EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

A Framework for Grammatical Error Detection and Correction System for Punjabi Language Using Stochastic Approach

L. Jindal[1,*], H. Singh[2], S. K. Sharma[3]

[1]SBBS University, Jalandhar, Punjab India
[2]Department of Computer science and Engineering, SBBS University, Jalandhar, Punjab, India
[3]Department of Computer Science and Applications, DAV University, Jalandhar, Punjab, India

Abstract

INTRODUCTION: In this modern era of internet and technology natural language processing task has emerged as one of the major research area in computer science. Grammatical error detection and correction system assists to detect and correct syntactic errors present in written text. OBJECTIVES: In this research article, author investigate the applicability of stochastic approach for the development of grammatical error detection and correction system for Punjabi language. METHOD: Author used corpus based stochastic approach to developed the system. The corpus used was taken from Indian language corpora initiative. RESULTS: On testing, the developed system shows a precision as 82.5%, recall as 89% .and f-measure as 85%. The results of the proposed system outperform the existing rule based system that shows precision of 76.79%, recall of 87.08%, and F-measure of 81.61%. CONCLUSION: author concluded that for syntax analysis stochastic approach can perform better than rule based approach.

Keyword: Punjabi GEC, syntactic analyzer, Grammar checker, HMM, stochastic

Received on 17 December 2020, accepted on 15 April 2021, published on 27 April 2021

doi: 10.4108/eai.27-4-2021.169421

*Corresponding author. Email:sanju3916@rediffmail.com

1. Introduction

Today, millions of people around the world are using Punjabi language for speaking and writing purpose. In fact, non-native Punjabi speakers currently outnumber than native speakers and their numbers will keep increasing in the future. Non-native Punjabi speakers usually make errors in written text, and further these errors are of various types according to their complexity. A practical grammatical error correction (GEC) system to correct errors in Punjabi text promises to benefit Millions of Punjabi language learners around the world. Further GEC has commercial perspective also i.e. there is a great potential for many other practical applications, such as proofreading tools that help non-native speakers to identify and correct their writing errors without human intervention or an educational software for automated language learning and assessment. One of the popular existing professional GEC systems is GRAMMARLY (used for detection and correction of spelling and grammar error in English language). The general architecture of GEC system can be viewed as shown in following figure1:

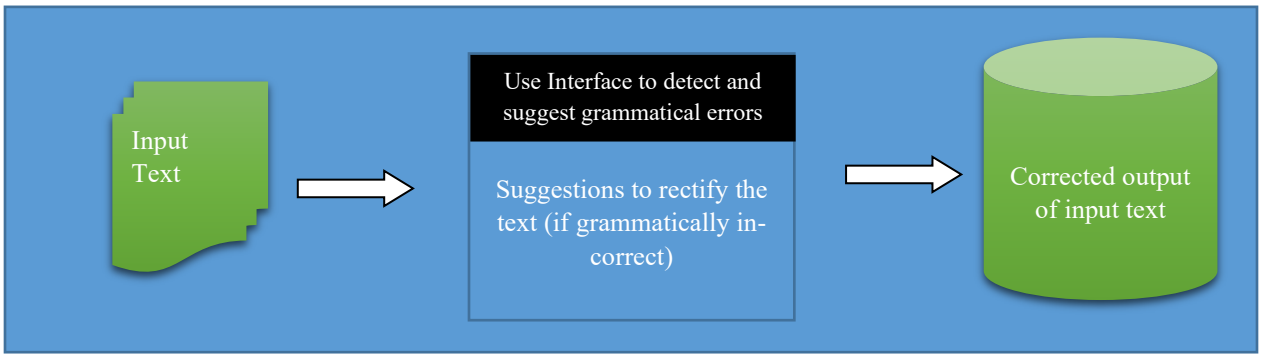

Figure 1. General representation of GEC

As shown in above figure1, the user will give input text in the form of paragraphs or sentence. The GEC system will detect the grammatical or syntactical correctness of input text as per the grammar rule of the language in which input text is written. If the input text is found correct then no error or suggestions will be provided otherwise, if the text is found grammatically incorrect then suggestions to rectify the errors will be provided by the system.

Punjabi is the official language of one of the state in India i.e. Punjab. There are approximately 125 million Punjabi speakers in India. Other than India, Punjabi is also spoken by a number of migrated peoples residing in Canada, USA, Australia and UK etc. Punjabi language is also used in Pakistan in written as well as in spoken form. Different scripts are used to write Punjabi language in India and Pakistan. The script used to write Punjabi in India is Gurmukhi while script used to write Punjabi in Pakistan is Shahmukhi. Center for technical development of Punjabi language had already developed a software called Sangam (Gurpreet Singh Lehal & Saini, 2014) that can convert Gurmukhi to Shahmukhi and vice versa. There are many organizations working on the technical development of Punjabi language. Main organizations working in this field are center for technical development of Punjabi language (Punjabi University Patiala), C-DAC Mohali and Thapar Institute of Engineering and Technology (TIET) Patiala. Besides these, researchers from TDIL (Technical development of Indian languages) and IIIT (International institute of information technology) Hyderabad are also working on technical development of Punjabi language. Some of the Punjabi language processing resources developed by these organizations include Punjabi spell checker (Dhanju, Lehal, Saini, & Kaur, 2015), Punjabi grammar checker (Gill, n.d.), Punjabi POS tagger (Adamson, 2009), Punjabi Morphological analyzer (Gill, 2007), Gurmukhi to Shahmukhi machine translation (Gurpreet Singh Lehal, 2009), Hindi to Punjabi machine translation (Goyal & Lehal, 2009), Punjabi to Hindi Machine translation system (Josan & Lehal, 2008), Punjabi Optical Character Recognition system (G. S. Lehal & Singh, 2002), Punjabi summarization (Gupta & Singh, 2012) etc.

2. Existing Work

As discussed in section1, various researchers and organizations are working on the development of natural language processing resources, but still a lot of work for development of GEC is in queue. After reviewing a number of literatures written by different authors it is observed that there are mainly rule based, classifier based and statistics based methods are used for GEC system development. Some of the observations from reviewed literatures is discussed in the following section.

2.1. Rule Based Approach

This is the oldest method used for development of GEC. In the beginning, simple pattern matching and string replacement techniques were used to implement rule based approach. Later on syntactic parsing using part of speech tagging, tree parsing and hand crafted rules were used (Heidorn, Jensen, Miller, Byrd, & Chodorow, 1982). The first grammar checking tools, such as the Unix Writer's Workbench (MacDonald et al., 1982) or EPISTLE and CRITIQUE (Heidorn et al., 1982), used hand-crafted rules and pattern matching techniques. The most widely used grammar checker nowadays from Microsoft Word text editor (Heidorn, 2000) relies mostly on a rule-based approach. Another recent example is LanguageTool2 (Miªkowski, 2010), which has been initially developed by Naber (2003).Further this approach is used by (Arppe et al., 1998; Baviskar & Scholar, 2019; Bopche &

Dhopavakar, 2012; Flachs, Lacroix, Rei, Yannakoudakis, & Søgaard, 2019; Kárason & Språkgranskningsverktyg, 2006; Megyesi, 1998; Naber, Kummert, Fakultät, & Witt, 2003; Poornima & Dhanalakshmi, 2011; Schmidt-Wigger, 1998; Science, 2017; Sidorov et al., 2013; Tesfaye, 2011). One of the disadvantage of the rule based system is that, most of the errors are complex and the rule-based systems fail to rectify those errors. Further it is not feasible to construct exhaustive set of rules to rectify all possible types of grammatical errors. Therefore, now in the development of most of the GEC systems, instead of employing only rule-based mechanism, stochastic or hybrid approach is preferred.

2.2. Classifier Based Approach

Now, because of easily availability of annotated corpus, various machine learning classifiers were developed to correct the incorrect sentences ((Han, Hall, Chodorow, & Leacock, 2008), (Rozovskaya, Tech, & Roth, 2016))). In this approach, GEC is simulated as a classification problem with multiple classifiers in which an incorrect candidate sentence may have multiple possible correct solutions. This approach is used by (Han et al., 2008) in which author trained a maximum entropy classifier to detect article errors and achieved an accuracy of 88%. Further (Tetreault & Chodorow, 2008) used maximum entropy models to correct errors for 34 common English prepositions in learner text. In this approach, one of the commonly used method is to build multiple classifiers, one for each error type and cascade them into a pipeline. Further a combination of rule-based and classifier models to build GEC systems (that can solve multiple errors) is tried by [23]. But the disadvantage of classifier approach is that, it can be applied to solve only those errors which are independent of each other and are unable to solve the dependent errors. The problem of dependent errors is solved by developing a system of multiple classifiers for a sentence containing dependent errors [27]. In addition (Dahlmeier & Ng, 2012) developed a beam-search decoder for correcting interacting errors.

2.3. Statistics Based Approach

Statistical approaches are tried by many researchers for the development of GEC. The main reason of using this approach was availability of digital data on the internet for training. Researchers used this digital text to train the system. Most of the statistical approaches are probability based in which various types of probabilities (e.g. transition, emission, n-gram etc.) from sequence of POS (part-of-speech) tags is calculated. The POS sequence of input text is evaluated against these probabilities and if they fall below some threshold values, then input sentence is considered as correct otherwise incorrect. Larger the annotated corpus more will be the accuracy of the system. Further the annotated corpus should be versatile i.e. it should cover as many different domains as possible. This approach also has some pitfalls as due to its statistical nature sometimes it provides unpredicted results, and it becomes difficult for the user to interpret these results. The main advantage of this approach is that it can be implemented on any natural language without the knowledge of the syntax of that language. First researcher to use this approach was Atwell, Eric Steven (Dahlmeier & Ng, 2012) in1987. After this various researchers like (Hasan, Mondal, & Saha, 2011; Moré, 2006; Ram & Fernando, 1985; Renau, 2012; Yuan & Kingdom, 2013) followed this approach. Further variants of this approach are shown in the figure 2.

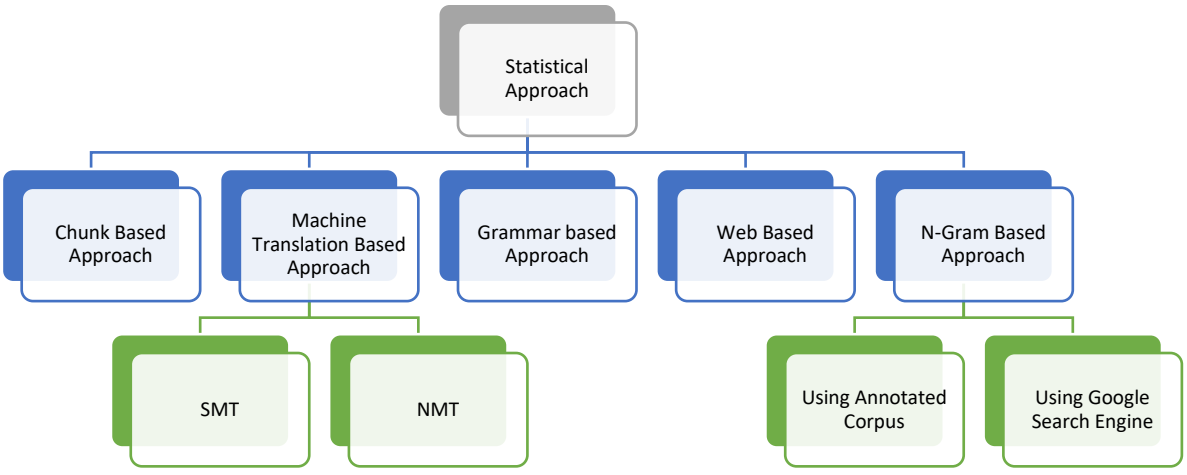

Figure 2. Various variants of statistical techniques used for grammar checking

As shown in figure 2, there are five major variants used to develop statistical GEC system. These approaches include chunk based used by (Lin & Soe, 2015), machine translation based used by (Brockett, Dolan, & Gamon, 2006), grammar based used by (Vladislav Kubofi and Martin Plaitek, 1994) , web based (Chen, 2009) and N-gram based (Alam, Uzzaman, & Khan, n.d.)(Renau, 2012). Machine translation based approaches have been further classified into Statistical Machine Translation (SMT) based approach and Neural Machine Translation (NMT) based approach. SMT model is used by (Brockett et al., 2006) to correct a set of 14 countable/uncountable noun errors made by learners of English language. Experiments show that their SMT system was generally able to beat the standard Microsoft Word 2003 grammar checker, although it produced a relatively higher rate of erroneous corrections. Further (Mizumoto, Komachi, Masaaki, Ntt, & Matsumoto, 2011) used this SMT based approach to develop Japanese language error detection system. Further the effect of training corpus size on various types of grammatical errors in English language is studied by (Mizumoto, Hayashibe, Komachi, Nagata, & Matsumoto, 2012) and concluded that a phrase-based SMT system is effective at correcting errors that can be identified by a local context, but less effective for correcting errors that need long-range contextual information. Another POS-factored SMT system is trained by (Yuan & Kingdom, 2013) to correct five types of grammatical errors (articles, prepositions, noun number, verb form, and subject-verb agreement). A combination of rule-based system and a phrase-based SMT system is proposed by (Felice, 2014). Another hybrid approach by combining MT and classifier model is developed by (Susanto, 2014). Another experiment to develop GEC is done by (Grundkiewicz, 2014) by employing word-level Levenshtein distance between source and target as a translation model feature. Further in this field, effect of f-score tuning on precision is studied by (Kunchukuttan, Chaudhury, & Bhattacharyya, 2014) and concluded that this will reduce the performance of the GEC. More recently, (Napoles & Callison-Burch, 2018) proposed a light weight approach to develop GEC called Specialized Machine translation for Error Correction (SMEC) which represents a single model that handles morphological changes, spelling corrections, and phrasal substitutions. Further, (Hermet, Edward, & Désilets, 2009) handled task on detection of preposition errors by generating a round-trip translation via French and their model identify 66.4% of errors. An all-errors task using round-trip translations obtained from the Google Translate API via eight different pivot languages is attempted by (Nitin, Tetreault, & Chodorow, 2012).

3. Stochastic and Statistical Techniques

Mathematical models can be classified into two broad categories (Edmondson, H. P. (1968)) i.e. deterministic and stochastic. A mathematical model will be stochastic if probability is involved otherwise it will be simple statistical model. If a mathematical model is stochastic, then it is reasonable to call the whole method stochastic. Now if we talk about the use of mathematical models in natural language processing then statistical term is more widely used as synonym to stochastic approach. Thus, in case of Natural language processing, one can say that a model or method is statistical (or stochastic) if it involves the concept of probability). Most commonly used statistical methods in natural language processing includes application methods that are used to solve an NLP problem P, by applying an algorithm A to a mathematical model M in order to solve an abstract problem Q approximating P, acquisition methods in which problem P is used to construct a model M that can be used in an application method for P, and evaluation methods in which problem P is used to evaluate application methods for P. stochastic techniques is used in topic modeling (Gao, W., Peng, M., Wang, H. et al ,2019), to identify domestic violence from online posts (S. Subramani, S. Michalska, H. Wang, J. Du, Y. Zhang and H. Shakeel, 2019), Measuring individuals' valuation distributions (H. Wang, D. Whittington, (2005)), Query optimization (Sharma, M., G. Singh, and R. Singh,2019), Sharma, M., et al (2015), rising number of COVID-19 cases (Sharma, M., and S. Sharma,2020), spam detection (Benczur, A. A., Csalogany, K., Sarlos, T., & Uher, M. (2005, May).), POS identification (Jassim, A. K., & Al_Bayaty, B. F. Z. (2021, February)), Morphhological analysis (Cheragui, M. A., & Hiri, E. (2020, February).)

4. Proposed Architecture

In this research, author experimented with stochastic approach to develop GEC system for Punjabi language. Author complete this work in two phases. In the first phase stochastic probabilities are calculated using ILCI annotated corpus and in the second phase, grammar of input sentence is checked for correction using the stochastic probabilities calculated in the first phase. The first phase is developed using single module (tag sequence probability calculations) while in the second phase, three modules (Preprocessing, pattern matching based error detection and grammatical error correction) are used. Figure3 shows the architecture of proposed GEC system. As shown in figure 3, there are basically two components of GEC. In the first component the probabilities of unique tag

sequences is calculated and in the second component, using these unique tag probabilities, error is detected in the input text. After detection of error, input text is rectified as per grammar agreement rules. Further details of the various components of proposed architecture are explained in the section 4.1 to 4.4.

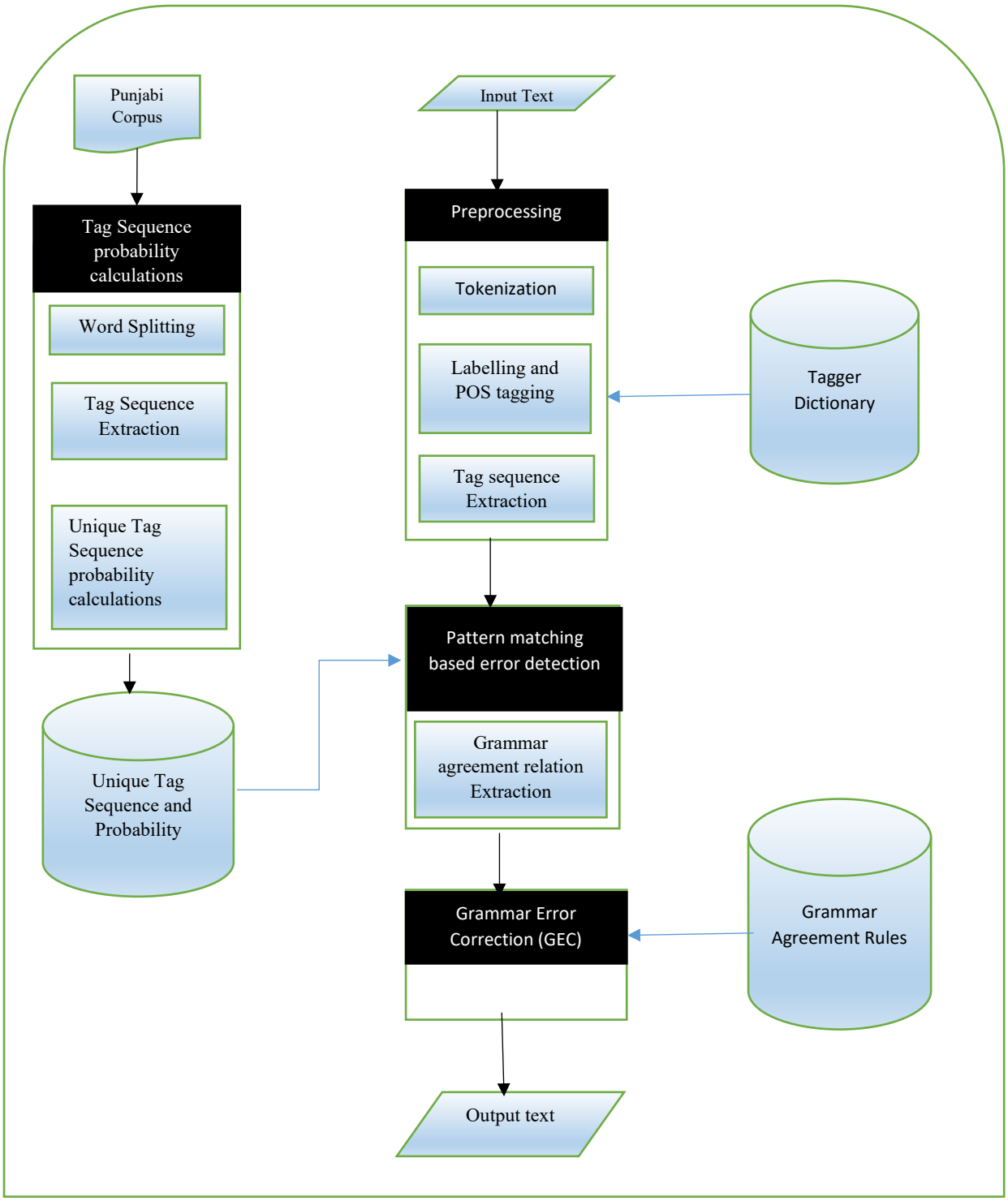

Figure 3. Proposed architecture of GEC

4.1. Annotated corpus used to calculate stochastic probabilities

As discussed above, the task of the first phase is to calculate the probability of tag sequences. In order to calculate these tag sequence probabilities annotated corpus of Punjabi language is required. The Punjabi annotated corpus used for this task is taken from Indian languages corpora Initiative (ILCI). This corpus includes data from various domains like sports news, agriculture, entertainment, tourism and health. Total 2, 64,474 number of sentences were taken to calculate the tag sequence probabilities. Further details of the annotated corpus are shown in following table 1.

Table 1. Details of the annoated Corpus used for calculating tag sequence probabilities

Type of Corpus/ Domain of the corpus	Number of files	Total Number of Sentences in the file	Sentences with Length 5 words	Sentences with Length 6 words	Sentences with Length more than 7 words
Agriculture	20	40258	99	213	372
Entertainment	20	137008	151	230	342
Tourism	19	37882	231	440	712
Health	25	49326	430	657	945
Total	84	264474	911	1540	2371

4.2. Phase 1 (Tag sequence probability calculation)

This is the first phase of this research work and this phase is completed in three steps (Word splitting, tag sequence extraction and unique tag sequence probability calculations). In word splitting, annotated training corpus (ILCI) is split in to list of tokens (individual words along with tags) and these tokens are stored in an array. Thereafter each individual token is processed to extract the tags from it. After extracting the tags from the tokens, these extracted tags are arranged to form tagged patterns. Some sample entries are shown below:

- _\N_NN_\N_NN_\N_NN_\PR_PRP_\N_NN_\V_VM
- _\N_NN_\PSP_\QT_QTC_\N_NN_\V_VM_\V_VAUX
- _\CC_CCD_\N_NN_\N_NN_\RP_INTF_\RB_\V_VM
- _\JJ_\N_NN_\PSP_\N_NN_\N_NN_\V_VM
- _\N_NN_\RP_INTF_\RP_RPD_\QT_QTF_\V_VM_\V_VAUX
- _\N_NN_\N_NN_\PSP_\N_NN_\V_VM_\V_VAUX
- _\N_NN_\N_NN_\N_NN_\PR_PRP_\N_NN_\V_VM
- _\N_NN_\PSP_\QT_QTC_\N_NN_\V_VM_\V_VAUX
- _\CC_CCD_\N_NN_\N_NN_\RP_INTF_\RB_\V_VM
- _\JJ_\N_NN_\PSP_\N_NN_\N_NN_\V_VM
- _\N_NN_\RP_INTF_\RP_RPD_\QT_QTF_\V_VM_\V_VAUX
- _\CC_CCD_\N_NN_\PRP_\RP_INTF_\RB_\V_VM
- _\JJ_\N_NN_\RD_\N_NN_\N_NN_\V_VM
- _\N_NN_\N_NN_\N_NN_\PR_PRP_\N_NN_\V_VM
- _\N_NN_\PSP_\QT_QTC_\N_NN_\V_VM_\V_VAUX
- _\JJ_\N_NN_\PSP_\N_NN_\N_NN_\V_VM

After generating these tag sequence, bigrams of each tag sequence is generated. After generating bigrams from all the tag sequences, probability of unique bigram tag sequence is calculated by using following formula:

Bi-gram Probability of tag i and tag j pair i.e.

$$P_{ij} = \frac{Number\ of\ time\ tagi\ and\ tagj\ pair\ occurs\ in\ the\ tagpairs}{Total\ number\ of\ tag\ pairs}$$

Some sample entries of bigram probabilities are shown in the table 2.

Table 2. Sample entries of bi-gram probabilities

Sr. No.	Tag sequence pair (bigrams)	Probability
1	N-NN_N-NN	0.124065
2	N-NN_CC	0.110202
3	CC_N-NN	0.038351
4	N-NN_JJ	0.143304
5	JJ_N-NN	0.219112
6	N-NN_PSP	0.733298
7	PSP_JJ	0.188384
8	JJ_V-VAUX	0.059797
9	JJ_JJ	0.128332
10	PSP_V-VM	0.14809
11	V-VM_N-NN	0.039182
12	N-NN_RP	0.392429
13	RP_JJ	0.089174
14	JJ_PSP	0.03885
15	JJ_RP	0.06919

4.3. Phase 2 (error detection):

In this phase, the input sentence entered by the user is checked against the bigram probabilities calculated in phase 1. In this phase, three modules are used. The first module is preprocessing, in which input text is split at sentence level followed by phrase level and in the last at word level till we get individual word as final token. However, if input text is in the form of paragraph, then the system will first split this paragraph into sentence then these sentences into tokens. In order to split the input text into tokens, special symbols are used as identifier other than tab space. These special symbols includes punctuation marks like comma (,), colon (:), question mark (?), semi-colon (;) and exclamation (!). After splitting, labeling is done. In labeling, all the individuals tokens separated in tokenization steps are assigned label as per morphology rules and tag tokens within their appropriate morphology based POS tag from the tagger dictionary. If a token is not present in the tagger dictionary then it will be assigned as "Unknown". The second module used in this phase is Pattern matching based error detection. In this module, various errors related with the mismatch of the agreement in an input sentence is detected. Various agreement errors detected by the system includes Subject-Verb, Object-Verb, Modifier-Noun, and Adverb-Verb grammar agreement errors. To identify these types of errors, probability of the tag pattern of input text is calculated.

4.4 Algorithm Used:

```
Read tagged_sentence
for i=1; i<tagged_sentence.count; i++
        N= tagged_sentence [i]
        If N is noun
                for m=i-1;
                    tagged_sentence[m]
                    is Adjective or Determiner
                    or Number or Pronoun;
                    m--
                    M=tagged_sentence[i-1]
                    Add N of tag + M of tag
                    into
                    Agreement_in_sentence
                End for
        end If
end for
return Agreement_in_sentence
```

As explained in above algorithm, input sentence is scanned from left to right and agreement between the noun and adjective or noun and determiner or noun and number is identified. If there is mismatched in the agreement then error message is displayed. The subject and verb agreement is done according to number, gender and person tag value of the subject and the verb. The subject of input sentence is identified from the tagged sentence having NN or PNP tags. Similarly agreement errors between modifier noun and noun adjective is identified.

4.4. Correction of error

After detection of error, last step of this second phase is the correction of detected error. This is most crucial step and need addition database i.e. morph. Correction is done on the basis of the mismatch component of tag. This is explained by following example.

Incorrect Punjabi sentence:

ਦੋ ਮੁੰਡਾ ਸਕੂਲ ਜਾਂਦੇ ਹਨ l(dō muṇḍā sakūl jāndē han)

Two boys go to school

After applying grammatical information (POS tagging):
Now the error detection system will provide the

(ਦੋ_CDPD ਮੁੰਡਾ_NNMSD ਸਕੂਲ_NNMXD ਜਾਂਦੇ_VBMAMPXXXINDA ਹਨ_VBAXBPT1 ।_Sentence)

following error:
From the POS tags it is clear that CDPD is plural and

Error Type: Modifier noun error

Description: The word ਦੋ_CDPD is not in grammatical agreement with word ਮੁੰਡਾ_NNMSD. Because the word ਮੁੰਡਾ_NNMSD is singular and the word ਦੋ_CDPD is plural.

NNMSD is singular. Therefore to correct this error, the word ਮੁੰਡਾ need to be converted in to plural form. Here the role of morph comes into play. From morph the plural word of ਮੁੰਡਾ is ਮੁੰਡੇ and hence to make the sentence grammatically correct, the word ਮੁੰਡਾ should be replaced with word ਮੁੰਡੇ.

Correct Punjabi sentence:

ਦੋ ਮੁੰਡੇ ਸਕੂਲ ਜਾਂਦੇ ਹਨ
(dō muṇḍē sakūl jāndē han)
Two boys go to school

5. Types of Error Covered

When we talked about the grammatical mistakes in written text then there may be countless number of errors in text. Thus it is very difficult to develop a single GEC that could detect and correct all possible errors present in written text. In this research author covered five types of errors. These errors with suitable examples is shown in table 3.

Table 3. Types of error covered with example

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 rihā hai.)	Subject Verb agreement Error.
3.	ਦੋਵੇਂ ਮੁੰਡੇ ਅਮਰੀਕਾ ਜਾ ਕੇ ਗੋਰਾ ਹੋ ਗਏ।(dōvēṃ muṇḍē amrīkā jā kē gōrā hō gaē.)	Noun and Adjective agreement Error.
4.	ਵੱਡਾ ਮੇਰਾ ਮੁੰਡਾ ਸ਼ਹਿਰ ਰਹਿੰਦਾ ਹੈ। (vaḍḍā mērā muṇḍā shahir rahindā hai.)	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. Evaluation Metrics

To measure the performance of the developed system, three basic parameters are used i.e. precision, recall and f-score. These metrics are explained as follow:
Let FCE = number of flagged correct grammar errors
FWE = number of flagged wrong grammar errors,
NFE = number non flagged grammar errors,
Then precision can be defined as percentage of relevant results and can be calculated by using the following formula:

$$\text{Precision} = \frac{CFE}{CFE+FWE}$$

Recall can be defined as percentage of total relevant results correctly classified by an algorithm and can be calculated by using the following formula:

$$\text{Recall} = \frac{CFE}{CFE+NFE}$$

F-measure denotes the accuracy of the system and can be calculate by taking the geometric mean of the precision and recall as shown in the following formula:

$$\text{F-measure}=\frac{2*Precision*Recall}{Precision+Recall}$$

7. Test Result and Discussion

The developed GEC system is manually tested on 600 sentences using mixture of correct and incorrect Punjabi sentences and the output of the test results are recorded manually. To test the system, 410 Punjabi correct grammar sentences and 190 incorrect grammar sentences are taken. Out of 410 correct sentences, 210 sentences are taken from reliable internet sources i.e. e-papers and 200 sentences are taken from standard Punjabi corpus available at ILCI. To perform the testing, mixture of correct and incorrect sentences are distributed into four sets containing 150 sentences in each set. These four sets are given the label as test_set1, test_set2, test_set3 and test_set4. The complete details of the corpus used for testing is shown in table 4. The output of the system is manually evaluated by linguistic.

Table 4. Details of the corpus used for testing the proposed GEC

Type of corpus	Total No. of Input sentences in the corpus
From Punjabi e-papers	210
From ILCI corpus	200
Manually developed test data	190

The developed system is tested on the data mentioned in table 4 and the analysis of the results obtained are shown in table 5 and figure 4. It is clear from the table 5 that the developed system shows an average precision of 0.82, average recall of 0.89 and an average f-measure as 0.85.

Table 5. Test results of proposed GEC

Test data set (Having total 150 sentences in each set)	Actual number of in-correct sentences in the corpus (A)	Statistics based system (Punjabi Grammar Checker)				
		Number of correctly identified in-correct sentences (CFE)	Number of In-correctly identified incorrect sentences (FWE)	Precision $\frac{CFE}{CFE+FWE}$	Recall $\frac{CFE}{CFE+NFE}$	F-measure $\frac{2*Precilon\ X\ Recall}{Precision+Recall}$
Test_set1	59	49	8	0.86	0.96	0.91
Test_set2	46	35	8	0.81	0.92	0.86
Test_set3	53	41	7	0.85	0.87	0.86
Test_set4	32	22	6	0.78	0.84	0.80
Total	190	147	29	Avg:0.82	Avg:0.89	Avg:0.85

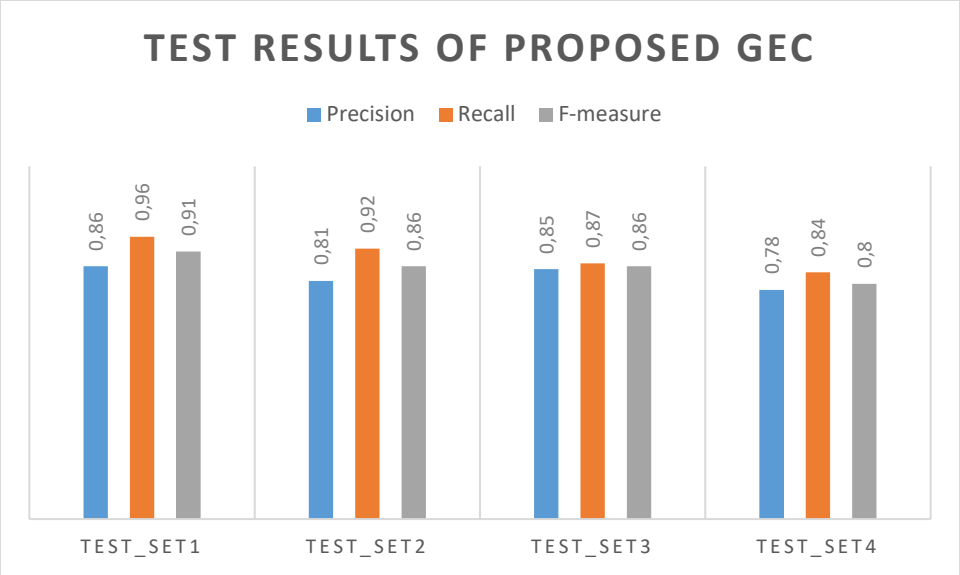

Figure 4. Test results of proposed GEC

8. Comparison with existing Punjabi grammar checker

Rule based grammar checker for Punjabi language (Gill, 2008) Identifies grammatical errors in Punjabi texts such as modifier and noun agreement, subject and verb agreement, noun and adjective, order of modifier of noun in a noun phrase, order of verb in a verb phrase and the like. To detect the errors the system passes through few steps or phases initially, pre-processing task is done on the input text which is tokenization, morphological analysis, rule-based part of speech tagging, chucking and finally, using the grammatical error checking rule. Grammatical errors internal to the phrases and the sentences are identified and correction suggested. The evaluation of the grammar checker shows precision of 76.79%, recall of 87.08%, and F-measure of 81.61%. The researchers stated that the system generated some false alarms for complex and compound sentences.

9. Conclusion and future scope

In this research article, author developed statistics based Punjabi grammar checker in which he used pattern matching along with n-gram probability for detection of errors and class agreement rules for correction of errors. On testing the system on a dataset of 600 sentences, system shows a precision of 0.82, recall as 0.89 and f-measure as 0.85. Further the test data used for testing the system contains 410 correct sentences and 190 incorrect sentences. These incorrect sentences were manually generated by incorporating those errors for which this system has been designed. This grammar checker mainly checks four types of errors i.e. error related to subject verb agreement in terms of number and gender, modifier noun agreement in terms of number and gender, use of KE after the oblique case and order of modifier. In future this system can be extended for long Punjabi sentences like compound and complex sentences and also for some other types of errors alike order in verb phrase, errors related to contractions and long term dependencies etc.

References

[1]. Arppe A. Developing a grammar checker for Swedish. In Proceedings of the 12th Nordic Conference of Computational Linguistics (NODALIDA 1999) 2000 Dec pp. 13-27.

[2]. Brockett C, Dolan WB, & Gamon M. Correcting ESL Errors Using Phrasal SMT Techniques. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, Australia. Published by Association for Computational Linguistics .2006. pp. 249–256.

[3]. Chen HH. Evaluating Two Web-based Grammar Checkers - Microsoft ESL Assistant and NTNU Statistical Grammar Checker. Computational Linguistics and Chinese Language Processing. 2009. 14(2), pp.161–180.

[4]. Dahlmeier D, & Ng HT. A beam-search decoder for grammatical error correction. EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference.2012. pp. 568–578.

[5]. Dhanju, K. S., Lehal, G. S., Saini, T. S., & Kaur, A. Design and implementation of Shahmukhi spell checker. Indian Journal of Science and Technology. 2015. 8(27). pp. 1-12.
[6]. Felice M, Yuan Z, Andersen Ø E, Yannakoudakis H, & Kochmar E. Grammatical error correction using hybrid systems and type filtering. Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics.2014. pp. 15-24
[7]. Flachs S, Lacroix O, Rei M, Yannakoudakis H, & Søgaard A. A Simple and Robust Approach to Detecting Subject-Verb Agreement Errors. Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019).2019. pp. 2418–2427.
[8]. Gill MS., Lehal GS., & Joshi SS. A punjabi grammar checker. In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II.2008.pp.940-944.
[9]. Gill MS., Lehal GS., & Gill SS. A full form lexicon based Morphological Analysis and generation tool for Punjabi. International Journal of Cybernatics and Informatics. 2007. pp. 38-47.
[10]. Goyal V, & Leha, GS. Hindi-Punjabi Machine Transliteration System (For Machine Translation System). George Ronchi Foundation Journal, Italy, 2009. 64(1). pp. 17-23.
[11]. Junczys-Dowmunt, M., & Grundkiewicz, R. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task.2014. pp. 25-33.
[12]. Gupta V, & Singh G. Automatic Punjabi Text Extractive Summarization System. Proceedings of 24th International Conference on Computational Linguistics. 2012. pp. 191–198.
[13]. Han NR., Chodorow M., & Leacock C. Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. In LREC. In proceedings of the LREC 2004 Conference. 2004. pp. n.d
[14]. Azharul Hasan KM, Mondal A, Saha A. Recognizing Bangla Grammar using Predictive Parser. arXiv e-prints. 2012. 3(6), pp. 61–73.
[15]. Heidorn GE, Jensen K, Miller LA, Byrd RJ, Chodorow MS. The EPISTLE text-critiquing system. IBM Systems Journal. 1982. 21(3). pp. 305-26.
[16]. Hermet M, Désilets A. Using first and second language models to correct preposition errors in second language authoring. InProceedings of the fourth workshop on innovative use of NLP for building educational applications. 2009. pp. 64-72.
[17]. Josan GS, Lehal GS. A Punjabi to Hindi machine transliteration system. In International Journal of Computational Linguistics & Chinese Language Processing, Volume 15, Number 2, June 2010 2010 Jun.pp. 157–160.
[18]. Kárason ÖH. Rule based grammar checking on the cheap: Regla. KTH Royal Institute of Technology. 2006.pp. 1–11.
[19]. Kunchukuttan A, Chaudhury S, Bhattacharyya P. Tuning a grammar correction system for increased precision. InProceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task 2014 Jun. pp. 60-64.
[20]. Lehal GS, Singh C. A post-processor for Gurmukhi OCR. Sadhana. Sadhana - Academy Proceedings in Engineering Sciences, 2002 Feb 1. 27(1) pp. 99-111.
[21]. Lehal GS. A Gurmukhi to Shahmukhi transliteration system. Inproceedings of ICON-2009: 7th international conference on Natural Language Processing 2009. pp. 167-173.
[22]. Lehal GS, Saini TS. Sangam: A Perso-Arabic to Indic script machine transliteration model. In Proceedings of the 11th International Conference on Natural Language Processing 2014 Dec. pp. 232-239.
[23]. Lin NY, Soe K, Thein N. Chunk-based grammar checker for detection translated English sentences. International Journal of Computer Applications. 2011. 28(1).pp. 7-12.
[24]. Megyesi B. Brill's rule based part-of-speech tagger for Hungarian. Master's thesis, University of Stockholm. 1998.
[25]. Mizumoto T, Hayashibe Y, Komachi M, Nagata M, Matsumoto Y. The effect of learner corpus size in grammatical error correction of ESL writings. InProceedings of COLING 2012: Posters 2012 Dec. pp. 863-872.
[26]. Mizumoto T, Komachi M, Nagata M, Matsumoto Y. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. InProceedings of 5th International Joint Conference on Natural Language Processing 2011 Nov. pp. 147-155.
[27]. Kubon V, & Platek M. 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.1994. pp. 906-910
[28]. Moré J. A grammar checker based on web searching. Digithum. 2006. (8). pp 1-5.
[29]. Napoles C, Callison-Burch C. Systematically adapting machine translation for grammatical error correction. InProceedings of the 12th Workshop on Innovative use of NLP for Building Educational Applications. 2017. pp. 345-356.
[30]. Madnani N, Tetreault J, Chodorow M. Exploring grammatical error correction with not-so-crummy machine translation. InProceedings of the Seventh Workshop on Building Educational Applications Using NLP. 2012. pp. 44-53.
[31]. Bustamante FR, León FS. GramCheck: A grammar and style checker. arXiv preprint cmp-lg/9607001. 1996.pp.91-96

[32]. Piotrowski M, Mahlow C, Dale R. Proceedings of the Second Workshop on Computational Linguistics and Writing (CL&W 2012): Linguistic and Cognitive Aspects of Document Creation and Document Engineering. In Proceedings of the Second Workshop on Computational Linguistics and Writing (CL&W 2012): Linguistic and Cognitive Aspects of Document Creation and Document Engineering 2012 Apr. pp. 27-34.

[33]. Rozovskaya A, Roth D. Grammatical error correction: Machine translation and classifiers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2016 Aug. pp. 2205-2215.

[34]. Schmidt-Wigger A. Grammar and style checking for German. InProceedings of CLAW 1998. Vol. 98, pp. 76-86.

[35]. Giri KK, Tekchandani RG. *Efficient Rule-Based Grammar Checker with Word Sequencing* (Doctoral dissertation, Thapar University).2017

[36]. Sidorov G, Gupta A, Tozer M, Catala D, Catena A, Fuentes S. Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2). In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task 2013 Aug. pp. 96-101.

[37]. Susanto, R. H., Phandi, P., & Ng, H. T. (2014, October). System combination for grammatical error correction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 951-962.

[38]. Tesfaye, D. (2011). A rule-based Afan Oromo Grammar Checker. IJACSA Editorial. 2(8), pp.126–130.

[39]. Tetreault J, Chodorow M. Native judgments of non-native usage: Experiments in preposition error detection. InColing 2008: Proceedings of the workshop on human judgements in computational linguistics 2008 Aug. pp. 24-32.

[40]. Yuan Z, Felice M. Constrained grammatical error correction using statistical machine translation. InProceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task 2013 Aug. pp. 52-61.

[41]. Kubon V, Platek M. A grammar based approach to a grammar checking of free word order languages. InCOLING 1994 Volume 2: The 15th International Conference on Computational Linguistics 1994.pp. 906-910.

[42]. Gao W, Peng M, Wang H, Zhang Y, Xie Q, Tian G. Incorporating word embeddings into topic modeling of short text. Knowledge and Information Systems. 2019 Nov. 61(2): pp.1123-45.

[43]. Subramani S, Michalska S, Wang H, Du J, Zhang Y, Shakeel H. Deep learning for multi-class identification from domestic violence online posts. IEEE Access. 2019. pp. 46210-24.

[44]. Wang H, Whittington D. Measuring individuals' valuation distributions using a stochastic payment card approach. Ecological Economics. 2005 Nov 1.55(2). pp.43-54.

[45]. Sharma M, Singh G, Singh R. A review of different cost-based distributed query optimizers. rogress in Artificial Intelligence. 2019 Apr. 8(1). pp. 45-62.

[46]. Sharma M, Singh G, Singh R, Singh G. Analysis of DSS queries using entropy based restricted genetic algorithm. Applied Mathematics & Information Sciences. 2015 Sep 1. 9(5). pp. 2599-2608

[47]. Sharma M, Sharma S. The rising number of COVID-19 cases reflecting growing search trend and concern of people: a Google Trend analysis of eight major countries. Journal of Medical Systems. 2020 Jul;44(7):1-3.

[48]. Benczur AA, Csalogany K, Sarlos T, Uher M. Spamrank–fully automatic link spam detection work in progress. InProceedings of the first international workshop on adversarial information retrieval on the web 2005 May. pp. 1-14.

[49]. Jassim AK, Al_Bayaty BF. A Stochastic Approach to Identify POS in Iraqi National Song using N-Iterative HMM using Agile Approach. In IOP Conference Series: Materials Science and Engineering 2021 Feb 1 (Vol. 1094, No. 1, pp. 12-19).

[50]. Cheragui MA, Hiri E. Arabic Text Segmentation using Contextual Exploration and Morphological Analysis. In2020 2nd International conference on mathematics and information technology (ICMIT) 2020 Feb 18. pp. 220-225.