An Automated error detection system for Indian Language Using Statistical Approach

[1] Misha Mittal, [2] Vikas Verma, [3] S.K.Sharma

[1] *Assistant Professor, Department of Computer Science and Applications, Maharishi Markandeshvar Engineering college, Mullana, Ambala.*

[2] *Research Scholar, Department of Computer Science and Applications,DAV University, Jalandhar, India.*

[3] *Associate Professor, Department of Computer Science and Applications,DAV University, Jalandhar, India.*

[1] Mittalmisha14@gmail.com, [2] vikas2005verma@yahoo.co.in, [3] sanju3916@rediffmail.com

ABSTRACT

Grammatical error detection system also called grammar checker or syntactic analyzer is one of the advance tool for natural language processing. This tool plays an important role in proof reading and for development of many other natural language processing applications like machine translation, summarization, question answering system etc. In this research article, we proposed a framework for detection of grammatical error using statistical approach. Further in statistical approach, we used N-gram approach for detection of the grammatical errors. Corpus used for generation of n-grams is taken from Indian Languages Corpora Initiative. This corpus is annotated by using morphological analyzer followed by part of speech tagger. Bi-gram, tri-gram and quad gram of part of speech tags are generated by using the annotated corpus. On testing the proposed algorithm on self-generated test data for Punjabi language, Overall accuracy was 100 percent, recall was 87.2, and the f-measure was 93.16, according to us.

Keywords**:** Error detection system, NLP, N-gram, Syntactic Analyzer, Morphological analyzer, POS tagger.

INTRODUCTION

Natural language processing (NLP) is one of the sub-branch of Artificial Intelligence. NLP is the domain that deals with designing and development of various language processing applications or tools. These applications may include basic applications like morphological analyzer (MA), part of speech (POS) tagger, Spell checker, machine translation systems, machine transliteration systems, question answering systems, dialogue processing systems, summarization systems, error detection and correction systems, clause identification system, sentence identification system, voice recognition systems, voice generation systems and many more. All the tools developed for language processing can be used either independently or in combination with each other to perform bigger task like grammar checker. Error detection system is such a language processing system that is used for identification of grammatical mistakes in the written

text. In other words, we can also say that error detection system is a language compiler that detects the correctness of sentence as per the grammatical rules of the language in which it is written. This system cannot be developed without the help of basic language processing tools like morphological analyzer and part of speech tagger. Since the error detection system is language dependent, therefore, it is the need of the time to develop such algorithm that could be applied on all the languages after applying minimum changes. The application of error detection system lie in the field of proof reading, machine translation system, question answering systems, for students learning grammar of language and many more. In this research paper, we have developed an error detection framework that can be implemented for all Indian languages with a change of corpus. We tested this framework on Punjabi language (Regional language of India). Punjabi is mostly used in the north regions of India. Punjabi is spoken as the primary language in the state of Punjab and most regions of Haryana. It is spoken by 100 million persons in India. Other than India, Punjabi language is spoken by many migrated Indian in Canada, UK, Australia and many other countries. Punjabi is also spoken in the neighbor countries like Pakistan.

EXISTING WORK

Development of automatic grammar checking system is one of the challenging task in the field of natural language processing. Many commercial automated grammar checkers are available in the market. Most commonly used grammar checker are Grammarly, Grammarix and grammar checker embedded in various versions of Microsoft word etc. For the development of error detection system various methods have been tried. These methods includes syntax based approach, rule based approach and statistics based approach. Some also tried machine learning techniques. Various syntax based methods are used by Bernth (1997) [1], Hein (1998) [8], Ravin (1998) [10], Young-soog (1998) [11], Martin et al. (1998) [2] and Kabir et al. (2002) [14]. Statistics based methods are used by Alam et al. (2006) [3], Carlberger et al. (2002, 2004) [12-13], Ehsan and Faili (2010) [5], Temesgen and Assabie (2012)[6], Kinoshitaet al. (2006) [21] and Verena Henrich and Timo Reuter (2009) [7]. Similarly rule based method is used by Schmidt-wigger (1998) [9]. Kann (2002) [4], Naber (2003) [15], Rider (2005) [16], Faili (2010) [5], Tesfaye (2011) [17], Jiang et al. (2011) [18], Kasbon et al. (2011) [19], Singh and Lehal (2008) [20], Bal and Shrestha (2007). In case of machine learning approaches, Ghosh et.al. [25] used neural network networks for text correction and completion in keyboard decoding for English language, Smith [26] developed grammar inference for detection of grammatical mistakes using special type of neural networks i.e. RNN, further Huang et.al. [27] tried memory based neural networks i.e. long short term memory (LSTM) for development of grammar checker system, in the same way Gudmundsson et. al. [29] used LSTM for development of spell and grammar checker for Swedish language, Lewis [28] used RNN to develop error detection system.

N-GRAM BASED APPROACH

This is a statistical method and is widely used for developing application for natural language processing. It is also used by [3] and [23] for detection of grammatical errors in Bangla and English languages. Under this method, first of all annotated corpus is

collected or generated. From this annotated corpus, n-gram patterns are generated. The correctness of input sentences is then checked against these n-gram patterns. Generally, one error at a time is corrected by using this technique. Using n-gram the error detection problem can be stated as:

> Given a sentence S = w1, w2, …, wn, and web-scale n-gram, webgram. Our goal is to train two statistical machine translation model TM and back-off model TMbo to correct learners' writing. At run-time, trigrams (wi , wi+1, wi+2) in S (i =1, n-2) are matched and replaced using TM and the back-off model TMbo to translate S into a correct sentence T.

PROPOSED METHODOLOGY

In this research article, we used N-gram method for detection of grammatical errors in the sentence. The corpus used for generation of n-grams is taken from Indian Languages Corpora Initiative (ILCI). The details of the corpus is provided in table 1.

Type of Corpus/ Domain of the corpus	Number of files	Total Number of Sentences inthe file
Agriculture	20	202568
Entertainment	20	137118
Tourism	19	31435
Health	25	51326
Total	84	422447

Table (1)- Details of corpus taken from ILCI

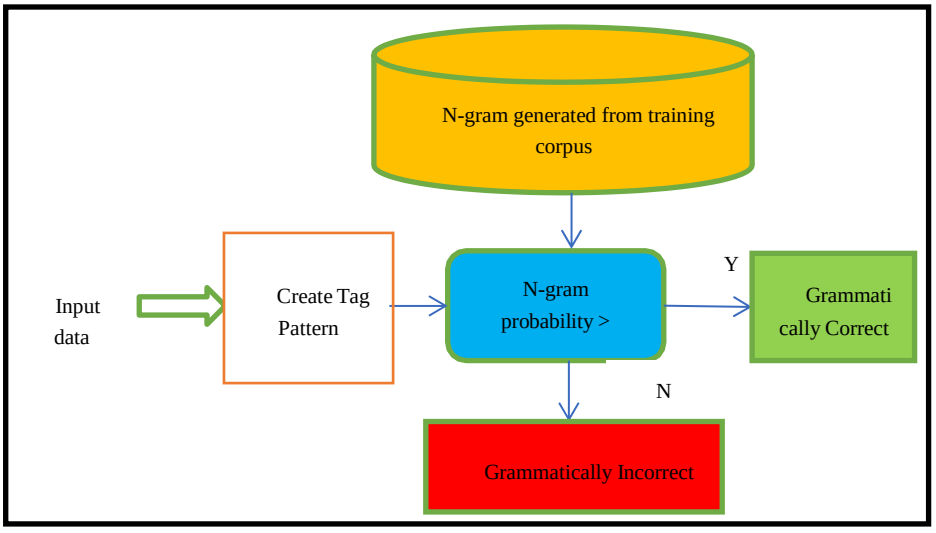

Fig. 2 - Proposed architecture for development of n-gram based error detection system for Punjabilanguage

As shown in the diagram 2, the overall process of error detection takes place in two steps. In the first step, preprocessing is performed. In pre-processing, the corpus is passed through morphological analyzer where each and every word of the corpus is assigned a part of speech tag containing the grammatical information of that word. The tag set used was designed by [24] especially for development of automatic grammar checking system for Punjabi language. Since Hindi and Punjabi languages are very similar in nature and also both of these languages are morphologically rich in nature, therefore tag set designed for grammar checking of Hindi language can be used for the error detection in Punjabi language sentences. The tag set used reflects the word class of the word along with its number, gender and other related information like case, transitivity and tense etc. We developed the morphological analyzer by using the full form lexicon technique. After passing through the morph, self-developed HMM based part of speech tagger is applied to resolve the problem of ambiguity. Thus after passing through the POS tagger annotated corpus is ready for generating n-grams. To calculate the n-gram probabilities following formulas are used:

Creation of N-grams probabilities:
N-grams (bigram, trigram and quad-gram) are generated from the ILCI corpus.
Probability bi-gram = count(ti_ti+1)/ Total number bi_gram
Probability of tri-gram = count(t_i_t_{i+1}_t_{i+2})/Total number of tri-grams Probability of quad-gram = count(t_i_t_{i+1}_t_{i+2}_ t_{i+3})/Total number of quad-grams

In the second phase, grammar checking is performed. In this step, input sentence is passed through morph and part of speech tagger. After that POS patterns of the input sentence are generated from the tags associated with each word of the sentence. This POS pattern is then checked for error detection using n-gram probabilities generated in step 1Validation and discussion

This is the first time that n-gram approach has been used for error detection of Punjabi language. Before this n-gram was used for grammar checking of Bangla and English languages [3], but the results were not up to the mark. When n-gram technique used for English, the system shows performance is 63% (when tested give 545 sentences as correct, out of 866 correct sentences). In case of Bangla language, the performance of the system was 53.7%. (When tested give 203 as correct sentences out of 378 correct sentences). Another system using n-gram was developed by [23] for English language and claimed an overall Precision of 64.58, Recall 47.69 and F-measure as 54.86. Although a literature on rule based grammar checking system for Punjabi language is also observed [22] but no information about the accuracy is mentioned by the author. As shown in table…it is observed that the author obtained better results using n-gram approach for Punjabi language as compare to the Bangla and English languages.

RESULTS

The proposed method is tested on test data set. All tests are conducted on a standard computer with an Intel Core Duo processor at 2.0 GHz and 2 GB of memory. The program runs on window operation system with. XML is used as database for storing the n-grams. The test dataset is generated in three sets and is shown in figure 3. As clear

from the figure 3, the first set is developed by randomly collecting the corpus from ILCI and inserting errors manually. Second set is generated by collecting the sentences from online sources like e-papers and other websites containing stories and novels and by manually adding the error in the sentences. The third data set is generated from notebooks of school children (4th to 7th standard) studying Punjabi as second language. After testing the system on the test data sets, the results obtained are shown in table 2 and figure 4. As clear from table 2, total 1000 sentences are used for testing the system and the system shows a 100% precision for all three types of data sets. Maximum recall (91.2%) is shown on the dataset created from the note books of school students. High precision and recall of the system is because the test data is self-made. It is further observed that as the length of the input sentence increase the accuracy of the system decreases. This because to handle the long term dependencies we have to use the higher value of n.

CONCLUSION AND FUTURE SCOPE

In this research article, author used n-gram based statistical technique for detection of errors in the Punjabi sentences. Author used Indian Languages corpora initiative corpus for generation of n-grams. On testing the proposed algorithm on self-generated test data, overall accuracy was 100 percent, recall was 87.2 percent, and the f-measure was 93.16 percent, according to the author. Although the precision, recall and f-measure of the proposed system seems to be good, but this result is good for self-generated test corpus and this test corpus is generated by incorporating those errors for which this algorithm has been designed. This technique can be further improved by improving algorithm by adding more options for detection of more errors. This technique can be further improved by using HMM technique or by using combination of HMM with rule based technique. Further machine learning and deep learning based approaches can also be implemented for detection of grammatical errors in the Punjabi language.

REFERENCES

[1]. Bernth, A.: EasyEnglish: a tool for improving document quality. In: 5th Proceedings on Conference on Applied NLP natural language processing. ACL (Assoc. for Computational Linguistics), pp. 159-165. (1997).

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

[3]. Alam, M. J., Mumit, K., & Naushad, U.: N-gram based Statistical Grammar Checker for Bangla and English. In: 9th International Proc. on Computer and IT (ICCIT), (2006).

[4]. Bigert, J., Kann, V., Knutsson, O., & Sjobergh, J.: Swedish Grammar checking for second language learners, 33-47(2004).

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

[6]. Temesgen, A., & Assabie, Y.: Development of Grammar Checker for Amharic

Using Morphological Features of Words and N-Gram Based Probabilistic Methods, IWPT (2013).

[7]. Henrich, V.: LIS Grammar Checker: Statistical Language Independent Grammar Checking (Doctoral dissertation, Reykjavík Univ.) (2009).

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

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

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

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

[12]. Carlberger, J., Kann, V., Domeij, R., & Knutsson, O.: A grammar checker for Swedish. Submitted to Computational. Linguistics, oktober (2002).

[13]. Carlberger, J., Kann, V., Domeij, R., & Knutsson, O.: Swedish grammar checker development and performance: A language engineering perspective. Natural language engineering, 1(1) (2004).

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

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

[16]. Rider, Z.: POS tagging Grammar checking using rules matching. In: Proceedings of Conference on Class of 2005 on NLP Natural Language Processing. (2005).

[17]. Tesfaye, D.: An Afan Oromo Grammar rule-based Checker. IJACSA Editorial.(2011).

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

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

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

[21]. Kinoshita, J., Menezes, C. E. D., & Salvador, L. N.: CoGrOO: a Portuguese - Brazilian CETENFOLHA Corpus based Grammar checker. In: Proceedings of 5th international conference on LRE, Language Resources and Evaluation, LREC. (2006).

[22]. Bopche, L., Kshirsagar, M., & Dhopavkar, G.: Rule Based Morphological Process Grammar Checking System for an Indian Language. In: Proceedings of 4th International Conference on GTISSA, Global Trends in Information Systems and Software Applications. (2011).

[23]. Nazar, R., & Renau, I.: N-gram corpus grammar checker for Google books. In: Proceedings of 2nd Workshop on CLW, Computational Linguistics and Writing. Cognitive and Linguistic Aspects of Document Engineering and Document Creation. Association for Computational Linguistics, pp. 27-34. (2012).

[24]. Gill, M. S., Joshi, S. S., & Lehal, G. S.: POS Part of speech tagging for Punjabi grammar checking. The Linguistic Journal, *4*(1), 6-21(2009),

[25]. Ghosh, S., & Kristensson, P. O.: Text Correction using neural networks and completion in keyboard decoding, arXiv preprint arXiv: 1709.06429.(2017).

[26]. Smith, A.; Recurrent neural networks grammar inference. Department of Computer Sc., University of San Diego, California, www. cse. ucsd. edu/~ atsmith. (2003).

[27]. Huang, S., & Wang, H.:Bi-LSTM Chinese grammatical error diagnosis using neural networks. In: Proceedings of 3rd Workshop on NLP Natural Language Processing Techniques for Educational Applications (NLPTEA2016), pp. 148-154. (2016).

[28]. Lewis, G.: Recurrent Neural Networks and Sentence Correction. Department of Computer Sc., Stanford University. (2016).

[29]. Gudmundsson, J., & Menkes, F.: Natural Language Processing using Swedish using LSTM Long Short-term Memory Neural Networks: A ML-powered Grammar and Spell-checker for the Swedish Language. (2018).