Rule Based Grammar Checking Systems (A Survey)

Sanjeev Kumar Sharma
DAV University Jalandhar, India

ABSTRACT

In this research article, a brief introduction to rule based technique used by different researchers in developing typical grammar checking systems is provided. There are many researchers who worked on development of a grammar checking system. Distinct approaches have been used by different researchers. Some researchers used syntax based approach, some used rule based approach and other followed the statistical based approach.

1. INTRODUCTION

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.

2. RULE BASED TECHNIQUE

This is a language dependent technique. To implement this technique, a large number of hand crafted rules are required. The disadvantage of this approach is that it is language specific and an exhaustive number of rules have to be developed; which are of no use for other languages. Some of the advantages of this technique are that the developed rules can be edited, new rules can be added and existing rules can be deleted. This rule based approach has been successfully implemented on a number of languages like English, Afan Oromo (language widely spoken and used in Ethiopia), Punjabi, Tagalog Filipino (the official language of the Philippines), Chinese, Persian, Malay, Bangla etc.

- A rule based grammar checking system to detect morpho-syntactic errors was developed for Dutch language by Vosse (1992)[1]. The errors covered by this system includes homophonous words (words having same pronunciation but different spellings), homophonous words that differed only in their inflection, agreement errors, repeated words, and errors in idiomatic expressions.
- Another grammar checker for free word languages like free word order languages – Czech and Bulgarian was developed as a part of the Language Technology for Slavic Languages (LATESLAV) project by Kuboň and Plátek (1994)[2]. This grammar checker is based upon the idea of reducing the size and complexity of sentence by deleting those words from input that do caused any error. Another improved version of this system was developed by Holan et al. (1997)[3].
- A grammar and style checking system for simplified English text was developed by Adriaens (1994)[4]. Around 150 rules organized into four major categories for text, syntax, lexical use, and punctuation related errors were used. Two types of rules i.e. the rules that reported an error that the system was certain about and should be corrected, and the rules that detect some weakness that might be corrected, if possible were used.
- Another rule based grammar checking system for Swedish language was developed by Hein (1998)[5]. In this system, local error rules were used to detect structural and non-structural errors. This system was composed of two components a parser and a chart scanner. The input text was passed through the parser and the parser generated a chart. That chart was then fed to chart scanner to identify any error related to feature violation.
- A rule based style and grammar checking system was developed for technical documentation written in German language by Schmidt-Wigger (1998). As this system was designed to work on technical documentation texts, therefore, full parse was not attempted and instead, the error rules worked as simple pattern matching rules on the morphologically analyzed text in feature bundle format.
- Text-critiquing system (**CRITIQUE**) to detect the grammar errors and style weaknesses in English texts was proposed by Ravin (1998)[6] and was developed at IBM Thomas J. Watson Research Center. Style checking was performed only after the grammar checking. Parser containing about 200 phrase structure rules was used for syntactic analysis of sentences in the input. The style component of CRITIQUE had more than 300 phrase structure rules to detect style weaknesses.
- A grammar checker for Korean language was developed by Young-Soog (1998)[7]. Partial parsing was used to detect grammatical errors. Since Korean is a partially free word order language, therefore, the grammar used for parser is dependency grammar. A correction rule table was used to suggest corrections. In order to prevent excessive creation of candidate words for error replacement, this system used a high frequency word dictionary derived from corpus and part-of-speech pattern. The system was reported to have achieved an average precision of 99.05% and an average recall of 95.98%.
- A grammar corrector for Danish was developed by Paggio (2000)[8]. A full parser was used to detect the errors. The grammar of the parser was an augmented context-free grammar consisting of rewrite rules where symbols were associated with features. It used error rules to detect structural errors. The error rules contained error messages and error weight associated with them. In this way, if a particular error rule detected an error in the text then it could show a useful message to help the user correct the error. When this system was tested against a test corpus (having grammatical errors mixed in randomly chosen text), this system reached 58.1% error coverage as compared to 53.5% for Microsoft Word, the precision reported was 20.6% for this system and 15.9% for Microsoft Word on that same test corpus.

- A commercial grammar checking system for integration in Microsoft Word for French, German, and Spanish languages was developed by Helfrich and Music (2000)[9] at Microsoft Corporation. This system excluded some obvious errors that are simple to detect but most users don't bother about them. The authors presented the design process used to find out the features or errors that the grammar checker needed to cover and then in evaluation point out that how important it is to keep false alarms close to zero. They suggested the use of highly edited documents for testing to achieve false alarm count close to zero. As this system was commercial software, therefore, the inner details of the system's working were not presented.
- Another grammar checking system to be used as a part of project on developing a Computer Assisted Language Learning (CALL) system for French as a foreign language was developed by Vandeventer (2001)[10]. Government and binding theory based syntactic parsing system for French – FIPS was used to develop this system. This grammar checking system worked by relaxing three constraints for agreement – gender, number, and person. The parser used was based on chart parsing algorithm and returned partial parses in the form of chunks for the sentences that failed to go through complete analysis.
- Another grammar checking system for Swedish was provided by Carlberger et al. (2002, 2004)[11]. This grammar checking system combined both the probabilistic and rule-based methods to achieve high efficiency and robustness. Error rules were used to detect various grammatical errors and to give suggestions. HMM (Hidden Markov Models) based part-of-speech tagger is used in this system. This system had 200 scrutinizing rules and 50 help rules.
- A grammar checker to detect agreement error in noun phrase of German texts was developed by Fliedner (2002)[12]. A finite state automata based shallow parsing along with constraint relaxation was used to detect agreement errors in noun phrases. All the words in a noun phrase needed to agree in terms of number, gender and case. The precision and recall of this system was around 67%.
- A grammar checker for second language learners of Swedish was discussed by Kann (2002)[13] and Bigert et al. (2004)[14]. This system was an extension of Granska system developed by Carlberger et al. (2002)[16] and used a hybrid approach. In hybrid approach, a combination of three approaches – manually constructed error rules, based on POS trigram frequencies from a tagged corpus, and machine learning of automatically constructed errors was used. As all these approaches are focused on detecting fairly different set of errors, so a combination of these approaches gave better results.
- A grammar checking system based on two pass parsing approach for Urdu was presented by Kabir et al. (2002)[18]. In the first pass, the sentence was parsed using basic phrase structure grammar rules and if it failed to get completely parsed, then movement rules were applied to convert the sentence into its desired base form and then reparsed to check for errors. Movement rules were used to convert the input sentence into the form recognized as base form. If the input sentence failed to get parsed in the first pass and also no movement rules could be applied then it meant that the structure of the sentence was probably incorrect, thus, a structural error was flagged. The phrase structure grammar rules were designed only for base structure forms or kernel sentences. Only simple declarative sentences in subject, object, and verb (SOV) order were taken into consideration for grammar checking. The grammatical errors covered by the system were disagreement in terms of number, gender and case, internal to noun phrases and between noun and verb phrases in a sentence. Structural errors covered were missing noun, missing verb phrase, misplaced adjective phrase etc. For any detected error, the system provided corrections and showed the final corrected output to the user.
- A purely rule-based open source grammar and style checker for English was discussed by Naber (2003). QTAG (a freely available probabilistic part-of-speech tagger for non-commercial use, described by Tufis and Manson (1998)) was used for part-of-speech tagging along with a rule-based module to help the tagger by eliminating some of the ambiguous tags before sending it to the tagger. The rule-based module was added to the tagger as it has manually developed rules, which could be blocked, edited or new rule could be added. The other reason for this was that the incorrect results of the probabilistic taggers were difficult to interpret, as they depended completely on the training corpus used. POS tagset used by this system was BNC C5 tagset. A rule-based phrase chunking was used, i.e. a set of rules were defined that described which POS tag sequences would constitute a phrase. It then applied manually developed grammar checking rules on the POS tagged and phrase chunked text. Pattern matching grammar checking rules were used with patterns designed to match a sequence of words, POS tags, or chunk tags. If such a pattern was found in the input text, the input is termed as erroneous. An error message was displayed explaining what was wrong in the input, suggestions (if possible) to correct the error and example sentences displaying an incorrect and a correct sentence, for the particular error. There were 54 grammar rules, 81 false friend pairs, 5 style rules, and 4 built-in Python rules in this style and grammar checker.
- **FiniteCheck**, a grammar checking system for detecting errors in primary school children's texts written in the Swedish language was developed by Hashemi (2003). A finite state approach was used by this system and it had only positive rules and no rules described the error structures to be detected. No part of speech tagger for disambiguating POS tagging information was used; rather it saved all the possible POS tags for words and disambiguates some of this information in parsing phase using filtering transducers. There are components of its grammar; a narrow grammar that accepted only grammatical patterns and a broad grammar with relaxed rules were used to parse both grammatical and ungrammatical structures. Those sentences or segments that could be parsed using broad grammar but not narrow grammar were marked as erroneous. This system found errors related to noun phrase agreement, and use of finite and non-finite verb forms in main and subordinate clauses.
- Another rule based system for English text was proposed by Rider (2005). In this system, both the manually constructed error rules and randomly generated rules were used for error detection in English texts. The manually constructed rules worked on the POS tagged text and if a match was found then that particular segment was marked as erroneous. The random rules generated were then tested on a corpus of correct English and all those rules that flagged errors in that corpus were removed from the set of error rules.

- A grammar checking system for Brazilian Portuguese language was proposed by Kinoshita et al. (2006). This system was developed for use in OpenOffice. Local error rules were applied to outputs of POS tagger and chunker. Structural error rules were applied on the outcome of grammatical relation finder (establishes subject, verb, and predicate relations).
- Another rule based grammar checking system for Nepali language was developed by Bal and Shrestha (2007). Various types of grammatical errors covered by this system were nominal and verbal agreement, structural errors (for clause and sentence structure). This system included a tokenizer, morphological analyzer, POS tagger, chunker/parser, syntax checker etc.
- A rule based grammar checker was developed for Persian language by Ehsan and Faili (2010). Hand crafted rules were applied on the tagged input text. Another grammar checking system for Arabic/Persian language was developed by Shaalan (2005). In this system, a rule based chart parser was used. Rules were developed to check the agreement of verb with particles. These rules were implemented in the form of constraints. Therefore, if a particular constraint is not satisfied, the system will generate an error message.
- A rule based grammar checker for Afan Oromo (language widely spoken and used in Ethiopia) was developed by Tesfaye (2011). A set of 123 hand crafted rules was constructed. The set contained the rules related to match the grammatical agreement between subject and verb, subject and adjective, main verb and subordinate verb in terms of number, gender and tense. The system showed an overall precision of 88.89% and a recall of 80%.
- A rule based Chinese grammar checker was developed byJiang et al. (2011). A number of hand crafted rules were developed. Some of these rules were related with the misuse of quantifier and particle. Some others were used to check the mismatch between various word classes like mismatch between verb and object.
- A rule based grammar checking system for Malay language was proposed by Kasbon et al. (2011). The Tatbahasa Devan corpus was used to obtain the rules of Malay language. Before performing the grammar checking, the system performed two additional tasks.
- A rule based grammar checker for Punjabi language was developed by Singh and Lehal (2008). An exhaustive set of hand crafted rules were created and the input sentences were checked against these rules. These rules were designed to check the grammatical agreement between subject and verb, noun and its modifier etc. in terms of number, gender and case. Many other components like pre-processor, morphological analyzer, POS tagger, phrase chunker etc. were also developed. They used agreement matching techniques for grammar checking.

REFERENCES

[1]. Vosse, T. 1992. Detecting and correcting morpho-syntactic errors in real texts. In *Proceedings of the third conference on applied natural language processing*. Association for Computational Linguistics. pp. 111-118

[2]. Kuboň, V., &Plátek, 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

[3]. Holan, T., Kuboň, V., &Plátek, M. 1997. A prototype of a grammar checker for Czech. In *Proceedings of the fifth conference on applied natural language processing*. Association for Computational Linguistics. pp. 147-154

[4]. Adriaens, G. 1994. The LRE SECC Project: Simplified English Grammar and Style Correction in an MT Framework. In *Language engineering convention* pp. 1-8.

[5]. 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.

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

[7]. 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.

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

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

[10]. Helfrich, A., & Music, B. 2000. Design and evaluation of grammar checkers in multiple languages. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. pp. 1036-1040

[11]. Vandeventer, A. 2001. Creating a grammar checker for CALL by constraint relaxation: a feasibility study. *ReCALL*, *13*(01), pp. 110-120.

[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]. Fliedner, G. 2002. A system for checking NP agreement in German texts. In *Proceedings of the ACL Student Research Workshop*. pp. 12-17.

[15]. Kann, V. 2000. CrossCheck–a grammar checker for second language writers of Swedish.

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

[17]. 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).

[18]. 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.

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

[20]. Hashemi, S. S. 2007. Ambiguity resolution by reordering rules in text containing errors. In *Proceedings of the 10th International Conference on Parsing Technologies*. Association for Computational Linguistics. pp. 69-79.

[21]. SofkovaHashemi, S., 2003. *Automatic Detection of Grammar Errors in Primary School Children's Texts. A Finite State Approach*. Göteborg University,

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

[23]. 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*.

[24]. Bal, B. K., & Shrestha, P. 2007. Architectural and System Design of the Nepali Grammar Checker. PAN Localization Working Paper.

[25]. 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

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

[27]. 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

[28]. 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.

[29]. 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.