A Systematic Review of Automated Grammar Checking in English Language

Madhvi Soni, Jitendra Singh Thakur

Department of Computer Science and Engineering, Jabalpur Engineering College, Jabalpur, Madhya Pradesh 482011, India.
madhvi.soni21@gmail.com , jsthakur@jecjabalpur.ac.in

Article Info
Page Number: *860 – 895*
Publication Issue:
Vol 70 No. 1 (2021)

Abstract: Grammar checking is the task of detection and correction of grammatical errors in the text. English is the dominating language in the field of science and technology. Therefore, the non-native English speakers must be able to use correct English grammar while reading, writing or speaking. This generates the need of automatic grammar checking tools. So far many approaches have been proposed and implemented. However, fewer efforts have been made in surveying the literature in the past decade. The objective of this systematic review is to examine the existing literature, highlight the current issues and suggest the potential directions of future research. This systematic review is a result of analysis of 20 primary studies obtained after following a search strategy for selecting papers from different resources. Among the main observations, we found that the existing approaches have concentrated mainly on syntax errors. The existing approaches have used an imbalanced dataset for experiments that consists of almost 78% sentences with syntax and semantic errors and almost 16% with sentence level errors. No tool is suitable for real time applications such as proofreading and writing assistance. In this paper, we present a possible scheme for the classification of grammar errors. Also, we present several useful illustrations- most prominent are the schematic diagrams that we have designed for each primary study and some other representations which collate these approaches along different dimensions. This facilitates better understandability, comparison and evaluation of the existing work in this domain.

Article History
Article Received: *20 September 2020*
Revised: *15 October 2020*
Accepted: *02 December 2020*
Publication: *31 January 2021*

Abstract

Grammar checking is the task of detection and correction of grammatical errors in the text. English is the dominating language in the field of science and technology. Therefore, the non-native English speakers must be able to use correct English grammar while reading, writing or speaking. This generates the need of automatic grammar checking tools. So far many approaches have been proposed and implemented. However, fewer efforts have been made in surveying the literature in the past decade. The objective of this systematic review is to examine the existing literature, highlight the current issues and suggest the potential directions of future research. This systematic review is a result of analysis of 20

primary studies obtained after following a search strategy for selecting papers from different resources. Among the main observations, we found that the existing approaches have concentrated mainly on syntax errors. The existing approaches have used an imbalanced dataset for experiments that consists of almost 78% sentences with syntax and semantic errors and almost 16% with sentence level errors. No tool is suitable for real time applications such as proofreading and writing assistance. In this paper, we present a possible scheme for the classification of grammar errors. Also, we present several useful illustrations- most prominent are the schematic diagrams that we have designed for each primary study and some other representations which collate these approaches along different dimensions. This facilitates better understandability, comparison and evaluation of the existing work in this domain.

1. Introduction

English is a West Germanic language which is the second most common language of the world. Over 600 million speakers use English as a second language (ESL) or English as a foreign language (EFL). While writing text in their second or foreign language, people might make errors. Therefore, it is essential to be able to detect these grammar errors and correct them as well. Grammar checking by a human becomes inconvenient at times such as when human resource is limited, the size of the document is large or the grammar checking is to be done on a regular basis. Therefore, it would be beneficial to automate the process of grammar checking. A grammar checking tool can provide automatic detection and correction of any faulty, unconventional or controversial usage of the underlying grammar.

The trend of developing such tools has been evolved from 80's till now. Earliest grammar checking tools (e.g., Writer's Workbench [1]) were aimed at detecting punctuation errors and style errors. In 90's, many tools were made available in the form of commercialized software packages (e.g., RightWriter [2]). In recent decades, rapid development has been seen in this field. For example, Park et al [3] developed a grammar checker as a web application for university ESL students, Tschumi et al [4] developed a tool aimed at French native speakers writing in English, Naber developed an tool named LanguageTool [5] to detect a variety of English Grammar errors, Brockett et al [6] presented error correction using machine translation and Felice et al [7] presented a hybrid system. Existing approaches are hard to compare since most of their tools are not available. Moreover, they are developed on different datasets and targets detection of different types of errors. Study and comparative analysis of previous literature is important to gain future research directions, yet very few efforts have been put to survey grammar checking approaches in the last decade. Therefore, we are highly motivated to review the existing literature for identifying the related issues and concerns, and present them in a single study to our research community.

This paper reports on a systematic review [8] that focuses on various approaches for automatic detection and correction of grammar errors in English text. While reviewing the literature, we have tried to summarize as many details as possible, explaining the complete step by step workflow of the approach along with its strengths and limitations. Our intention is to provide a platform for comparing the existing approaches that will help in taking further research decisions. Also, we have searched the literature to find various types of errors, but found that all the researchers are addressing a set of errors that is different from each other. Thus, we identify major types of errors and suggest an error classification scheme based on a five point criteria. We explain these types of errors along with their demonstrative examples. To the best of our knowledge, our study is the first one of its kind.

The paper is organized into following sections: Section II presents the method of performing systematic review. This section describes our research questions, search strategy, paper selection criteria and method of data extraction from the selected papers. Section III presents our suggested scheme to classify various English grammar errors. Section IV presents the classification of grammar checking techniques. Section V presents a detailed review of our primary studies. Section VI discusses the results and research findings. Finally, section VII concludes our paper and suggests some directions for further research.

2. Systematic Review Method

We report a systematic review on automated grammar checking in English language. As per the recommended guidelines [8], we have adopted five necessary steps to carry this review. In the first step, we formulate the research questions that will be addressed by this systematic review. In the second step, we design a strategy to search for the research papers from different resources. Third step defines the paper selection criteria to identify relevant works. The fourth step is extraction of data from primary studies and finally, in the last step we examine the data.

2.1 Research Questions:

RQ1: What are the different types of errors in English grammar? How can we classify them? Is there any classification scheme in the literature?
RQ2: Which techniques can be used for grammar checking? What are their Limitations?
RQ3: What are the existing approaches or studies of grammar checking? What steps are taken by these approaches to find errors in the English text?
RQ4: What types of errors are detected and corrected by these approaches?
RQ5: Have the performance of the approach been evaluated? If yes, what results have been obtained?
RQ6: How far these approaches are able to correctly identify each type of error?
RQ7: Is there any tool support available to carry further experiments?

2.2 Search Strategy:

We performed an exhaustive search on different online resources to identify the papers to be reviewed. The search was performed using free text keywords such as “grammar checking”, “grammar correction”, “grammatical error correction”, “English grammar errors”, “types of errors”, “error classification”, “ESL errors”, “automatic detection” and “automatic correction”. More articles were also identified by examining the reference list of the articles. Since the search resulted in collection of a large number of papers, it is necessary to identify only the useful papers that can answer our specific research questions. Thus, we applied inclusion/exclusion criteria to select papers that can serve as primary studies in this systematic review.

2.3 Inclusion/exclusion criteria:

Our inclusion/exclusion criteria are completely based on our previously defined research questions. For each paper, we read the paper's title and abstract to identify the relevant papers. Furthermore, full text was read to take the final decision. Following points were considered while deciding on the selection of primary studies:

- Papers irrelevant to the task of grammar checking are excluded.
- Papers proposing grammar checking on languages other than English are completely ignored.
- Papers describing types of errors made by native speakers of a specific language (e.g., errors made by only Arab writers) were excluded.
- Papers that do not provide sufficient technical information of their approach were excluded. (e.g., [9])
- In case of a shared task (CoNLL-2013 and 2014), we include only the best performing approach.

After the electronic search, a total of 141 papers were identified to investigate. 39 duplicates were eliminated and 43 papers were eliminated in the first round by reading the abstract and introduction. So, 82 papers were remaining for further investigation. After reading full-text, 60 papers were eliminated and finally 1 more was eliminated [9] due to lack of implementation details. Thus, we identified 20 primary studies.

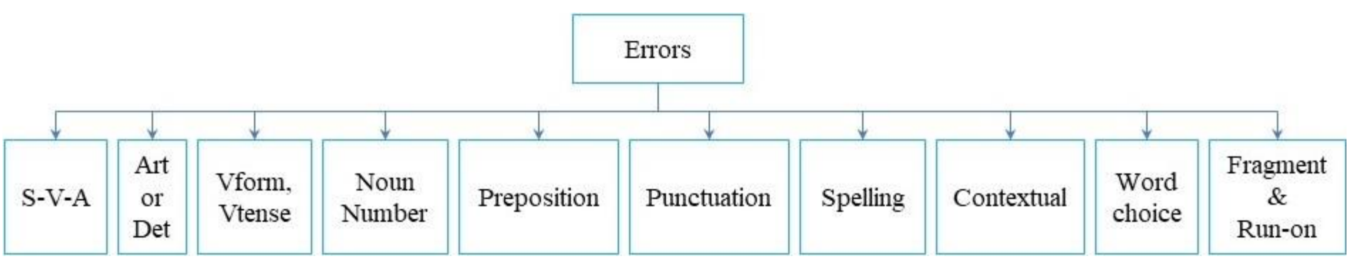
2.4 Data Extraction:

For data extraction, we used a tabular format where each primary study is reviewed under table headings such as name of the approach, technique used, steps involved in the approach, types of the errors addressed by the approach, experiments conducted by the authors (if any), dataset used in the experiment, outcomes of the experiment, name of the software tool designed (if any), and strengths and shortcomings of the approach (if any). Later, content of this table is used to write a detailed review of each primary study.

3. Classification of Errors

This section will address our research question RQ1. Before actual implementation of any grammar checking approach, it is important to identify major types of errors and their classification on the basis of some criteria. For example, some researchers have classified the errors in the corpus based on whether they are automatically detectable or needs human assistance. Naber [5] classifies various errors into four types namely spelling errors, style errors, grammar (syntax) errors and semantic errors. Wagner et al [10] reports four types of errors namely agreement errors, real word spelling errors (contextual errors), missing word errors and extra word errors. Lee et al [11] reports two types of errors namely syntax errors and semantic errors. Z Yuan in her doctoral thesis [12] states five types of errors namely lexical errors, syntactic errors, semantic errors, discourse errors and pragmatic errors. Other than this, there is no general classification of grammar errors to the best of our knowledge. However an overview of major types of errors can be found in many web articles. Thus, we are highly motivated to suggest an error classification scheme. Please see figure 2 and 3 for comparison of our scheme with previous schemes. We have considered following points while designing our suggested classification scheme.

- *Frequency of error:* More frequent errors should be kept in separate groups. For instance, five types of syntax errors as the most frequent errors that occur in ESL text [13] so they are classified into separate groups. Wrong verb is the most common type of error [45]. Similarly, spelling and punctuation errors are also very common. See figure 1 (a).
- *Validity of text:* Errors should be separated on the basis of how it makes the text invalid. For instance, syntax error invalidates a text due to violation of grammar rules. Similarly, sentence structure error invalidates a sentence due to violation of sentence structuring rules [14] and a spelling error invalidates a word if it violates language orthography. See figure 1 (b).
- *Level of an error:* Some errors are detected at sentence level while others can be detected at word level i.e., taking two or three words. For instance, checking words before and after a preposition would be sufficient to detect a preposition error. While fragments are detected using parse tree pattern of a complete sentence. See figure 1 (c).
- *Nature of error:* The errors that are more annoying and difficult to detect should be separated from simpler ones. For instance, spelling error is rather formal which can easily be detected using a spell checker, while detection of a semantic error requires real-world knowledge.
- *Error type overlap:* The error types in the classification scheme are overlapping. It cannot be completely avoided but we have tried to minimize it. For example, a run-on sentence can also be a punctuation error and a missing preposition error can also be a sentence structure error.



(a)

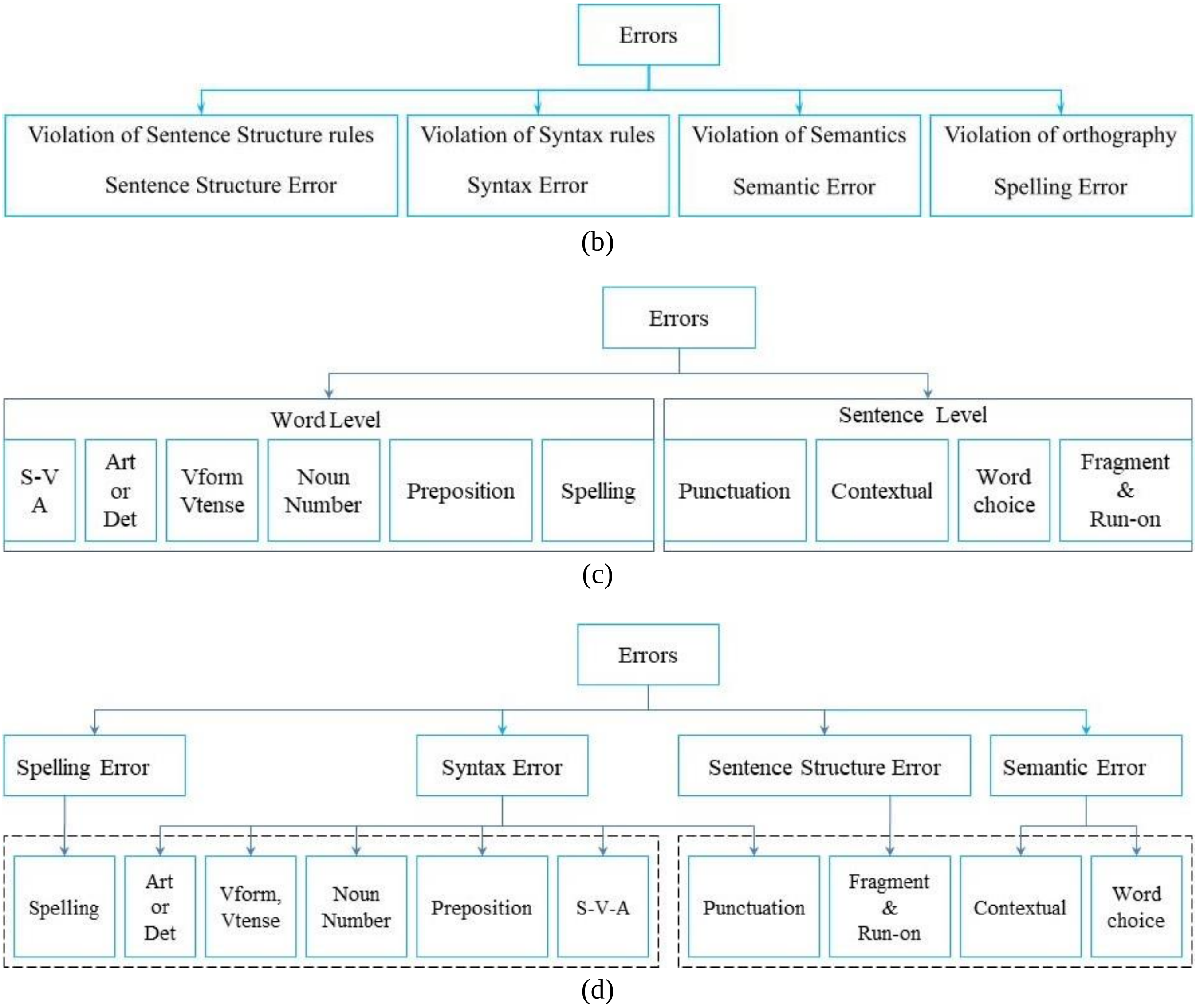


Figure 1: Classification of errors based on (a) frequency, (b) validity, (c) level and (d) combining (a), (b) and (c).

Again considering the frequency, nature and validity, we kept punctuation rules into a separate class of errors. Trying to minimize the overlapping, we reached to the final classification shown in figure 2.

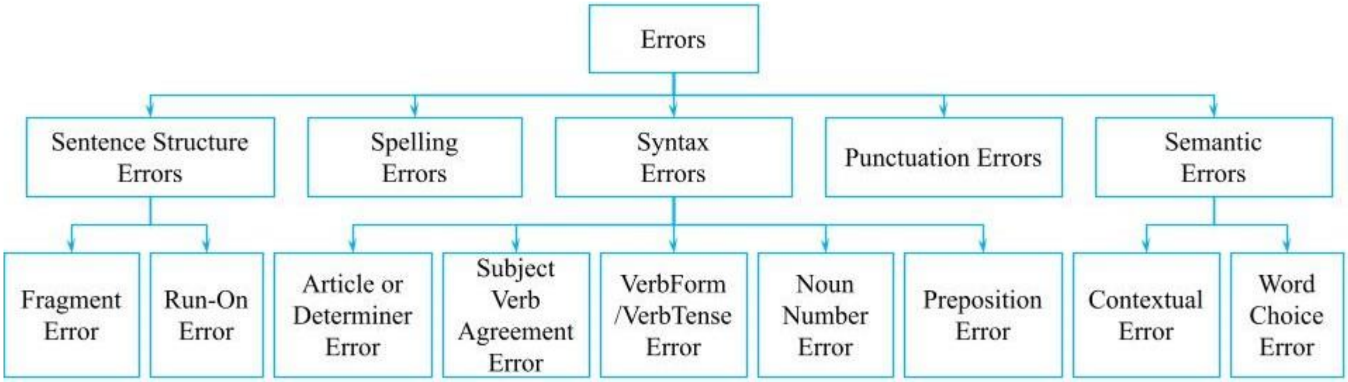
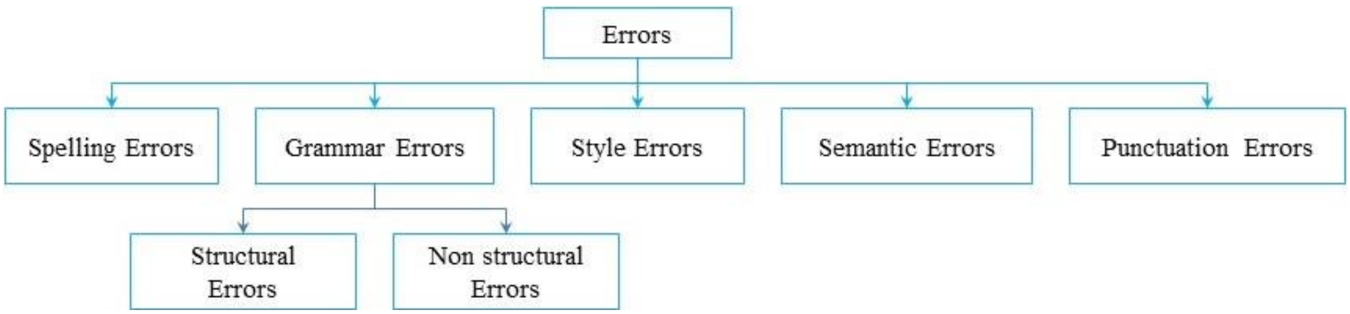
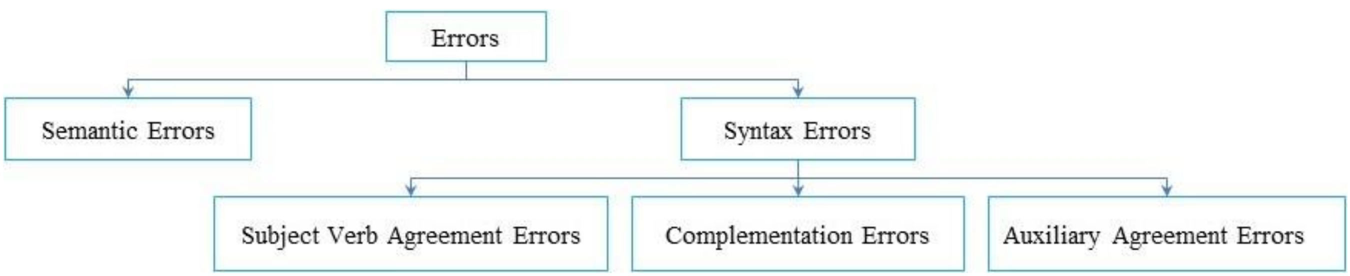


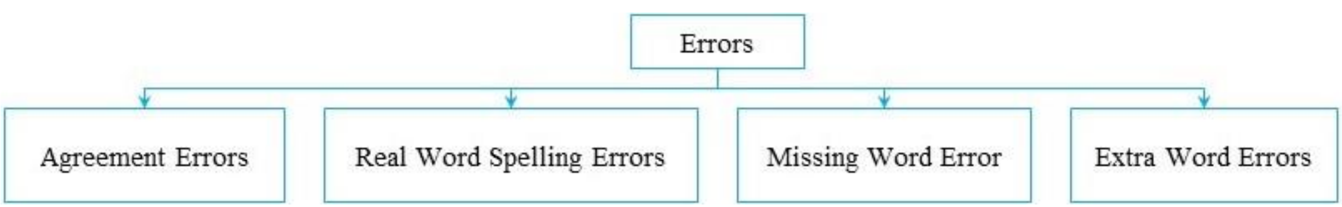
Figure 2: Our Suggested Scheme for Classification of Errors



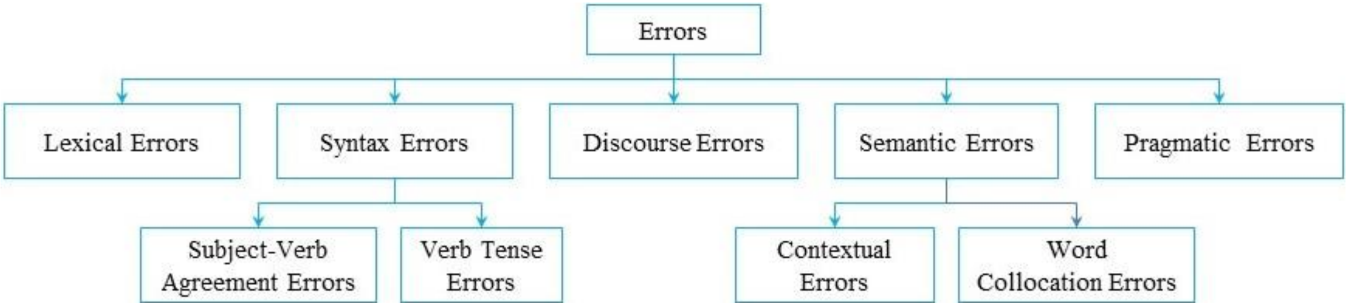
(a)



(b)



(c)



(d)

Figure 3: Classification schemes given by (a) Naber [5], (b) Lee et al [11], (c) Wagner et al [10], (d) Z yuan [12]

Here, we are describing our suggested classification of errors. We give erroneous sentences for each type of error and their corrections are given in the bracket. All the examples have been taken from [15].

(1) Sentence Structure Error:

Sentence structure refers to the organization of different POS components within a sentence to give it a meaning. Structuring has a high impact on sentence's readability. Hornby [14] has formulated 25 patterns of English sentences. Thakur et al also framed a set of sentence structuring rules for English sentences [16], [17]. If none of those patterns or rules are found, the sentence can be considered as ill-formed or say erroneous. Such an ill-formed sentence can further be classified as **fragments** and **run-ons.** A fragment is an incomplete sentence in which either subject or verb is missing or it may be a sentence having dependent clause without the main clause [18]. A run-on sentence is two independent clauses missing a punctuation or necessary conjunction between them, which affects the readability of text. Sentence structure errors may contain other type of errors within them. Examples 1, 2 are correctly constructed while examples 3,4,5,6,7 are erroneous. Examples 4,5,6 are fragments while example 7 is a run-on.

Example 1- She began singing. (S-V-Gerund)
Example 2- She wants to go. (S-V-to-infinitive)
Example 3- She began to singing. (Misplaced ‘to’ or ‘-ing’)
Example 4- Wants to go. (Subject is missing)
Example 5- A fair little girl under a tree. (Verb is missing)
Example 6- Because he is ill. (‘because’ makes it a dependent clause, main clause is missing)
Example 7- I ran fast missed the train. (Conjunction ‘but’ is missing)

(2) Punctuation Error:

Punctuation marks like comma, semi-colon, full stop etc. are used to separate sentence elements. A missing punctuation or unnecessary punctuation can alter the meaning of the sentence. Hence, it is important to detect and correct the punctuation errors in English text.

Example 8- He lost lands money reputation and friends. (lands, money, reputation and friends)
Example 9- Alas she is dead! (Alas! She is dead.)
Example 10- How are you? Mohan? (How are you, Mohan?)
Example 11- Exactly so, said Alice. ("Exactly so," said Alice.)

(3) Spelling Error:

Spelling error is the generation of a meaningless string of characters. A common reason for such errors is the typing mistakes done by the writers. These are the most common error types that can be found easily by any spell or grammar checking tool. Generally these tools have a list of known words. Any word outside this list is considered as a spelling error.

Example 12- Death lays his icey hand on kings. (icy)
Example 13- Many are called, but few are choosen. (chosen)

(4) Syntax Error:

Any error violating the English grammar rules is called as syntax error. Syntax errors can be of many types depending upon the inherent relationship between the words of a sentence. Most grammar checkers aims at finding and detecting various types of syntax errors. Syntax errors can be subdivided into five subtypes:

(4.1) Subject-Verb Agreement Error: A sentence written in English must have an agreement between subject and verb in terms of person and number. This agreement is shown in examples 14 and 15.

Example 14- He is not to blame. (subject- 'he' (third person singular) (verb-'is' (third person singular))
Example 15- They are not on good terms. (subject- 'they' (third person plural)(verb-'are' (third person plural))

(4.2) Article or Determiner Error: This type of error occurs either when an article or determiner is missing in the sentence or when a wrong article or determiner is used.

Example 16- Book you want is out of print. (The book)
Example 17- He returned after a hour. (an hour)

(4.3) Noun Number Error: In English, uncountable or mass nouns do not have plurals. So noun number error occurs when a plural form of uncountable nouns is used in the text.

Example 18- He paid a sum of money for the informations. (information)
Example 19- The sceneries here are very good. (The scenery here is very good.)

(4.4) Verb Tense or Verb Form Error: Verb tense or verb form conveys the time and state of the idea or event. This type of error occurs when a writer uses a different tense or form of verb from the intended one.

Example 20- It is raining since yesterday. (has been raining) ('since' gives the idea that the event has started in the past and is still continuing)
Example 21- She leaves school last year. (left) ('last year' indicates a finished event of the past)
Example 22- The boys are play hockey. (playing) (the event is currently happening, so -ing form of verb is required)

(4.5) Preposition Error: Prepositions are the words preceding a noun or pronoun, used to express a relation to other element in the clause. In literature, preposition errors are addressed separately because of the fact that it is difficult to master them.

Example 23- He sat a stool. (He sat on a stool.)
Example 24- He has recovered of his illness. (from his illness)

(5) Semantic Error:

The errors that do not violate English grammar rules, but make the sentence senseless or absurd, are called as semantic errors. A semantic error can be a **contextual error** [19] or **wrong word choice** error. When a wrongly typed word is a real word in the language, it is not detected as a spelling error, yet it does not fit in the given context; such errors are called as contextual errors. Wrong word choice error is using a rare word (possibly due to limited knowledge of vocabulary) which is often not used in the given context. Examples 25,26 are contextual errors while 27,28 are word choice errors.

Example 25- Our team is better then theirs. ('then' is not a spelling mistake, but the context gives an idea of comparison, indicating correct word as 'than')

Example 26- The jury were divided in there opinions. (their opinions)
Example 27- A group of cattle is passing. (A herd of cattle)
Example 28- I am going to the library to buy a book. (use 'bookstore' instead of 'library')

4 Classification of Techniques

This section will address our research questions RQ2. See figure 4. There are three main techniques of grammar checking:

4.1 Rule based technique:

The classical approach of grammar checking is to manually design grammar rules as shown in [5]. These High quality rules are designed by linguistic experts. An English text tagged with parts-of-speech (*henceforth POS*) is checked against the defined set of rules and a matching rule is applied to correct any error. The technique appears to be simple as it is easy to add, edit or remove a rule; however, writing rules needs extensive knowledge of the underlying language's grammar. Rule based systems can provide detailed explanation of flagged errors thus making the system extremely helpful for the purpose of computer aided language learning. The limitation with this technique is that rules usually have exceptions and it is difficult to utilize corpus statistics to handmade rules. Also, manual maintenance of hundreds of grammar rules is quite tedious.

4.2 Machine Learning based technique:

Machine learning is currently the most popular technique of grammar checking. A method that uses supervised learning provides best results [20]. These methods use an annotated corpus which in turn is used to perform statistical analysis on the text to automatically detect and correct grammar errors. Unlike rule based systems, it is difficult to explain the errors resulted by these systems. Machine learning based systems does not require extensive knowledge of the grammar since it is completely dependent on the underlying corpus. Non-availability of a large annotated corpus hinders the application of such techniques for grammar checking purpose. Also, the results greatly depend on how clean the corpus is.

4.3 Hybrid technique:

A combination of machine learning and rule based techniques can be utilized to improve the performance of the system. Since, some errors are better solved by rule based technique (e.g., use of `a' or `an') and some are better solved by machine learning (e.g., determiner errors). So, each part of the hybrid technique should be implemented according to its ‘competence’ [20]. As experimented in [21], the corpus of text can be used to train the system for identifying correct pattern of sentences and the results can be filtered by applying some hand-crafted rules. Hybrid technique is helpful in addressing a wide range of complex errors. Also, the tedious job of writing so many rules can be reduced to a greater extent.

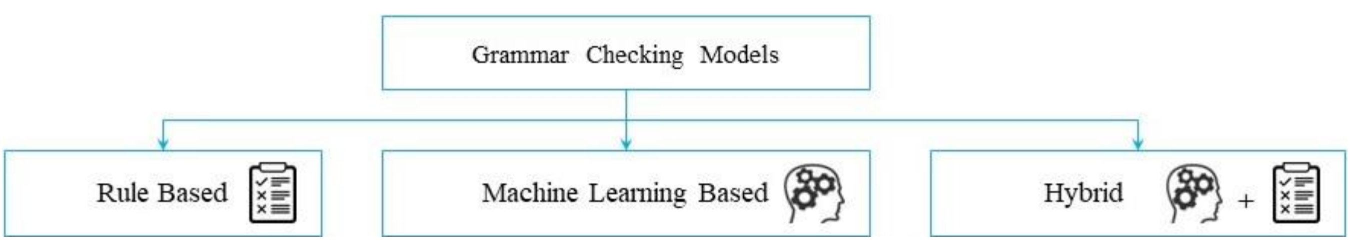


Figure 4: Classification of Grammar Checking Techniques

ISSN:2094-0343
2326-9865

5 Digest of the Primary Studies

In this section, we present our study of various approaches that we have selected as our primary studies. For each primary study, we present the working of the approach, its graphical representation, and discuss the types of errors that can be detected or corrected by it, their strengths, limitations and the reported performance. This section will address RQ3 and RQ5 (also see figure 25). RQ4 and RQ7 will be addressed by table 2 and table 1 respectively.

Table 1: Summary of Various Grammar Checking Approaches

	Technique used	**Target Error types**	**Dataset**	**Tool support**	**Strengths**	**Limitations**
[3]		Wrong capitalisation, SVA, vform, missing fragments	Essays written by university students.	No	Simple, Customizable to identify frequent errors	Handmade rules, No automatic correction
[4]		Spelling, SVA, vtense, wrong word choice, sentence error, noun error	Self created Corpus of 27 000 words of text by French native speakers	No	reduced processing time for tagging	Error Overflagging is high, 43.5%
[5]		Punctuation, Syntax, semantic, style error	Mailing list error Corpus of 224 sentences	Yes	Simple, Large set of rules, Easy rule addition	No automatic correction, difficult to manage large number of handmade rules
[22]		Determiner, noun number, vtense, wrong word choice, Vform	SST Corpus of 221 sentences	No	Better error correction due to best fit method used	error over flagging, poor performance for SVA, Vform errors
[6]		Mass noun error	Reuters news wire articles, CLEC Corpus, English sentences from Chinese websites	No	Automatic correction	Unable to detect error when mass noun is also a count noun

ISSN:2094-0343
2326-9865

[21]		Preposition errors	MetaMetric Corpus of 1100 & 1200 lexile text, newspaper text Chinese, Japanese and Russian's ESL essays	No	Provides better results due to hybrid approach	Insufficient number of rules, low recall, many errors were deliberately skipped
[23]		SVA, article noun number, verb error wrong adjective, adverb or pronoun	Reuters- 21578 Corpus, sentences from book "Avoid Errors" by AK Mishra	No	Language independent method, quicker response by frequent detectors	False error detection
[24]		Syntax, wrong word choice, sentence structure error	Hiroshima English learners Corpus, Japanese learners of English Corpus, Chinese learner error Corpus	No	Good feature set, provides better error detection	Does not detect spelling error
[25]		Spelling, phrases , SVA, punctuation, article ,preposition, gerund misuse and other POS errors	Self created Corpus of incorrect and correct sentences collected from lang-8.com	No	Automatic rule generation	Most of the detected errors are spelling or phrasal errors
[26]		Article, preposition	NUCLE Corpus, Giga word Corpus, section 23 of Wall Street journal	No	Automatic correction, good performance	False flags, UN identified errors low recall, low precision
[13]		Noun number, SVA, vform, ArtOrDet, preposition	NUCLE, Google web 1T 5-gram Corpus	No	Automatic correction good performance	Low recall for preposition, inconsistency in error detection

[7]		Total 28 types of errors [7]	NUCLE, CLC, FCE, EVP, CoNLL-2014 dataset	No	Automatic correction, good performance on punctuation, spelling, capitalization, noun number, vform and SVA errors	Cannot handle fragments, run ons, acronyms, idioms, word reordering and co-location errors
[27]		Total 28 types of errors [7]	Lang-8.com, Google web 1T 5-gram Corpus, CoNLL-2014 dataset	No	Custom feature set improves performance	Individual contribution of classified and ML systems are not clear
[28]		Spelling, vtense	Not specified	No	Simple	No automatic correction, does not cover all error types
[29]		Determiner, preposition, noun, verb , punctuation errors	Lang-8, JFLEG	Yes	Artificial rules and custom features set, improved performance	Does not cover all other types
[30]		Total 21 types of errors	W&I, LOCNESS, NUCLE, Lang-8	Yes	Optimize performance on all error types	Does not detect run-ons, semantic errors
[31]		Morphological errors, article, preposition, spelling errors	JFLEG, FCE Corpus	Yes	Useful for low resource languages, requires minimal annotated data	Covers very few type of errors, increase number of false positives
[32]		Spelling, punctuation, Syntax error	News crawl 2017, W&I , LOCNESS	No	Use of simple tools like Google translation, pyspellchecker	high number of false positives, noun , verb missing words not detected

[33]		Spelling, punctuation, Syntax error	NUCLE, Lang-8, NYT2007, CoNLL 2014, JFLEG	No	Less correction per sentence, better recall for all error types	Poor performance for noun, verb errors
[34]		21 error types	lang-8, NUCLE, FCE, W&I, LOCNESS	No	Optimized f score	Performance of different error types is unknown

5.1 English Grammar Checker: Park et al[3] developed a web interface named English grammar checker which aims at the detection of grammatical errors commonly made by university students. The approach utilizes Combinatory Categorial Grammar (CCG) to derive syntax information of a sentence in a categorical lexicon. Each categorical lexicon is a collection of lexical entries. An entry is a kind of rule which defines acceptable categories of words that are local to a given word. For example, for article ‘a’, the entry would describe that after article ‘a’, category NP (third person singular) is expected and further category VP (compatible with NP) can be expected to form a sentence. If a sentence derivation violates such rule, associated error message is displayed. The authors have tested the approach for identifying errors made by students of University of X in their English essays. See figure 5.

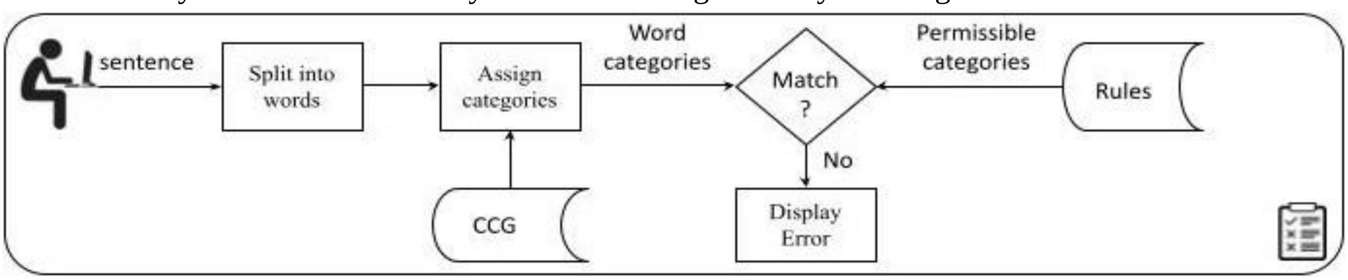


Figure 5: Schematic Diagram of English Grammar Checker [3]

This is a purely syntactic approach, where grammar errors concerning the wrong syntax of a sentence can be detected. A sentence is rejected, if its derivation is not acceptable. To accept a sentence, simply add a new entry in the lexicon. The approach is able to detect spelling errors, article or determiner error, agreement errors, missing or extra elements and verb tense error. *Limitations:* All other type of errors such as wrong word choice errors, preposition errors and run-ons could not be detected. Also, it reported some level of misdiagnoses. This interface in currently not available on the web.

5.2 Island processing based Approach : Tschumi et al [4] developed an English grammar checking tool for native speakers of French language using a method called island processing. The tool works in four steps. In the first step, input text is broken into sentences and words. The words are assigned a syntax category (POS tag). In the second step, a set of finite state automata is used to identify noun phrases, verb phrases and prepositional phrases as the important islands in the sentence. Depending upon the type of noun or preposition, they are assigned specific features and stored in registers. The third step calls the error detection automata which matches the word features to decide on an error and suggests a correction to it. The authors have compared their prototype with other commercial grammar checkers and have reportedly performed better. However, they did not discuss the data which they have used for comparison. See figure 6

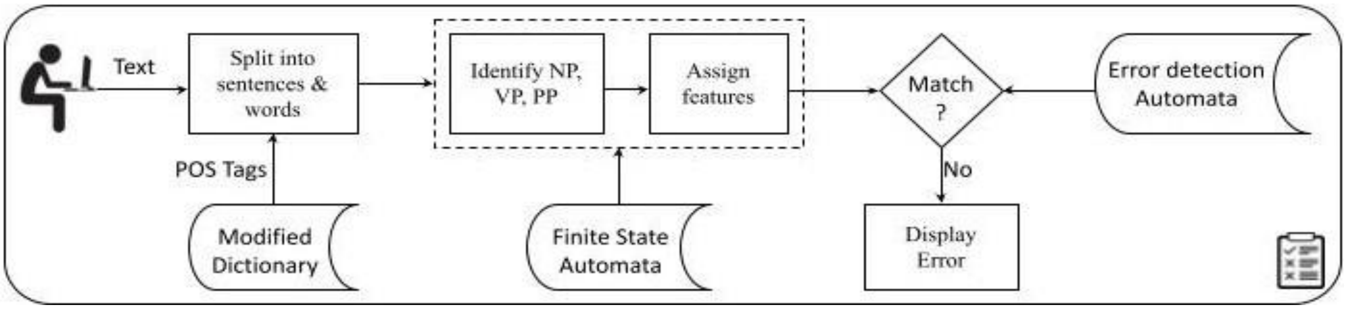


Figure 6: Schematic Diagram of Island Processing based Approach [4]

The proposed prototype uses a scaled-down version of full dictionary which consist of words along with its syntactic category to assign POS tag instead of using a parser which saves time when parsing an ill-formed sentence. Also, island processing method lowers the processing time. To reduce overflagging, an error must be correctly identified. For this purpose the tool provides a user interaction module asking question to user and a problem word highlighter explaining problematic word usage. *Limitations:* Though the method successfully reduces overflagging of errors, it does not support automatic correction. The tool explains an error, suggests possible corrections and often asks question to the user, which seems annoying. The tool is not available online.

5.3 LanguageTool : Naber [5] proposed LanguageTool which is an open source English grammar checker based on traditional rule based approach. The method splits the text into chunks and all the words are POS tagged. The task of spell checking is done by Snakespell python module integrated with the system. It uses a probabilistic tagger Qtag with a rule based extension for POS tagging and a rule based chunker for chunking of text into phrases. Next the manually designed XML based rules are applied to detect errors in the text. These rules define the erroneous pattern of POS tags. When applied, each rule matches the tag pattern given in the rule with the tag pattern present in text. If a match occurs, an error is detected and the system provides explanation messages and example sentences. See figure 7.

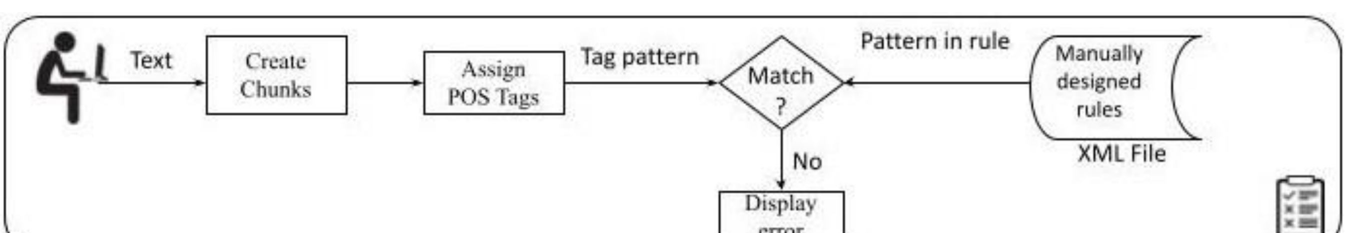


Figure 7: Schematic Diagram of LanguageTool [5]

LanguageTool can be used as a standalone web application as well as can be integrated with a text editor. It supports more than 20 languages with different number of rules. For English, it has 1614 XML rules. The rule set can be extended by simply adding the rule in the XML file. This tool is available at https://languagetool.org. *Limitations:* The obvious drawback of such system is the complex and time consuming task of rule development. Also, the large number of rules to cover majority of errors, results in its low recall. The author did not discuss about the data on which the tool was tested.

5.4 Arboretum: Bender et al [22] proposed Arboretum, a tool to correct English grammar sentences based on some rules called as mal-rules. The authors have classified mal-rules in three categories namely syntactic construction mal-rules, lexical mal-rules and mal-lexical entry. The rules are then used to map correct string from incorrect one using a best-first generation method which they named as `aligned generation'. Aligned generation generates a sentence that closely matches to structure and lexical yield of some reference sentence. Priorities are assigned to the generation tasks and working up with the priorities, the first complete tree found by the generator is considered as the closest to the reference parse. See figure 8

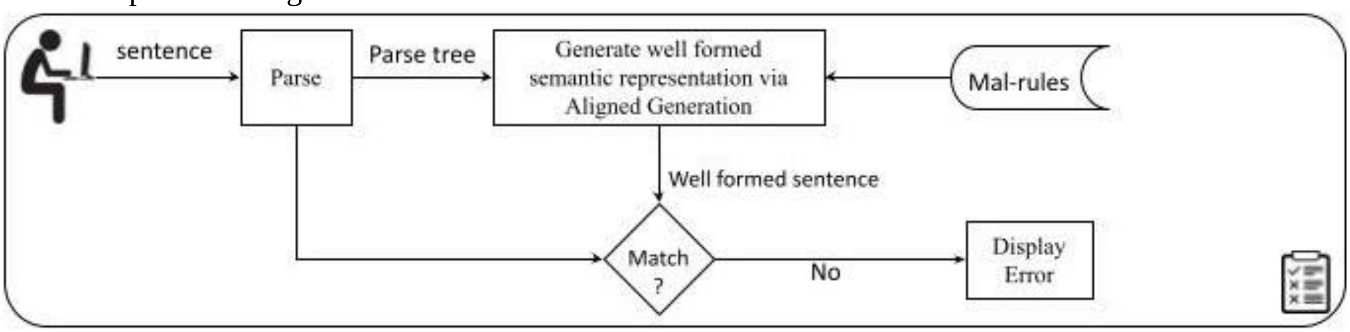


Figure 8: Schematic Diagram of Arboretum [22]

The system was tested on a sample of 221 items taken from SST [35] corpus. The authors report that the tool is able to generate correct string in 80% of cases of the experiment. However, the experimental dataset taken was small with an aim of finding a few types of errors. *Limitations:* The proposed strategy failed in some cases due to its inability in identifying lexical entries and phrasal tasks. The tool is not available online, so it is not clear whether it supports automatic correction or not.

5.5 SMT based approach : The approach proposed by [6] makes use of Statistical Machine Translation (SMT) to detect and correct grammar errors. Aiming at mass noun errors, the authors advocate translation of the whole erroneous phrase instead of individual words. A noisy channel model was used for error correction using SMT technique. The work identifies 14 nouns that frequently occurred in CLEC corpus [36] The sentences containing these errors are used to create training data which can map erroneous string to correct one. See figure 9

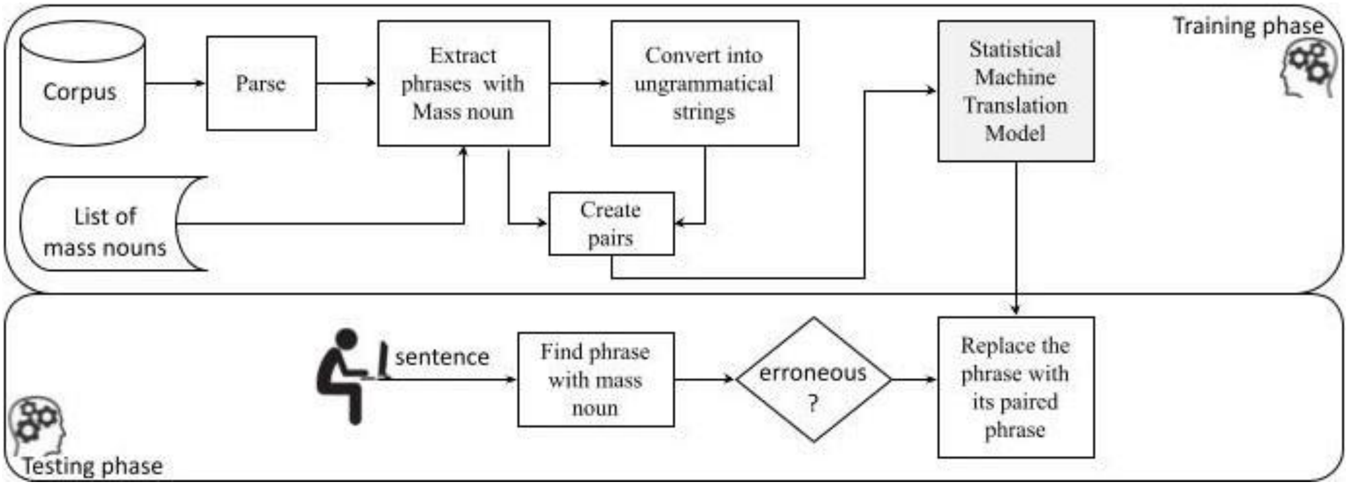


Figure 9: Schematic Diagram of SMT based Approach [6]

The system was tested on 123 example sentences taken from English websites in China. During testing, the approach was able to correct 61.81% of mass noun errors. *Limitations:* Errors like subject verb agreement and punctuation errors were simply ignored. The system was not able to correct an error where a word is both mass noun and count noun; for example, the word 'paper' in the given two phrases- 'many paper' and 'five pieces of papers'. Also the training data did not cover all other types of grammar errors made by ESL learners. This system is not available online.

5.6 Maximum Entropy Classifier based approach : The approach proposed by [21] aims at detecting prepositional errors in a corpus of ESL text. For this task a maximum entropy model is used which is trained with prepositions along with a set of associated feature-value pairs (its context). The sentences are POS-Tagged and chunked. 25 features were used to train that model where each feature is associated with some values. The feature-value pairs having very low frequency of occurrence were eliminated. The model is then tested on a different dataset. The model predicts the probability of each preposition in the given context and then compares it with the preposition used by the writer. The erroneous preposition is replaced with most probable preposition. Subsequently, each context is classified into one of the 34 classes of preposition. To solve the problem of detecting extra preposition, authors devised two rules- Rule1 deals with repetition of same preposition and error is detected when same POS tag is used. Rule2 deals with wrong addition of a preposition between a plural noun and a quantifier. See figure 10

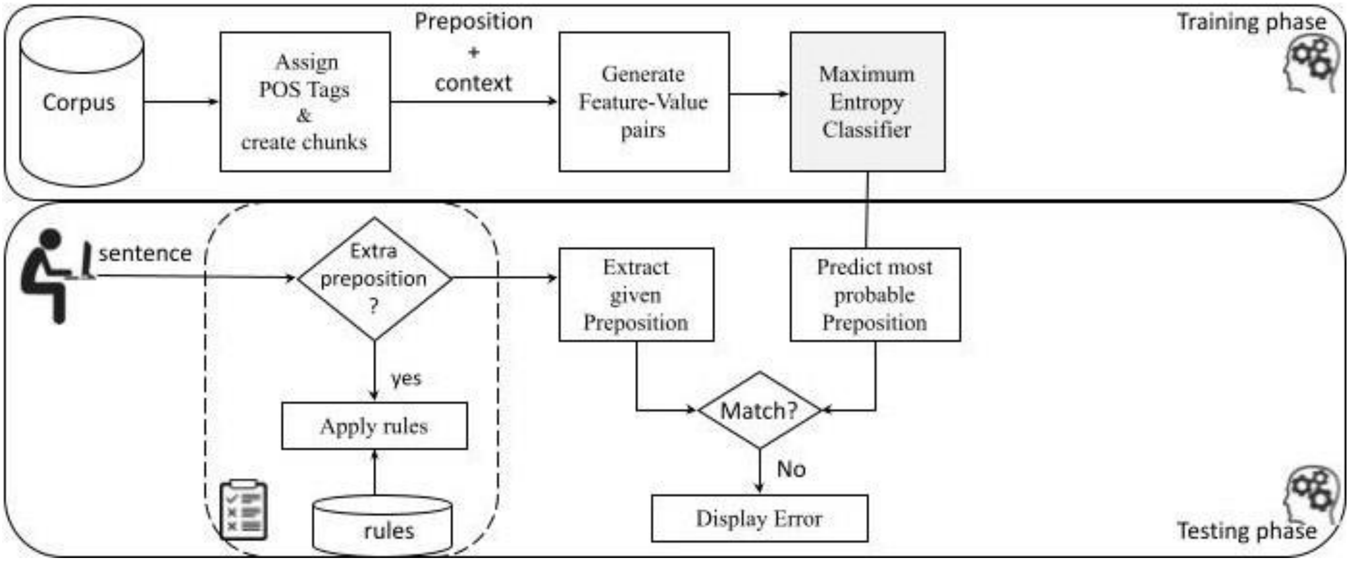


Figure 10: Schematic Diagram of Maximum Entropy Classifier based Approach [21]

This approach uses a huge dataset for training and testing purpose. Training is done on 7 million prepositional contexts taken from MetaMetrics corpus and newspaper text and testing is done on 18157 prepositional contexts taken from a portion of Lexile text and 2000 contexts from ESL essays. *Limitations:* This approach deliberately skips the contexts in the following cases- when there is a slight difference between the most probable and second most probable preposition, when adjacent words are misspelled, when there are comma errors, when the writer uses antonym of a preposition, and also in case when benefactives are used. Also, the rules for detecting extraneous preposition are insufficient to cover other types. No tool support is available for this approach.

5.7 AIS based approach : Kumar et al [21] proposed an approach of grammar checking, inspired from human immune system. Like the human immune system generates immune cells to detect antibodies, similarly a large corpus can be used to detect ungrammatical sentences by generating detectors. The detectors are the sentence constructs that do not appear in the corpus. A test sentence is taken to form bigrams, trigrams and tetragrams. These are tagged with extended POS tags. The sequence of tags which does not exist in the corpus is called as detector and is used to flag error. Next the detector is cloned to repair ungrammatical construct into correct one. The authors have used Real Valued Negative Selection Algorithm to generate detectors and also to fine tune the set of detectors which are capable of identifying errors better and quicker. See figure 11

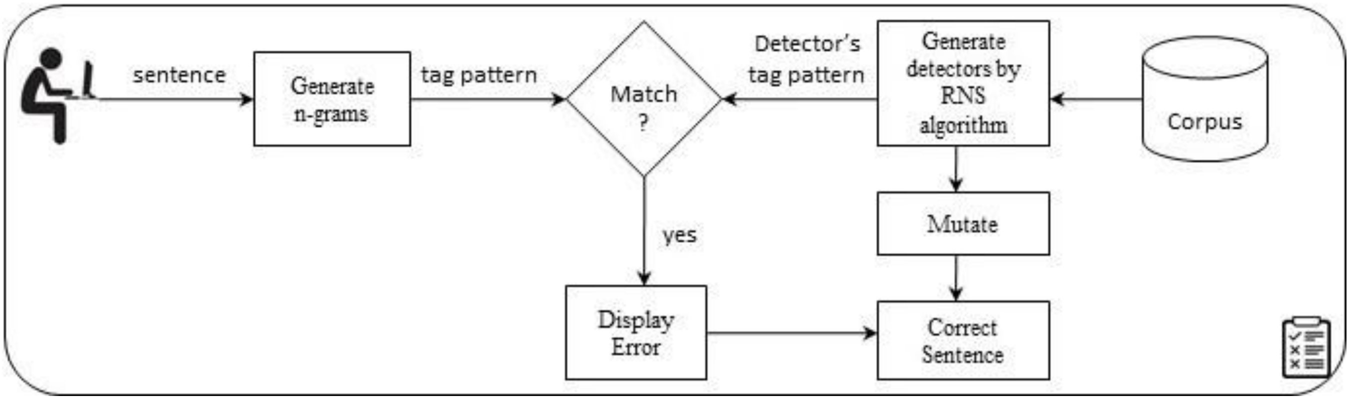


Figure 11: Schematic Diagram of AIS based Approach [23]

This is a language independent approach based on Artificial Immune System (AIS) where the underlying corpus (Reuters-21578) mimics the human immune system. Any sentence construct outside the corpus is regarded as error even though it is grammatically correct. The authors have tested the system on sentences taken from a book named "Avoid errors by A.K.Misra". However, the size of the testing data and the results of the experiments are not discussed in their paper. *Limitations:* It identifies 8 type of errors namely subject-verb agreement errors, wrong verb tense, adverb, adjective error, article, pronoun, wrong noun number error and missing verb errors. All other types of errors such as preposition errors, semantic errors etc. are not detected by the approach. The authors argue that these shortcomings of the approach can be solved by extending the POS tag set. Still, the task of creating a corpus which is large enough to include all type of correct sentences seems practically infeasible. No tool support is available for this approach.

5.8 LSPs based approach : Sun et al [24] proposed an approach which combines pattern discovery and machine learning to classify a sentence into two classes: correct and erroneous. To build this classification model, labeled sequential pattern (LSP) is used as an input feature. The training data is POS tagged and the frequently occurring patterns are discovered from both correct and erroneous sentences. Based on whether the pattern satisfies the given constraints for support and confidence, these patterns are labeled as erroneous or correct. Along with LSPs, other linguistic features like syntactic score, lexical collocation, function word density and perplexity are also used to detect different types of errors. See figure 12

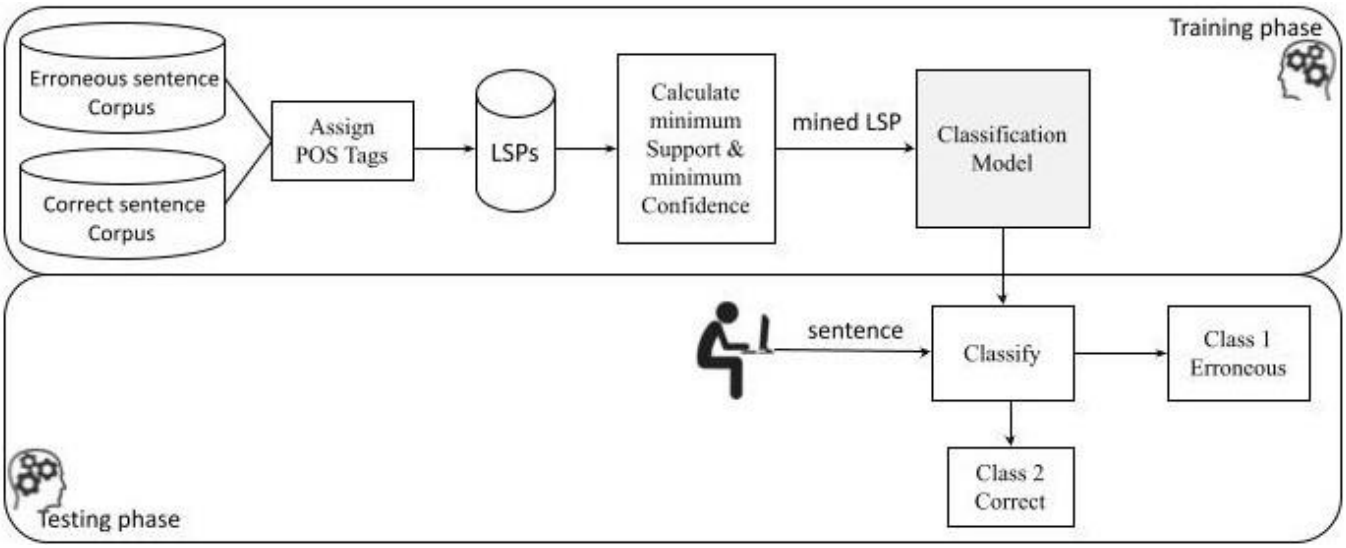


Figure 12: Schematic diagram of LSP based Approach [24]

The method was implemented on SVM and Bayesian classifiers using Hiroshima English Learners' Corpus, Japanese Learners of English Corpus and CLEC corpus [36]. Different experiments were carried out to analyze and compare various results. The Authors found that LSP feature performs better in every case. Also they compared their method with two prototypes and it outperformed the other two in terms of precision, recall and F-score. The method can detect various grammar errors, lexical collocation errors and wrong sentence structure errors. *Limitations:* The automatic correction of detected errors is not supported. Also spelling errors are simply ignored. No tool support is available for this approach.

5.9 Auto-Editing: Huang et al [25] developed an online tool for automatic grammar error correction. The approach uses a manually created corpus of paired sentences collected from a website lang-8.com. A pair consists of an erroneous sentence and its respective correct sentence. The corpus is used to derive sentence correction rules. A rule is represented as A→B where A is a word pattern found in erroneous sentence and B is the pattern found in its respective correct sentence. These patterns are identified by calculating Edit distance at word level. An edit distance (Lavenstein distance) is the minimum number of insert, delete or substitute operations required to transform erroneous pattern to correct pattern. This would result in the generation of candidate rules. Among these, the rule which is able to transform erroneous sentence of the corpus to its correct form is applied while others are ignored. See figure 13

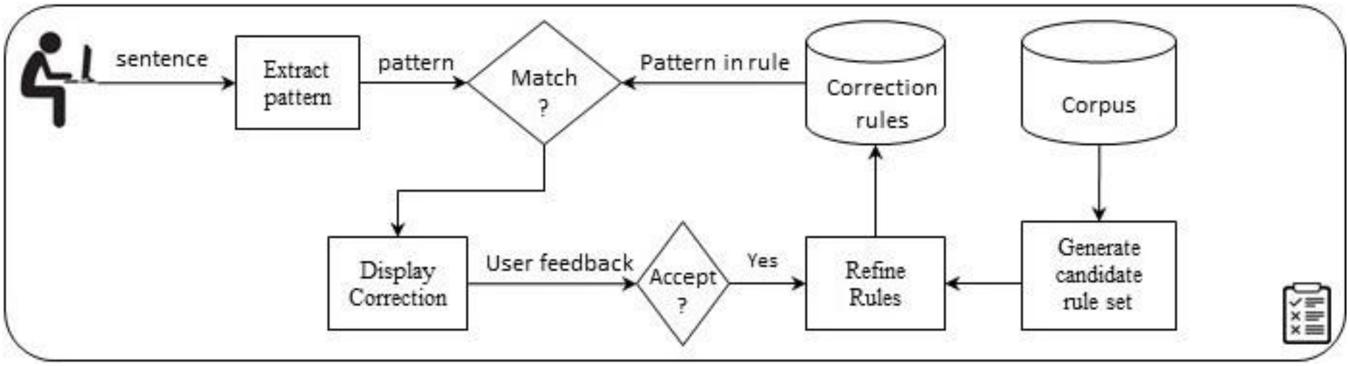


Figure 13: Schematic Diagram of Auto-Editing [25]

This approach is an application of pattern mining on English sentences. In this approach, the rules are automatically derived from the corpus itself. The candidate rule set is refined using a condensing algorithm and ranking the rules based on user feedback (frequently used rules are top ranked). It achieves better precision (40.16) and recall (20.28) when compared with ESL assistant and Microsoft Word 2007 grammar checker. Limitations: The tool detects mostly spelling and phrasal errors and does not cover other types of grammar errors like fragments and run-on sentences. The demo webpage of auto-editing is currently not available.

5.10 ASO based approach: Dahlmeier et al[26] proposed grammar error correction using a linear classifier. They aim at correction of article and prepositional errors. The article or preposition and its context is treated as feature vectors and the corrections are treated as the classes. They used a combination of learner and non-learner text for training. The classifier is trained using Alternating Structure Optimization (ASO) algorithm. ASO algorithm is learning the common structure of multiple related problems. This common structure can be learned by creating auxiliary problems. Auxiliary problems are helpful in predicting the wrong article or determiner in the user text. Then the classifier can be trained for these auxiliary problems to classify articles into 3 classes and prepositions into 36 classes. See figure 14

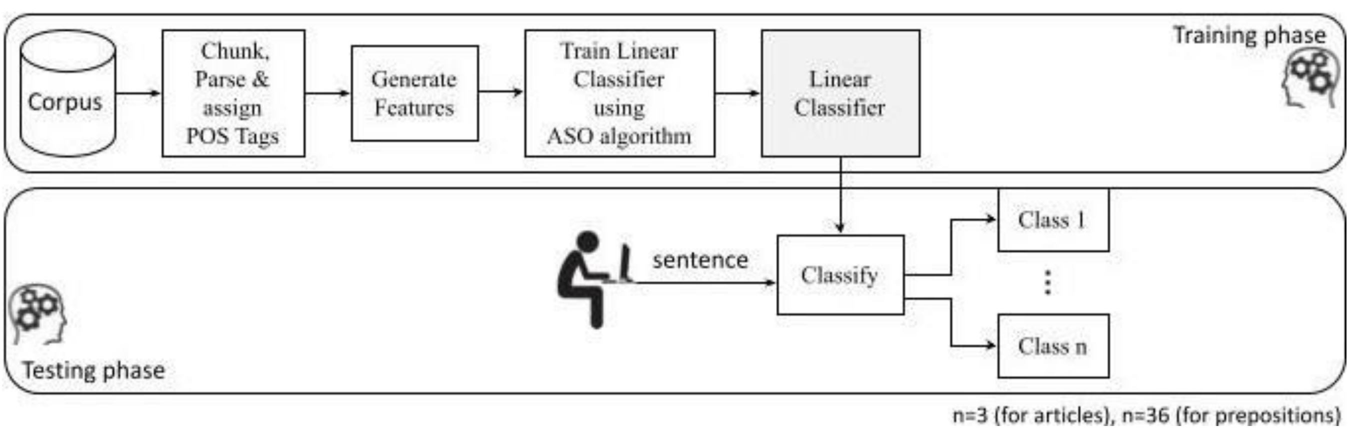


Figure 14: Schematic Diagram of ASO based Approach [26]

Testing of the ASO model was done on text from Wall Street Journal [37]. The results were compared with a classifier trained on Gigaword corpus (https:// catalog.ldc.upenn.edu/LDC2009T13) , a classifier trained on NUCLE corpus [38] and two commercial grammar checking tools whose names were not disclosed. The ASO method outperformed all the systems. *Limitations:* The F-score when compared with other classifiers (19.29% for articles and 11.15% for prepositions) is very low. Also, the problem of unidentified errors and false flags still persists in the system. No tool support is currently available for this approach.

5.11 UI System : This system was developed by [13] at CoNLL-2013 shared task which aims at correction of five types of errors namely article/determiner, preposition, noun number, subject-verb agreement and verb form errors. The University of Illinois (UI) system is a combination of five classifier models, where each model is specialized to correct a specific type of error. To correct the article errors, Averaged Perceptron (AP) model is used, which is trained on NUCLE corpus using a rich set of features generated by POS tagger and chunker. Artificial article errors were introduced in the NUCLE corpus to reduce the error sparseness. To correct other four type of errors, Naïve Bayes (NB) classifier is used which is trained on Google web 1T 5-gram corpus[39] using word n-gram features. Each individual model predicts the most probable word from its candidate set. The candidate set for articles and prepositions are (a, the , ϕ) and 12 most frequent prepositions, respectively. For noun, verb agreement and verb form errors the candidate set includes their respective morphological variants. Finally, the results of each individual classifier are combined, filtered for false alarms and then applied to correct the sentence. See figure 15

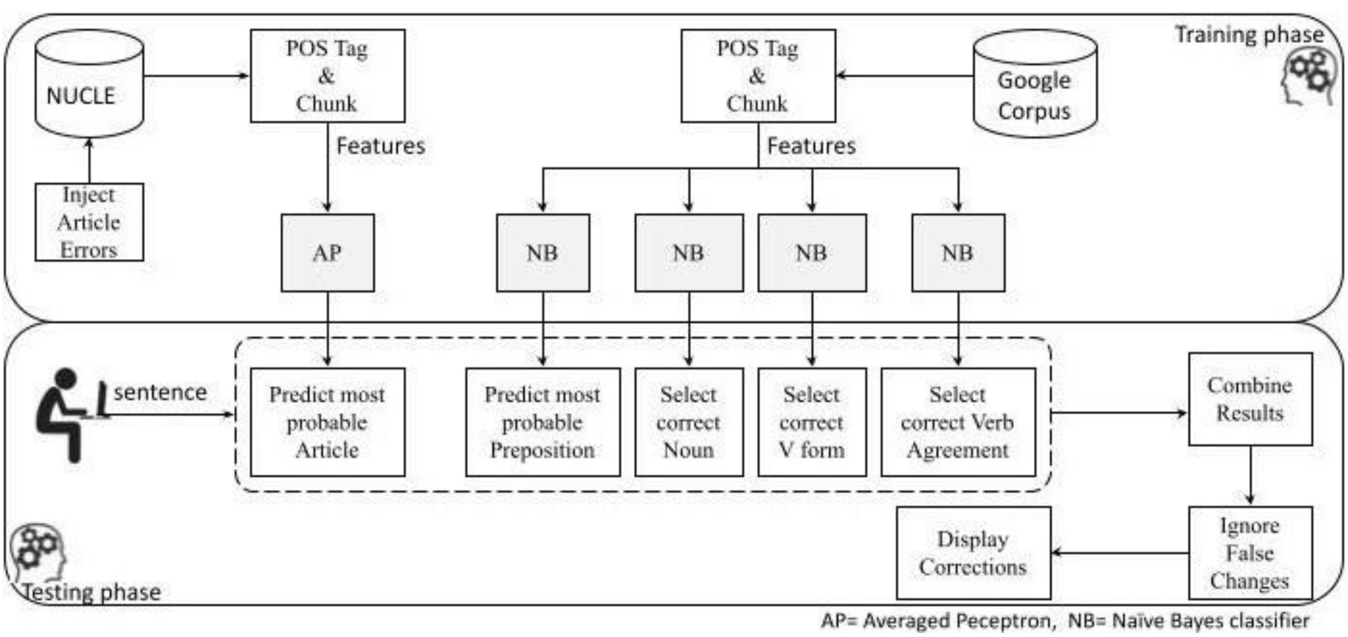


Figure 15: Schematic Diagram of UI System [13]

This system was later extended in CoNLL-2014 shared task where it was implemented to correct more types of errors and to address correction of two or more related errors by using joint inference method [40]. *Limitations:* The system corrects only 12 out of 36 prepositions. The F1-score for preposition errors (14.84) is low. The developed system is not available online.

5.12 Hybrid System : The hybrid system was developed by [7] at CoNLL-2014 shared task which combines a rule based and statistical machine translation systems in a pipeline. The rule based module automatically derives rules from Cambridge Learner Corpus(CLC) [41] that detects erroneous unigram, bigrams or trigrams and generates a list of candidate corrections. These candidates are ranked (most probable correction is top ranked) using a Language Model (LM) built from Microsoft's web n-grams. The results of the LM are pipelined into the Statistical Machine Translation system. The SMT model was trained on multiple corpora including NUCLE v3.1, 2014 shared task dataset, IELETS dataset from CLC corpus [41] , EVP corpus (http://www.englishprofile.org/wordlists) and FCE [42] corpus. The SMT model generates 10 best correction candidates which are further ranked by the language model. Next, the unnecessary corrections (corrections having error types: reordering, word acronym or run-ons) are filtered out and best correction is applied to replace the input string. See figure 16

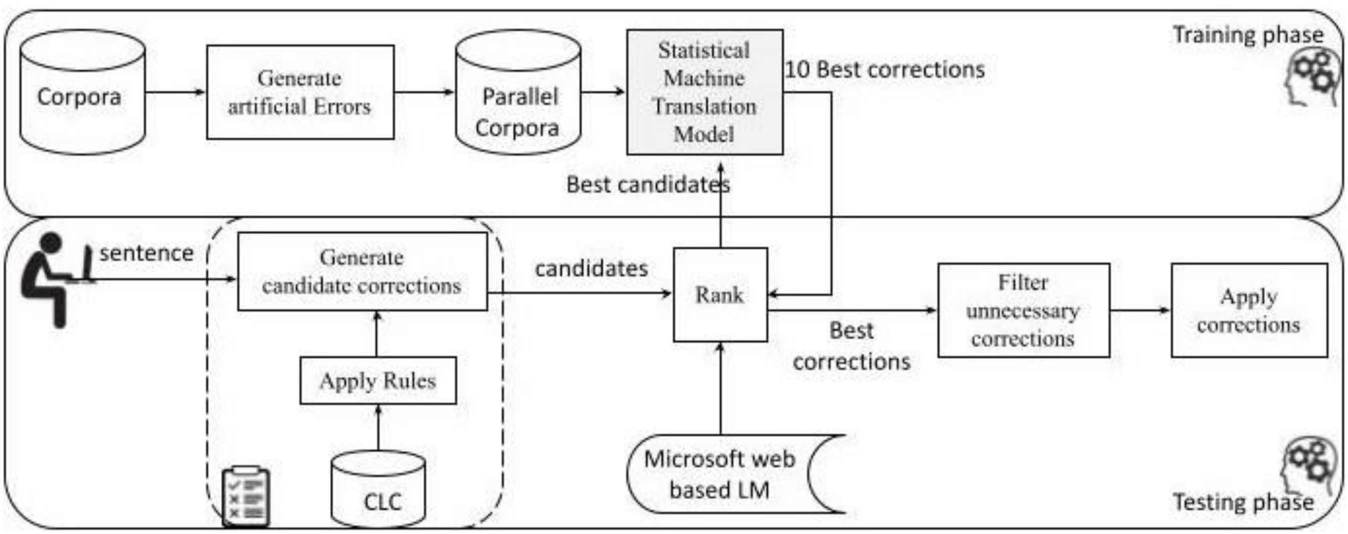


Figure 16: Schematic Diagram of Hybrid System [7]

The SMT system is developed using Pialign [43] for word alignment, IRSTLM [44] to build target language model and Moses for decoding. The system is able to correct agreement errors, verb form errors, noun number, pronoun reference, punctuation, capitalization and spelling errors. *Limitations:* The system performed poorly on sentence fragments, run-ons, word reordering and collocations errors. The developed system is not available online.

5.13 Machine Translation and Classifiers: Rozovskaya et al [27] combined the classification approach with Machine Translation. They used CoNLL-2014 training data as learner data and Lang-8 (https://lang-8.com) as native data. They trained the classifiers on customized data per error type. For example, they used prediction of a classifier trained on native data as features to train a classifier on learner data to detect article errors. Similarly, error patterns from learner data injected into native data to detect verb form errors. For noun number errors, classifier was trained on native data. MT system trained on native data was applied to the output of classifiers in a pipeline architecture and finally reported the best system with an F-score of 47.40. See figure 17

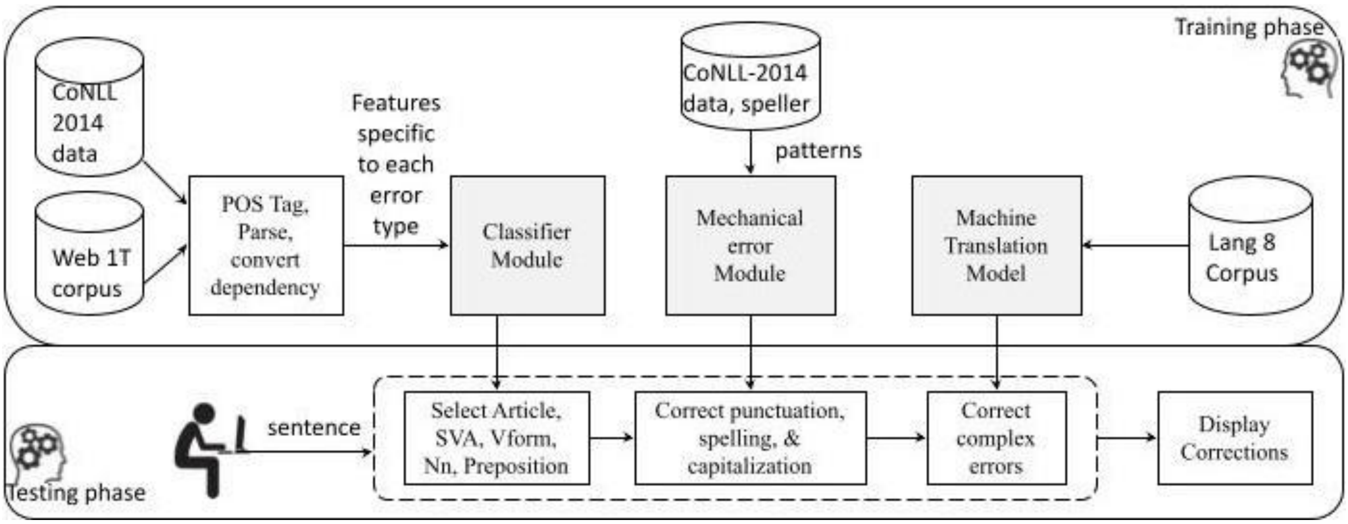


Figure 17: Schematic Diagram of Machine Translation and Classifiers [27]

This was an improvement on CoNLL-2014 shared task. The individual strengths of classifier approach (such as ability to be trained on customized features) and machine translation approach (such as handling complex errors) were utilized. *Limitations:* Individual contribution of classifiers and ML system in the correction task are not clear. The authors did not discuss the system's performance on individual errors. The developed system is not publicly available.

5.14 Frequency and Rule based approach : [28] developed a frequency based spell checker and a rule based grammar checker focusing mainly on spelling and verb tense error. The spell checker was based on detecting the misspelled word using dictionary lookup and n-gram analysis methods. Further the spelling errors are corrected by a word with minimum edit distance and highest frequency. The grammar checker was similar to LanguageTool [5] to detect only verb tense errors. See Figure 18

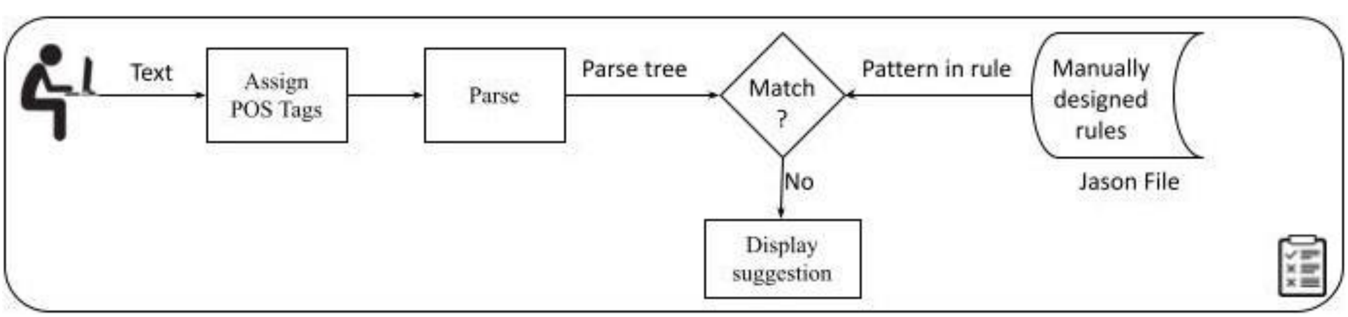


Figure 18: Schematic Diagram Frequency and rule based system [28]

The system is simple and combines all the advantages of rule hand crafting. *Limitations:* The system has no automatic correction and also does not cover all types of error.

5.15 Specialized Machine Translation: Napoles et al [29] experimented with various customization like morphological changes ,spelling corrections and phrasal substitution on Machine translation model. The SMT model was trained on Lang-8 parallel corpus which was customized by introducing artificially generated transformation rules for error correction. The rules were assigned some score by several feature functions. The features were extracted from various operations such as insertion, deletion, lavenshtein distance etc. The system is available at https://github.com/cnap/smt-for-gec with a reported performance of 52.2 M2 score. See figure 19

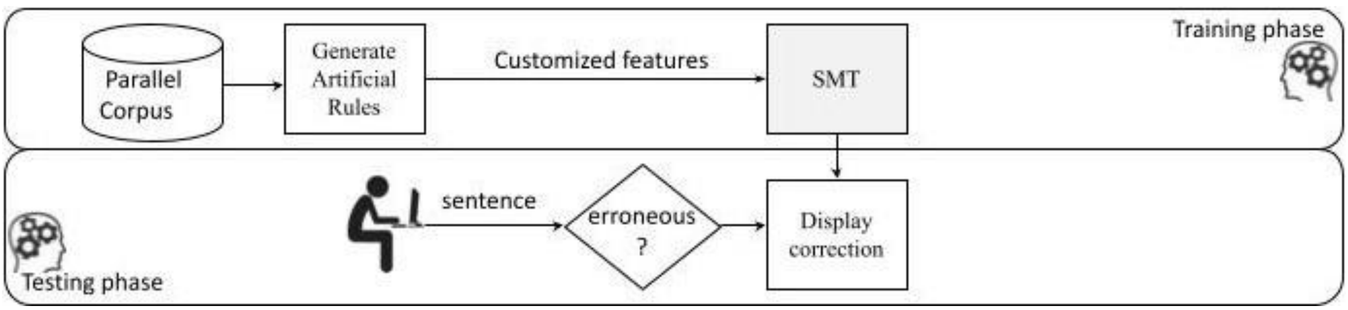


Figure 19: Schematic Diagram of Specialized Machine Translation [29]

This system successfully demonstrated that artificially generated rules improved the performance by 10%. Also large training data improves performance by negating the presence of noise. Testing was done on JFLEG corpus. *Limitations:* Even though the system was trained on a huge customized feature set, it contributed a very small performance gain.

5.16 Language Model based GEC: Bryant et al [31] developed a language based model trained on small annotated data using the fact that low probability sentence is more likely to be erroneous. So the authors calculated the normalized probability of the input text, built a confusion set and re-scored the sentence substituting each candidate in each of the confusion set in an iterative way correcting one word at a time. The sentence with probability above the threshold is applied as correction. The approach checks for morphological errors, articles, prepositions and spelling errors. The system source code is available at https://github.com/chrisjbryant/lmgec-lite. See figure 20

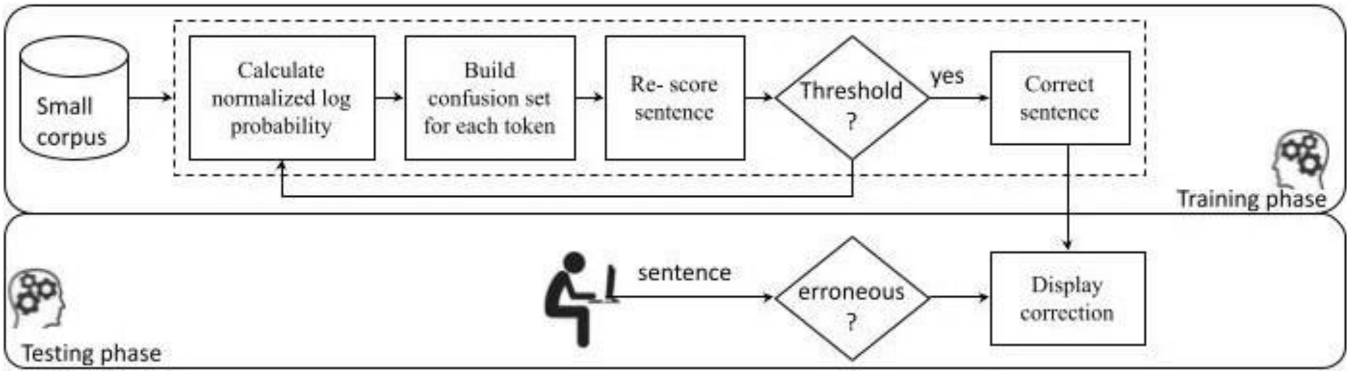


Figure 20: Schematic Diagram of Language Model based GEC [31]

The approach does not require large amount of annotated data unlike other state-of-the-art. The system achieved M2 score of 57.08 on JFLEG dataset and 60.04 GLEU score on FCE dataset. *Limitations:* The approach covers very few types of errors. The threshold used in the system is not an optimized one which may increase number of false positives.

5.17 Comparable SMT: Katsumata et al [32] developed an unsupervised SMT model trained on comparable corpus. The comparable corpus was created using comparable text pairs of English text and other language text translated into English using Google Translation. N-gram embeddings were created on both the text and mapped on shared space to create cross lingual embeddings which in turn were used to create phrase table which is further refined by the SMT model to eliminate noise. The SMT system trained on comparable corpus was applied after a spell checker to generate correct text. See figure 21

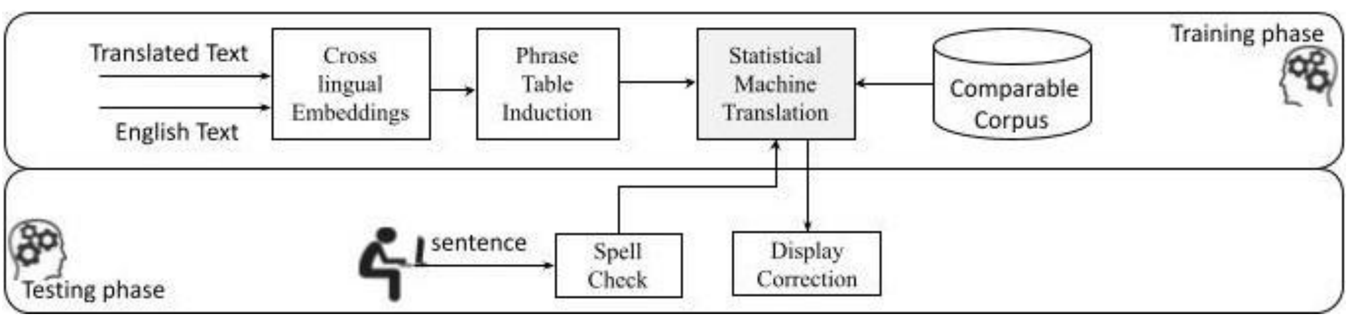


Figure 21: Schematic Diagram of Comparable SMT [32]

The system was presented at BEA2019 with an F-score of 28.31 and performed very well on spelling and punctuation errors. It uses almost 2 million text pairs for training and testing purpose. *Limitations:* The system performed poorly on noun and verb errors and also not able to detect semantic errors.

5.18 UE system at BEA: This system [30] was presented at BEA 2019 shared task, based on pre training of sequence to sequence transformer model with artificial parallel data generated using a spellchecker. The artificial data is created by introducing artificial errors in the sentence using substitution, insertion, deletion and swapping operations. Substitution is done from a confusion set which was built by calculating edit distance and phonetic equivalent distance between proposed word and the input word. Next, the transformer model was pre trained using this data. The sentence pairs were re-ranked by right to left model to improve the performance. The source code is available at https://github.com/grammatical/pretraining-bea2019. See figure 22

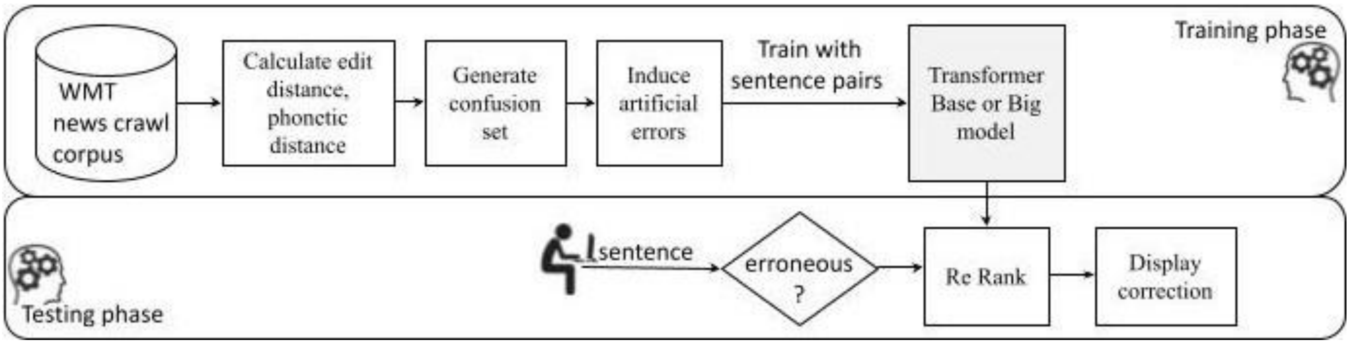


Figure 22: Schematic Diagram of UE system [30]

This was the best performing system at BEA2019 with a reported score of 69.47. The system performed very well on almost all types of targeted errors. *Limitations:* The system does not address other errors such as run-ons and semantic errors.

5.19 WER based Approach: The authors [33] proposed controlled grammar checking using Word Edit Rate. From the annotated training data, Lavenstein distance is calculated which in turn normalized with respect to the sentence length to get WER token. Five equal subset of parallel sentences annotated with WER is used to train a multilayer Convolution neural network. The model generates five candidate corrections which are ranked by 5-gram Language Model to generate the corrected sentence. See figure 23

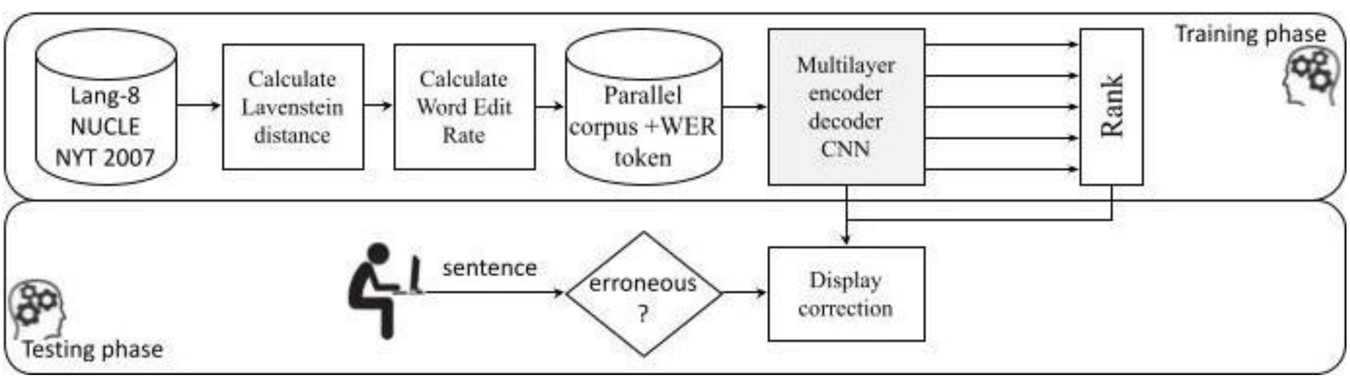


Figure 23: Schematic Diagram of WER based Approach [33]

The method controls the amount of correction required per sentence. The method seems simple and effective since WER increases recall for all target error types. It uses NUCLE, lang-8 and NYT 2007 as training set and CoNLL2014 and JFLEG as test set. *Limitations:* Recall of noun and verb errors (especially for WER token-5) were relatively low as compared to other error types. Also, it did not cover a variety of error types. The developed model is not available online.

5.20 Combined GEC: The proposed system [34] combines the results from individually trained models to achieve the optimized results. Firstly, corrected sentences are received from individual systems and M2 score files are created for each. All corrections are split into three subsets. Subset1 contains all corrections by system1 and system2 but not by system3 and system4. Subset2 contains all corrections by system3 and system4 but not by system1 and system2. Subset3 contains correction common to both S1 and S2. Again M2 files are generated for each subset and merged. A selection variable is calculated using the convex optimization algorithm with linear constraint on the merged file to display the final correction. See figure 24

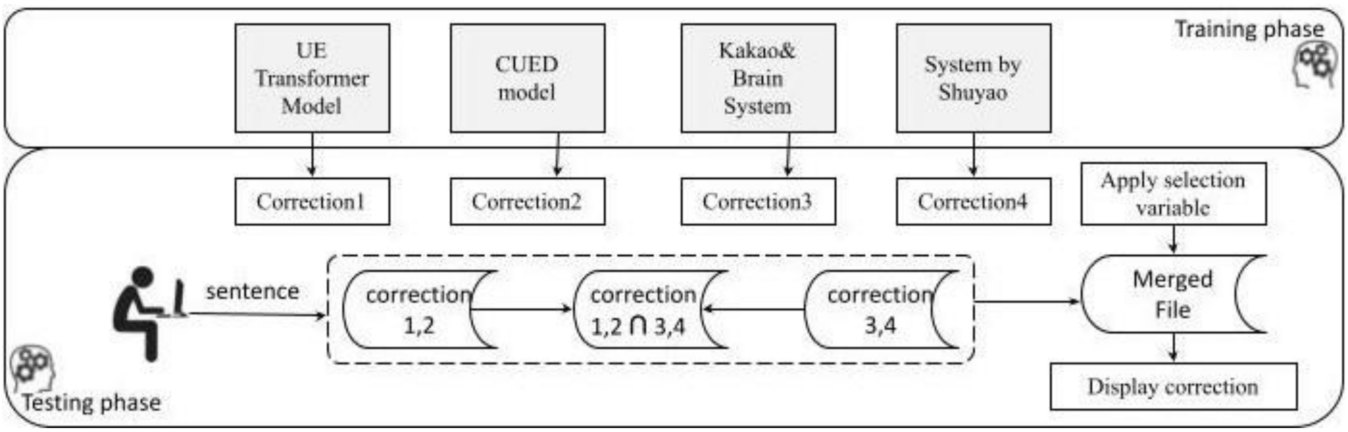


Figure 24: Schematic Diagram of Combined System [34]

This combines top four best performing systems of BEA2019 and achieved 73.18 F score. Unlike other approaches which uses either pipeline or rescoring for combination, this system directly optimized the F score by improving both precision and recall to 78.74 and 60.12 respectively. The system is useful when individual systems target different set of error types. *Limitations:* The system is not available online also the performance of the system on individual errors was not reported.

6 Research Findings

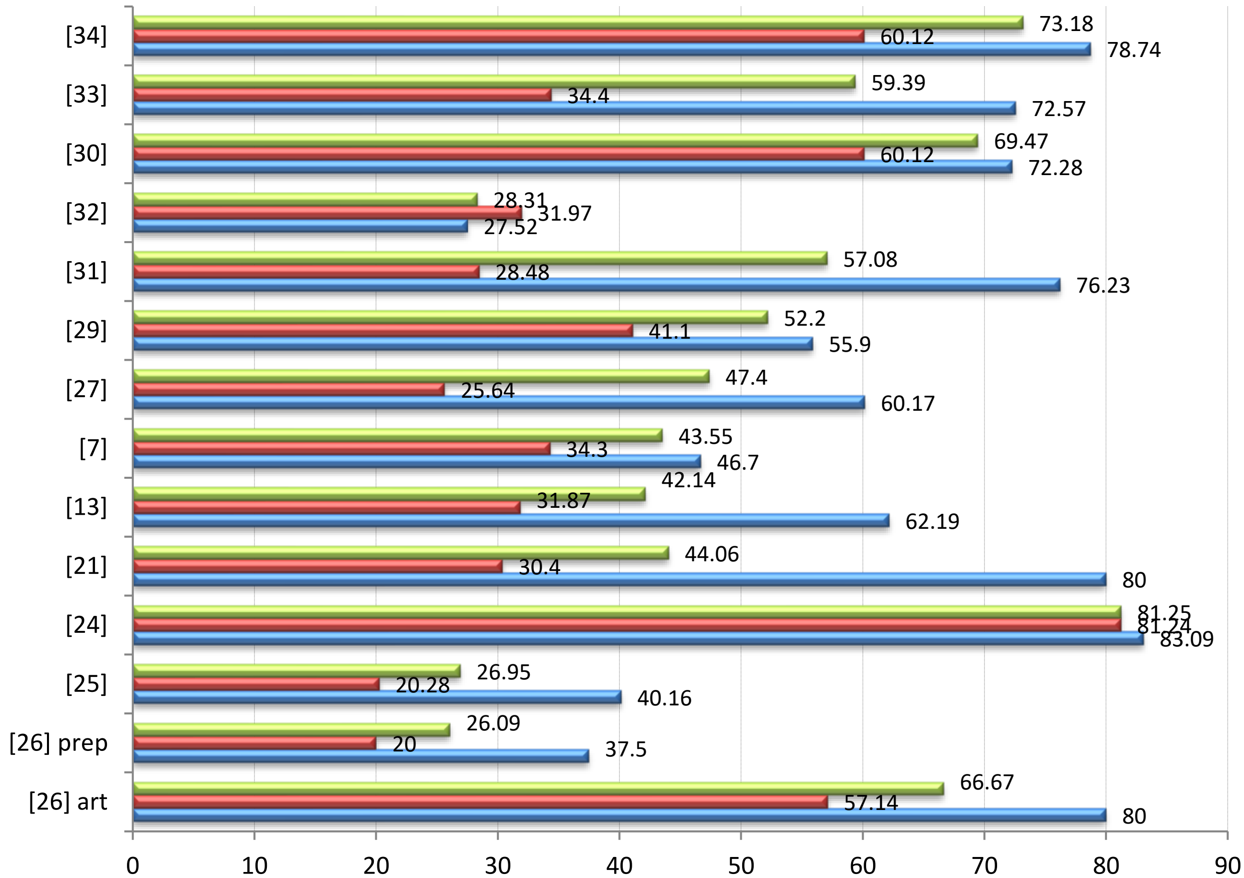
Table 2 presents the set of errors (as per our suggested scheme) detected by various approaches. We can clearly see that most of the approaches have concentrated mainly on syntax errors. Rule based systems detected mostly the spelling, subject-verb agreement, verb form/verb tense, noun number and article or determiner errors. Machine learning based systems detected mostly subject-verb agreement, verb form/verb tense, noun number, article or determiner errors and preposition errors, while the hybrid systems detected mostly subject verb agreement, verb form/verb tense, noun number, article or determiner and preposition errors. Machine learning and hybrid systems have shown better results in handling semantic level errors while very poor performance in handling sentence level errors. There are two reasons: (1) There is an imbalance in the number of error types in training and testing datasets.

Almost 78% of errors belongs to syntax or semantic categories, while only 16% contribute to Fragments and run-ons [39]. (2) Detection of semantic errors requires context knowledge and are hard to correct using pre-defined rules. This suggest that rule based systems are more suitable for handling sentence level errors and machine learning based systems are more suitable for handling semantic errors. Grammar checking using machine translation is the most recent trend that has shown much improvement. Thus we suggest development of hybrid systems combining rule based approach with machine translation to cover all error types.

Table 2: Error types detected by various Grammar checking approaches

Approach	Sentence structure Error	Fragments	Run-ons	Spelling	Syntax Error	S-V Agreements	Vform, Vtense	Noun number	Art or Det	Preposition	Punctuation	Semantic Errors	Contextual
[3]		✓		✓		✓	✓		✓				
[4]	✓			✓		✓	✓	✓					
[5]				✓	✓	✓			✓		✓	✓	✓
[22]							✓	✓	✓				
[6]								✓					
[21]										✓			
[23]						✓	✓	✓	✓				
[24]	✓	✓		✓		✓	✓		✓				
[25]				✓		✓	✓		✓	✓	✓		
[26]									✓	✓			
[13]						✓	✓	✓	✓	✓			
[7]				✓		✓	✓	✓	✓	✓	✓		
[27]				✓		✓	✓	✓	✓	✓	✓		
[28]				✓			✓						
[29]							✓	✓	✓	✓	✓		
[31]				✓					✓	✓			
[32]				✓			✓	✓		✓	✓		
[30]				✓	✓	✓	✓	✓	✓	✓	✓		
[33]						✓	✓	✓	✓	✓			
[34]				✓	✓	✓	✓	✓	✓	✓	✓		

Figure 25 presents reported performance of primary studies. The studies that did not specify results are excluded from this figure. These results are not directly comparable since they are based on different datasets, experiment settings, techniques or metrics. However, the average F-scores suggests that the machine learning technique outperforms the other two techniques.



	[26] art	[26] prep	[25]	[24]	[21]	[13]	[7]	[27]	[29]	[31]	[32]	[30]	[33]	[34]
F score	66.67	26.09	26.95	81.25	44.06	42.14	43.55	47.4	52.2	57.08	28.31	69.47	59.39	73.18
Recall	57.14	20	20.28	81.24	30.4	31.87	34.3	25.64	41.1	28.48	31.97	60.12	34.4	60.12
Precision	80	37.5	40.16	83.09	80	62.19	46.7	60.17	55.9	76.23	27.52	72.28	72.57	78.74

Figure 25: Comparison of reported performance.

7 Conclusions and Future Research

Grammar checking is the current research trend in the field of Natural Language Processing (NLP). Much work has been done for the development of grammar checking tools in the past decade. However, fewer efforts are made for surveying the existing literature. Thus, we present a comprehensive study of English grammar checking techniques highlighting the capabilities and challenges associated with them. Also, we systematically selected, examined and reviewed 20 approaches of Grammar checking. The 20 approaches can be classified into three categories namely (1) Rule based technique, (2) Machine learning based technique, and (3) Hybrid technique. Each technique has its own advantages and limitations. Rule based techniques are best suited for language learning but rule designing is a laborious task. Machine learning alleviates this labor but it is dependent on the size and type of the corpus used. Hybrid technique combines the best of both techniques but each part of the hybrid technique should be implemented according to its suitability.

Based on our detailed review of various approaches, our observations are as follows: (1) No existing approach is completely able to detect all types of errors efficiently, (2) Most of the tools are not available for research or public use, (3) All approaches use different experimental data, thus it is hard to compare the results. (4) Machine learning systems have outperformed the other two systems in terms of average F-score. (5) Most of the approaches have addressed syntax errors and its subtypes while very few efforts have been done to detect errors at sentence level and at semantic level. (6) Rule based technique will be suitable for sentence level errors and machine learning will be more suitable for handling semantic errors. (7) Detection and correction of run-on sentences is yet another untouched research area, (8) No tools have been evaluated for real time applications like proofreading of technical papers, language tutoring, and writing assistance, hence not suitable for use in these applications. (9) Our research question RQ6 is still unanswered since we could not check the results of individual error types against gold standards, (10) Research trends shows that grammar checking using machine translation is the current state-of-art.

In this paper, we have also presented an error classification scheme which identifies five broad categories of errors namely sentence structure errors, punctuation errors, spelling errors, syntax errors, and semantic errors. These errors are further subcategorized. This classification scheme would help the researchers and developers in following ways: (1) identifying the most frequent errors would tell what type of errors must be targeted for correction, (2) identifying the level of the error would tell what length of text should be examined to detect any error, (3) identifying the cause of invalid text would suggest the error correcting techniques to remove the errors. This simplifies the task of grammar checking.

Based on our observations, we suggest the following emerging research directions:

Evaluation on a standard test data: As noted in our study, all the previous approaches have been evaluated on a different test set. So, it is difficult to compare their performances. A standard test set of erroneous sentences with their well-defined correct forms should be developed. Such type of parallel corpora will benefit in reporting which systems are more efficient and robust.

Analysis based on types of errors: Based on our review, all the previous approaches deal with a different set of error types to be corrected. An annotated corpus which labels erroneous sentences into one of the five types and their subtypes, with a proper balance in number of all error categories will be helpful. Thereafter, the study of the performance of various approaches on each of these types can be carried out. This will help in identifying the best method for handling a specific error type.

Automatic error annotation: Since an annotated corpus is of huge importance in grammar error correction, the annotation process can be automated to reduce human efforts.

Coverage of different types of errors: From the data in table 2, we observed that current approaches are limited in handling all types of errors, specifically sentence structure errors and semantic errors. Future work may focus on these areas.

References

[1] N. H. Macdonald, Human factors and behavioral science: The unix writer's workbench software:

[2] Rationale and design, Bell Labs Technical Journal 62 (6) (1983) 1891-1908.

[3] M. Neuman, Rightwriter 3.1 (1991).

[4] J. C. Park, M. S. Palmer, C. Washburn, An english grammar checker as a writing aid for students of english as a second language., in: ANLP, 1997, p. 24.

[5] C. Tschichold, F. Bodmer, E. Cornu, F. Grosjean, L. Grosjean, N. Kübler, N. Léwy, C. Tschumi, Developing a new grammar checker for english as a second language, Proc. From Research to Commercial Applications: Making NLP Work in Practice (1997) 7-12.

[6] D. Naber, A rule-based style and grammar checker.

[7] C. Brockett, W. B. Dolan, M. Gamon, Correcting esl errors using phrasal smt techniques, in: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2006, pp. 249 256.

[8] M. Felice, Z. Yuan, Ø. E. Andersen, H. Yannakoudakis, E. Kochmar, Grammatical error correction using hybrid systems and type ltering., in: CoNLL Shared Task, 2014, pp. 15-24.

[9] S. Keele, et al., Guidelines for performing systematic literature reviews in software engineering, in: Technical report, Ver. 2.3 EBSE Technical Report. EBSE, sn, 2007.

[10] M. Mozgovoy, Dependency-based rules for grammar checking with languagetool, in: Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on, IEEE, 2011, pp. 209-212.

[11] J. Wagner, J. Foster, J. van Genabith, A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors, Association for Computational Linguistics, 2007.

[12] J. Lee, S. Sene, Correcting misuse of verb forms., in: ACL, 2008, pp. 174-182.

[13] Z. Yuan, Grammatical error correction in non-native english, Tech. rep., University of Cambridge, Computer Laboratory (2017).

A. Rozovskaya, K.-W. Chang, M. Sammons, D. Roth, The university of illinois system in the conll-2013 shared task, in: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, 2013, pp. 13-19.

[14] Hornby, Guide to patterns and usage in english (1995).

[15] P. Wren, H. Martin, English grammar & composition, S. Chand & Company Ltd.

[16] J. S. Thakur, A. Gupta, Identifying domain elements from textual specications, in: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ACM, 2016, pp. 566-577.

[17] J. S. Thakur, A. Gupta, Automatic generation of analysis class diagrams from use case specifications, arXiv preprint arXiv:1708.01796.

[18] Y. Yeung, J. Lee, Automatic detection of sentence fragments., in: ACL (2), 2015, pp. 599-603.

[19] J. Bigert, Probabilistic detection of context-sensitive spelling errors., in: LREC, 2004.

[20] G. Sidorov, A. Gupta, M. Tozer, D. Catala, A. Catena, S. Fuentes, Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2)., in: CoNLL Shared Task, 2013, pp. 96-101.

[21] M. Chodorow, J. R. Tetreault, N.-R. Han, Detection of grammatical errors involving prepositions, in: Proceedings of the fourth ACL-SIGSEM workshop on prepositions, Association for Computational Linguistics, 2007, pp. 25-30.

[22] E. M. Bender, D. Flickinger, S. Oepen, A. Walsh, T. Baldwin, Arboretum: Using a precision grammar for grammar checking in call, in: Instil/icall symposium 2004, 2004.

[23] A. Kumar, S. Nair, An artificial immune system based approach for english grammar checking, Artificial immune systems (2007) 348-357.

[24] G. Sun, X. Liu, G. Cong, M. Zhou, Z. Xiong, J. Lee, C.-Y. Lin, Detecting erroneous sentences using automatically mined sequential patterns, in: ACL, 2007, pp. 23-30.

[25] A.-T. Huang, T.-T. Kuo, Y.-C. Lai, S.-D. Lin, Discovering correction rules for auto editing, Computational Linguistics and Chinese Language Processing 15 (3-4) (2010) 219-236.

[26] Dahlmeier, H. T. Ng, Grammatical error correction with alternating structure optimization, in:Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 915-923.

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

[28] S. P. Singh, A. Kumar, L. Singh, M. Bhargava, K. Goyal, B. Sharma, Frequency based spell checking and rule based grammar checking, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, 2016, pp. 4435-4439.

[29] Napoles, C. Callison-Burch, Systematically adapting machine translation for grammatical error correction, in: Proceedings of the 12th Workshop on Innovative use of NLP for Building Educational Applications, 2017, pp. 345-356.

[30] R. Grundkiewicz, M. Junczys-Dowmunt, K. Heafield, Neural grammatical error correction systems with unsupervised pre-training on synthetic data, in: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2019, pp. 252-263.

[31] C. Bryant, T. Briscoe, Language model based grammatical error correction without annotated training data, in: Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, 2018, pp. 247-253.

[32] S. Katsumata, M. Komachi, (almost) unsupervised grammatical error correction using synthetic comparable corpus, in: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2019, pp. 134-138.

[33] K. Hotate, M. Kaneko, S. Katsumata, M. Komachi, Controlling grammatical error correction using word edit rate, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 149-154.

[34] Y. Kantor, Y. Katz, L. Choshen, E. Cohen-Karlik, N. Liberman, A. Toledo, A. Menczel, N. Slonim, Learning to combine grammatical error corrections, arXiv preprint arXiv:1906.03897.

[35] Izumi, T. Saiga, T. Supnithi, K. Uchimoto, H. Isahara, The development of the spoken corpus of japanese learner english and the applications in collaboration with nlp techniques, 2004.

[36] S. Gui, H. Yang, Zhongguo xuexizhe yingyu yuliaohu.(chinese learner english corpus), Shanghai: Shanghai Waiyu Jiaoyu Chubanshe.

[37] M. Marcus, B. Santorini, M. A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank.

[38] D. Dahlmeier, H. T. Ng, S. M. Wu, Building a large annotated corpus of learner english: The nus corpus of learner english, in: Proceedings of the eighth workshop on innovative use of NLP for building educational applications, 2013, pp. 22-31.

[39] T. Brants, Web 1t 5-gram version 1, http://www. ldc. upenn. edu/Catalog/CatalogEntry. jsp? catalogId= LDC2006T13.

[40] A. Rozovskaya, K.-W. Chang, M. Sammons, D. Roth, N. Habash, The illinois-columbia system in the conll-2014 shared task., in: CoNLL Shared Task, 2014, pp. 34-42.

[41] D. Nicholls, The cambridge learner corpus: Error coding and analysis for lexicography and elt, in:Proceedings of the Corpus Linguistics 2003 conference, Vol. 16, 2003, pp. 572-581.

[42] H. Yannakoudakis, T. Briscoe, B. Medlock, A new dataset and method for automatically grading esol texts, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 180-189.

[43] Neubig, T. Watanabe, E. Sumita, S. Mori, T. Kawahara, An unsupervised model for joint phrase alignment and extraction, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 632-641.

[44] M. Federico, N. Bertoldi, M. Cettolo, Irstlm: an open source toolkit for handling large scale language models, in: Ninth Annual Conference of the International Speech Communication Association, 2008.

[45] K. Korre, J. Pavlopoulos, Errant: Assessing and improving grammatical error type classification, in: Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, 2020, pp. 85-89.