.9
200	198	2	100	99	99.5

As shown in table 2.2, HMM based grammar checker system after testing on three data sets shows an overall precision of 100, recall 93.8 and f-measure as 96.7. Further on analyzing the individual accuracy of three types of test data sets, it can be observed that manually created error data set shows maximum accuracy (99% recall). This is because the error sentences present in this data set contains the errors for which this system has been developed. Least accuracy is shown by the dataset having sentences from ILCI corpus and the errors were incorporated manually. This is due to the reason that almost all the sentences of the ILCI dataset are correct and the error introduces to 20 sentences was random errors. Further this system is compared with existing rule based system and same dataset is used for testing the rule based grammar checker. As shown in table 3, rule based grammar checker on tested shown an average precision as 91.43, recall as 80.53 and f-measure as 84.96. Although the differences in the values of precision, recall and f-measure between the rule based and HMM based grammar checker is very small but still it is significant and it can be further improved by adding more data in the training dataset used for generating emission and transition datasets. Figure 3 shows the comparative graphical representation of precision, recall and f-measure obtained by testing the rule based grammar checker and HMM based grammar checker.

Conclusion and Future Scope:

In this research work, author has proposed a Hidden Markov Model (HMM) based grammatical error detection system and he further tested this system on Punjabi language. In this research work, author developed emission and transition probability datasets and implemented the Hidden Markov Model using viterby algorithm. The complete system is developed using C#.net. Further test data for testing the system is also created by the researcher. After testing the system author observed that the HMM based system performs better as compare to rule based system for detection of in-correct sentences. As shown in table 2.1 and 2.2, author tested this system on three types of data sets and claimed an overall precision of 100, recall 93.8 and f-measure as 96.7. Further, in this proposed system, the work has been done only for the detection of incorrect sentences. As the detection of error only does not provides the complete solution for the grammar checker and suggestions for correction must also be provided. Hence this work can be further extended to provide suggestion for the correction of incorrect sentences. Further the same algorithm can be tried for other Indian languages that lie in the group of Indo-Aryan languages.

References:

[1]. B. Behera and P. Bhattacharyya, “Automated Grammar Correction Using Hierarchical Phrase-Based Statistical Machine Translation,” Proc. Sixth Int. Jt. Conf. Nat. Lang. Process., pp. 937–941, 2013.

[2]. Holan, T., Kubon, V., & Plátek, M. (1997, March). A prototype of a grammar checker for Czech. In Fifth Conference on Applied Natural Language Processing (pp. 147-154).

[3]. Atwell, E. S. “How to detect grammatical errors in a text without parsing it,” In Proceedings of the third conference on European chapter of the Association for Computational Linguistics, pp38-45,1987

[4]. J. Eeg-olofsson and O. Knutsson, “Automatic Grammar Checking for Second Language Learners – the Use of Prepositions,” 2001.

[5]. G. Fliedner, “A System for Checking NP Agreement in German Texts Correct : seinen Argumenten,” no. July, pp. 12–17, 2002.

[6]. K. M. A. Hasan, A. Mondal, and A. Saha, “Recognizing Bangla Grammar Using Predictive Parser,” vol. 3, no. 6, pp. 61–73, 2011.

[7]. S. S. Hashemi, Automatic Detection of Grammar Errors in Primary School Children’s Texts. 2003.

[8]. S. Hein, “A Chart-Based Framework for Grammar Checking Initial Studies,” no. 1996, pp. 1–12, 1998.

[9]. Kubon, V., & Plátek, M. (1994). A grammar based approach to a grammar checking of free word order languages. In COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics.

[10]. Paggio, P. (2000, April). Spelling and grammar correction for Danish in SCARRIE. In Proceedings of the sixth conference on applied natural language processing (pp. 255-261). Association for Computational Linguistics.

[11]. Park, J. C., Palmer, M. S., & Washburn, C. (1997, March). An English Grammar Checker as a Writing Aid for Students of English as a Second Language. In ANLP (Vol. 24, No. 10.3115, pp. 974281-974296).

[12]. Tschichold, C., Bodmer, F., Cornu, E., Grosjean, F., Grosjean, L., Kübler, N. & Tschumi, C. (1997). Developing a new grammar checker for English as a second language. In From Research to Commercial Applications: Making NLP Work in Practice.

[13]. T. Vosse and P. O. Box, “Detecting and Correcting Morpho-syntactic Errors in Real Texts Nijmegen Institute for Cognition and Information University of Nijmegen Cognitive Technology Foundation,” pp. 111–118, 1990.

[14]. Young-soog, “Improvement of Korean Proofreading System Using Corpus and Collocation Rules,” pp. 328–333, 1998.

[15]. M. S. Gill, “A Grammar Checking System for Punjabi,” no. August, pp. 149–152, 2008.

[16]. Schmidt-Wigger, “Grammar and Style Checking for German,” Proc. Second Int. Work. Control Lang. Appl. CLAW1998, no. Ii, pp. 76–86, 1998.
[17]. L. Bopche and G. Dhopavakar, “Rule Based Grammar Checking System for Hindi,” vol. 3, no. 1, pp. 45–47, 2012.
[18]. Arppe, “Developing a grammar checker for Swedish,” 12th Nord. Conf. Comput. Linguist. pp. 13–27, 2000.
[19]. R. Bustamante and F. S. León, “GramCheck: A Grammar and Style Checker,” 1996.
[20]. J. Carlberger and R. Domeij, “A Swedish Grammar Checker,” 2002.
[21]. H. Kabir, S. Nayyer, J. Zaman, and S. Hussain, “Two Pass Parsing Implementation for an Urdu Grammar Checker.”