Complexity Guided Active Learning for Bangla Grammar Correction

Bibekananda Kundu, Sutanu Chakraborti[1], Sanjay Kumar Choudhury
Language Technology, ICT and Services
Centre for Development of Advanced Computing, Kolkata
[1]Department of Computer Science and Engineering
Indian Institute of Technology, Chennai
E-mail: {bibekananda.kundu,sanjay.choudhury}@cdac.in,
sutanuc@cse.iitm.ac.in

Abstract

This paper introduces a novel active learning based Complexity Measurement Metric (CMM) that estimates the complexity of a sentence with respect to the grammar correction task. The CMM has been used to have an idea on how strongly users can rely on a correction suggested by a grammar checker. The proposed metric also circumvents the need of gold standard test corpora during comparison among different grammar checkers targeting different types of errors. Initially CMM has been applied on a grammar checker for Bangla language. To ensure better acceptability of CMM, the Bangla grammar checker is being evaluated for both Natural Language Generation based as well as Naïve Bayes classifier methods and the results are compared.

1 Introduction

Acquisition of foreign language is a very difficult task for second language learners. Often language acquisition is hampered due to incorrect suggestions provided by writing aids like Grammar Checker and Computer Assisted Language Learning (CALL) tools. In this context, it would be useful to devise a competence framework to alert the learners in situations where such systems are not confident. A lot of work has been done in grammatical error detection and correction, mainly in English language. However, very little work has been done for Indian languages. Probably, Punjabi grammar checker is (Gill and Lehal, 2008) is the first and only system developed for an Indian language. Alam et al. (2006) have proposed an n-gram based approach for Bangla grammatical error detection. However, there is no discussion regarding grammar correction. Moreover, none of the existing approaches have provided a measure of reliability of the system's suggestions. In this paper an initiative has been taken to provide a measures of reliability of the system's suggestions. The main contribution of the paper is a novel Complexity Estimation Matrix (CMM) that estimates the complexity of a sentence with respect to the grammar correction task. Based on this complexity score the user can have an idea how strongly he/she can rely on the correction suggested by the system. If a sentence is complex, the user should not be overtly reliant on the correction suggested by the system. Conversely if the complexity measure is low, the user can confidently choose the suggestion. In order to quantify any improvement, we need to devise a methodology for evaluating the effectiveness of a grammar checker. Due to the lack of substantially large standard test corpora, comparison among existing grammatical error detection and correction approaches is presently hindered (Leacock et al., 2010; Chodorow et al., 2012). In this paper, we have introduced a novel active learning based Complexity Measurement Metric (CMM) that estimates the complexity of a sentence with respect to the grammar correction task. The CMM has been used for evaluation of grammar checkers in order to circumvent the need of gold standard test corpora during comparison among the systems. Initially, the CMM has been applied on our Bangla grammar checker (Available at `http://14.139.223.144/nlpcdack/GrammarChecker`) based on Natural Language Generation (NLG) approach. An NLG based grammar correction technique has been reported in (Kundu et al., 2011). In this approach, an input sentence is transformed into a sequence of root words using a morphological analyser and then they are over-generated to form a trellis using a morphological synthesizer. Then linguistic hard constraints are used to prune the search space. The best correction is selected using a Language Model. Word Error Rate and BLEU (Papineni et al., 2002) score are used to ensure that the correct sentence is not too far from the input ungrammatical sentence. Since direct comparison between existing English grammar checkers and the NLG based Bangla grammar checker is not possible, the NLG based system has been compared against a prototype Bangla grammar checker based on standard Naïve Bayes classification. The Naïve Bayes classifier follows the method reported in (Golding, 1995). Four features, namely, word-word, word-tag, tag-word and tag-tag sequences have been used in this classification algorithm. The classifier has been trained on 4,68,582 well-formed Bangla unicode sentences and same number of ill-formed sentences.

2 Complexity Estimation

Complexity score (see Table 1) of an ill-formed sentence represents the amount of difficulty faced during correction of that sentence. Initially, we estimate complexity of the grammar correction for a given input test sentence. Then we find out correlation between the performance of the grammar checker and the complexity score of the test sentence. Our hypothesis is that, these correlation values will indicate how well a grammar checker performs for a given test sentence. Thus, even if two test data are not similar but have same complexity score then we can compare the performances of two systems depending on the complexity of the grammar correction problem. A significant research challenge is to estimate the complexity of a grammar correction problem in the context of a given erroneous test sentence. On the surface, this problem has some resemblances to the problem of estimating readability of text (Thompson, 2011). In the context of Bangla sentences, readability estimation has been explored by (Sinha et al., 2012). However, not all features used in the problem of estimating readability of text (Sinha et al., 2012; McCallum and Peterson, 1982; McLaughlin, 1969; Daller, 2010) are directly applicable in our case as we are dealing with erroneous text. As a result, we have introduced new features that have been explained in the next subsection.

Level of Complexity	Numerical Value
Very easy	0-25
Easy	26-50
Complex	51-75
Very Complex	76-100

Table 1: Complexity Score in different complexity levels

2.1 Feature Set of Complexity Estimation

Complexity of text can be classified as syntactic complexity and cognitive complexity. Syntactic complexity arises from elements such as sentence length, amount of embedding, and range and sophistication of structures (Lourdes, 2003; Bachman, 1976). On the other hand, cognitive complexity (Lohmann et al., 2010; Wang et al., 2012) arises from Hyperbole, Understatement and Metaphor. *"That hotel room was so small that even the mice had hunched shoulders."* is cognitive complex. Domain knowledge is required to interpret the meanings of those sentences. Here we will concentrate only on the syntactic complexity and will define syntactic/lexical features to measure cognitive load indirectly. Following features have been identified to be responsible for text and grammar correction complexity.

Presence of Comma: Presence of comma contributes to the overall readability of a sentence (Hill and Murray, 1998). Presence of comma in the appropriate places of a sentence can enhance readability and reduces the need to re-read the entire sentence. Commas also help reducing problems that arise from ambiguities. Several studies have provided evidence that readers experience difficulty when they read "garden path sentences" (Ferreira et al., 2001) like "*The old man the boat*". The garden path effect can be greatly reduced if commas are correctly present after introductory phrases and reduced relative clauses (Israel et al., 2012).

Multiple Parts of Speech (POS) of a Single Word: In most languages a particular word can have different POS. Generally, when a person reads a sentence the user builds up a likely meaning for each word and a meaning for the whole sentence word by word. At the time of sentence processing if a word appears that changes the meaning of the sentence, the user switches to the new meaning and continues. If a word has multiple POS tags and a tag which is infrequent is used in the sentence then it increases the complexity of the sentence. For example, the sentence "*The complex houses married and single soldiers and their families*" is complex because the word 'houses' is used as a verb here, which is infrequent as opposed to its use as a noun.

Syntactic Structure: If thematic roles in a sentence deviate from usual agent (do-er) before patient (do-ee) order, then patient increases cognitive load and thus increases sentence complexity. For example the sentence *"The man who killed the Tiger ⋯ "* is simpler than *"The Tiger whom the man killed ⋯"* A reversible passive sentence like *"The little rat is chased by the big cat."* is complex compared to *"The big cat chases the little rat"*. Sentence complexity using syntactic pattern can be defined as :

$$\text{Sentence Complexity (s)}=\frac{\#LV(s)+\#LN(s)}{\#Clauses(s)}$$

Where #LV(s) and #LN(s) are the number of verbal and nominal links and #Clauses is the number of clause in the sentence (Basili and Zanzotto, 2002). Coordinating conjuncts increase the complexity because relationships between clauses are not always used explicitly in the sentence.

Metaphor: One can detect metaphor by bigram analysis of noun-verb agreement. If P(Common Noun | Verb) is less than some predefined threshold then it can be considered as a metaphor. For example, *"He planted good ideas in their minds."* Here the verb *'planted'* acts on the noun *'ideas'* and makes the sentence metaphoric, thereby increasing its complexity. Generally in corpus the object that occurs more frequently with verb *'planted'* are *'trees'*, *'bomb'* and *'wheat'* etc (Krishnakumaran and Zhu, 2007).

Lexical Density: Psycholinguistic studies have long shown that less densely packed texts are more easily comprehended, particularly among non-proficient readers. Lexical density is a measure of the ratio of the number of different words to the total number of words in a text (McCarthy, 1986). In earlier work (Bradac et al., 1977), it has been shown that lexical density and comprehension test scores are strongly correlated.

T-Unit: T-Unit is an important feature responsible for text complexity. T-Unit is the "shortest grammatically allowable sentence or minimally terminable unit" (Hunt, 1965). T-Units which are longer in terms of number of words and have more subordinate clauses are more complex (Gaies, 1980). For example: *The Sun rose. The fog dispersed. The general determined to delay no longer. He gave the order to advance.* Here, the number of T-Units is 4 and the mean length of T-units = (Number of Words/Number of T-Units)=(19/4)=4.75. Consider now the single sentence rewrite: *At Sunrise, the fog having dispersed, the general, determined to delay no longer, gave the order to advance.* Here the number of T-Units is 1 and mean T-Unit length is (18/1)=18.00. It is quite obvious that the second sentence having greater mean T-Unit length is more complex than the first sentence.

Abstractness: Less frequent and abstract words increase text complexity because the presence of such words require a greater level of interpretation to understand the intended meaning.

Pronominal Reference: As pronouns are used as references and in many cases the references of the pronouns cross sentence boundaries, if the sentence starts with a particular pronoun then it is often difficult to identify the context in which that pronoun was used.

Confusion Set: The context window surrounding erroneous words and the set of possible corrections will be referred to as the confusion set henceforth. Consider a sentence S=$w_1 w_2 \cdots w_i XCY w_j \cdots w_n$. Where C is a confusion set such that $C = \{c_1, c_2, \cdots, c_n\}$ from where a particular word c_i need to be placed to make the sentence correct. X and Y are the left and right context windows of C. We will say that the given sentence is complex if $count(Xc_iY) = count(Xc_jY)+\theta$ Where $c_i \neq c_j$ and $\{c_i, c_j\} \in C$ and $0 \leq \theta \leq K$. The K is a constant. The value of K has been set empirically. Complexity of the sentence increases as the value of θ decreases and the size of the context window and the confusion set C increases. For example, the English sentence *"Ram is going C market"* is not complex when C= {*to, at*}, X={*going*} and Y={*market*}, because θ is very large as *frequency("going to market")* $\gg$ *frequency("going at market")* in general English corpus.

Other factors also can increase the complexity of the sentences like sentence length, presence of idiomatic expressions, figurative use of words and assimilation of foreign words and phrases in the source text. The variations of such features create different levels of complexity in different domains. Thus we have collected a set of features $F = \{f_i, f_{i+1}, \cdots, f_{i+n}, f_j, f_{j+1}, \cdots, f_{j+m}\}$ where the f_is are the features responsible for readability of text and the f_js features are responsible for severity of errors in text. Using these features we have designed a Multiple Linear Regression (MLR) model $\alpha_0 + (\alpha_i f_i + \cdots + \alpha_{i+n} f_{i+n}) + (\beta_j f_j + \cdots + \beta_{j+n} f_{j+n}) = \Omega$ where $\{\alpha_0, \alpha_i, \cdots, \alpha_{i+n}\}$ and $\{\beta_j, \cdots, \beta_{j+n}\}$ are the parameters of the MLR that need to be learned during training process and Ω is the complexity score. All feature values are normalized before training of MLR to account for differences between features. We collected Bangla Unicode sentences from the following online resources:

1. Bangla online news papers like *"Ananda Bazar Patrika"* (`http://www.anandabazar.com/`)

2. Online version of Bangla literatures written by Rabindranath Tagore, Sarat Chandra Chattapadhyay and Bankim Chandra Chattapadhyay (http://www.nltr.org/) published by Society for Natural Language Technology Research (SNLTR).

3. Bangla blogs (`http://www.amarblog.com/`) etc.

Special care needs to be taken at the time of selecting well-formed sentences due to different reasons. In Bangla literature, diglossic variations are found in the form of *"Sadhu"* and *"Chalit"*. Sentences written in *"Sadhu"* are mostly found writings of Bankim Chandra Chattapadhyay and writings of Rabindranath Tagore and Sarat Chandra Chattapadhyay. Sentences written in *"Sadhu"* are not used in day-to-day communication. On the other hand, most recent works follow *"Chalit"* form as it is used in daily communications. Due to informal communication, Bangla blogs contain sentences of "Benglish" (a mixed language of Bangla and English) language (Kundu and Chandra, 2012). Not all sentences of Bangla blogs are grammatically correct. For this reason, we have manually filtered the grammatical sentences from the set of sentences collected from Bangla blogs. Sentences written in *"Sadhu"* and *"Benglish"* are not important in our case, as our focus is to detect and correct grammatical sentences used in day-to-day written communication. Therefore, at the time of sentence selection from online websites (like http://www.nltr.org/, http://www.amarblog.com/ etc.), we have manually filtered out the sentences written in *"Sadhu"* form and *"Benglish"* language. In addition to that, we have collected sentences from a detective novel, namely *"Feluda Samagra"* written by Satyajit Ray. The reason behind selecting the novel is that sentences are written in *"Chalit"* and most of the sentences are simple and representative of those that are used in day-to-day communication. We have also collected sentences from *"Jekhane Dactar Nei"*, a Bengali book translated from the English work *"Where There is no Doctor"*. We have collected Unicode sentences from various domains including Literature, Sports, Health, Politics and Business (2005-2012). We assumed that the syntax and semantics of the collected sentences are correct as they are mostly collected from different newswires which are normally edited and proof-read. Corpora from multiple domains have been collected to avoid skewed distribution of data. From this set of collected 4,68,582 Bangla sentences, we have manually selected 1000 sentences having different levels of complexity to build the training data. Then we synthetically induced errors in those sentences following the methodology reported in (Kundu et al., 2012). The procedure begins with automatic creation of large number of erroneous sentences from a set of grammatically correct sentences by introducing noise using addition, deletion, substitution and transposition operations. A statistical confidence score filter has been implemented to select proper samples from the generated erroneous sentences such that sentences with less probable word sequences get lower confidence score and vice versa. A rule based mal-rule filter has been used to collect the sentences having improper tag sequences. Combination of these two filters ensure that no valid construction is getting selected within the synthetically generated error corpus. The resultant erroneous sentences had different level of error density. Then we defined the complexity score for four levels as "Very easy", "Easy", "Complex", and "Very complex". Thereafter, these erroneous sentences were given to two language experts and two native speakers for correction. We also asked them to enter a complexity score (see Table 1) according to the difficulty that they faced during correction of those sentences. Then the proposed MLR model was trained on this training dataset and the values of the parameters $\{\alpha_0, \alpha_i, \cdots, \alpha_{i+n}, \beta_j, \cdots, \beta_{j+n}\}$ were estimated. After learning the parameters of the MLR, we estimated the complexity score of five text domains (business, health, sports, literature and politics) each containing 500 erroneous sentences. We observed that the Relative Error of the MLR model is 0.39. Relative error (RE) is calculated as follows:

$$RE = \frac{1}{|N|} \sum_{i=1}^{|N|} \left| \frac{AS_i - PS_i}{AS_i} \right| \quad (1)$$

here $|N|$ is the number of test sentences. AS_i and PS_i are the actual complexity score given by user and predicted complexity score by the model. The main challenge in our research was to address the problem of high RE.

2.2 Feature Selection

While trying to analyze the cause of poor performance, we found that there are some irrelevant and redundant features which stand in the way of the accurate prediction of complexity. Large number of features have an adverse effect in efficiency and irrelevant features hamper the accurate prediction of complexity. So, there is a requirement for reduction of dimensionality by filtering the irrelevant and redundant features. Manual identification of important features from a large number of features is practically not feasible. Correlation analysis was performed on this training data to find out features which are more relevant for a particular complexity value by looking at the correlation values between the target variables and the features. Features having a low correlation (-0.1 to +0.1) with the complexity score have been removed. However, following this relevant feature selection procedure, the relative error of our multiple linear regression model becomes 0.36. The MLR model is unable to comprehensively explain the factors regarding features that contribute to the text complexity. To address this issue, a framework based on the idea of active learning has been employed for bettering our estimate of the text complexity.

3 Active Learning based Complexity Estimation

We have followed the PROTOS (Bareiss et al., 1990; Clark, 1987) architecture for active learning of grammar correction complexity for better generalization because of the need to elicit knowledge from an expert user and to provide a language specific feature that may benefit from guided explanation from linguists. The system retains the guided learning cases and also the causes of failures and the associated explanations for those specific cases. We have used the k-NN algorithm (Mitchell, 1997) for following PROTOS framework to estimate grammar correction complexity of a given input text. Initially the example-base contains examples in the form $[< f_1 : v_1 >, < f_2 : v_2 >, \cdots, < f_n : v_n >, c_i]$, where f_i is the attribute of the feature, v_i is its value and c_i is the complexity score of a sentence involving these features. For a given English sentence *"Ram *go to market"*, an example may look like $[< Num\_of\_words : 4 >, < Num\_of\_prepositions : 1 >, < Num\_of\_errors : 1 >, < Num\_of\_infrequent\_words : 1 >, 10]$. Here * indicates the erroneous word and the number 10 indicates the complexity of the sentence (The scale of the complexity score is shown in Table 1). At the time of training of the system, sentences of different complexity have been provided to the user in a multiple choice questions (MCQ) format. Then the user provides his/her choice and complexity value of sentence. This system estimates the complexity score of the same sentence based on the extracted feature values and k-NN algorithm. Given this setting, the following situations are possible.

Situation 1: User's selected option is correct and user's complexity score and system estimated complexity score is not same.

Situation 2: User's selected option is incorrect and user's complexity score and system estimated complexity score is not same.

If the user's selected option is incorrect and the complexity score provided by the user is very low and the score does not match with system generated score then the system will not ask user to supply an explanation regarding complexity of the given input sentence. In such a situation, it is

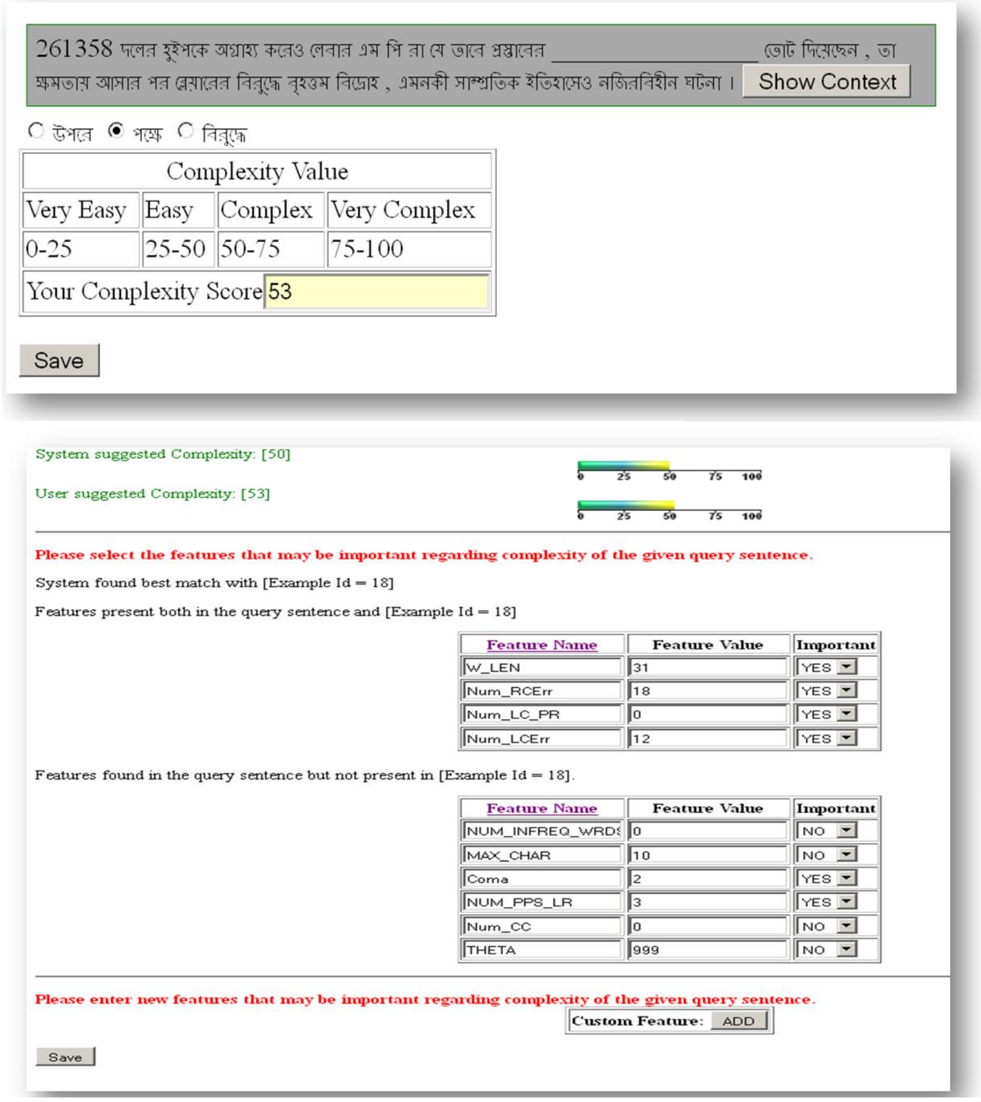

Figure 1: Screenshot of active learning framework for estimation of text complexity. The explanation of the feature names are available at `http://14.139.223.144/testComplexity/FeatDtl.spy`

assumed that the user is not confident enough to guide the system to make better inference as a part of his interaction. Other than the above mentioned situation, whenever the complexity score provided by the user and the one estimated by the system varies, the interaction based active learning procedure starts. In this case, the system provides explanations of its decision in the form of common features between the given input sentence and the nearest example from the example-base selected using the k-NN algorithm. It also provides the extra features in input sentence not present in nearest matched example or vice versa. The user then selects or adds the features that contribute to the complexity of the given sentence. After receiving the user's feedback, the system generates a new example with the selected and the newly added features. The new example is inserted into the example-base if it is not already present there. The system also remembers a link between the nearest example provided by k-NN and the new example generated depending on the user's feedback, so that whenever in future this nearest example is selected, then the system will automatically map it to the newly generated example. Our active learning based complexity estimation prototype is available at `http://14.139.223.144/testComplexity/`. The screenshot of the prototype is shown in Figure 1. The user can provide a class name (like "Very Easy", "Easy",

“Complex” or “Very Complex”) instead of entering specific complexity score.
Then the system’s generated complexity score will be mapped to one of these complexity classes. It has been seen that the relative error of the proposed active learning model is 0.16 which is much less than that obtained using multiple regression when tested on the same dataset. This clearly shows that subsequent steps have improved the system’s performance. Figure 2 shows complexity score obtained over 10 trials of each of the five domains. In each trial, we have randomly selected 50 sentences from 500 erroneous sentences of each domain and computed the average complexity score using our active learning based model. From the complexity score shown in Figure 2, it is apparent that complexity of the literature domain is higher than any other domain considered here. This is expected, since figurative uses of words are common in this domain, and nouns are ornamented with adjective and intensifiers. Rhetorical structures are usually found in sentences of literature domain. Idiomatic and colloquial patterns are used more than any other domain considered here. Sometimes phrases of foreign language are present as a part of the source language in its original orthographic representation or transliterated according to the source language. In Figure 3, we have shown POS tag distribution of five domains (business, health, sports, literature and politics). It is apparent that probability distributions of punctuation (RD_PUNC), quotative (CC_CCS_UT), subordinate (CC_CCS), personal pronoun (PR_PRP) and wh-pronoun (PR_PRQ) are appearing with higher frequency in literature domain than any other domain considered. Punctuations like multiple commas (,), semicolon (;) in a sentence represent that the sentence is a complex sentence according to the syntactic structure. Figure 4 shows the distribution of infrequent words of five domains (business, health, sports, literature and politics). In this figure we can see that the distributions of infrequent words are more in the literature domain than any other domains considered. Figure 5 shows the complexity scores obtained from 500 erroneous sentences of each domain and the respective accuracies obtained by our NLG based grammar checker. We have followed the NLG based grammar correction methodology reported in (Kundu et al., 2011). The Pearson’s correlation (Mangal, 2012) coefficient (r) of the complexity score against grammar correction accuracies obtained by the Naïve Bayes classifier and the NLG based grammar checker are -0.95 and -0.87 respectively. It shows a strong negative linear correlation of complexity scores with accuracies achieved by the two systems. Thus both classifiers have low accuracies when the complexity is high, and vice versa. This strengthens the case for the robustness of the proposed complexity measure. The proposed CMM has been integrated with the user interface of our NLG based grammar checker. It estimates the complexity score of the given input sentence and generates a colour bar (see Figure 6) to represent complexity of the sentence. Colours “green”, “blue”, “yellow” and “red” indicate different levels of complexity like “very easy”, “easy”, “complex” and “very complex” respectively. Users can rely on the suggestion provided by the system when the colour is green for a given input sentence. Conversely, “red” colour

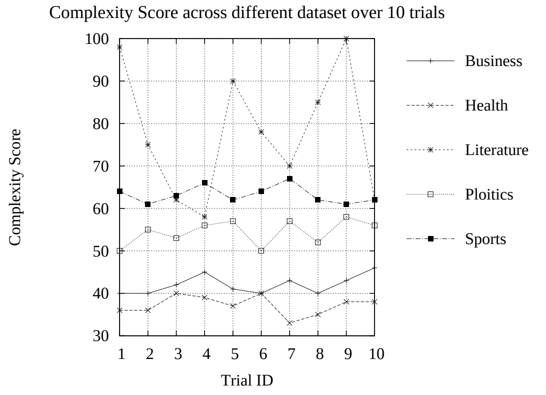

Figure 2: Complexity values across different datasets.

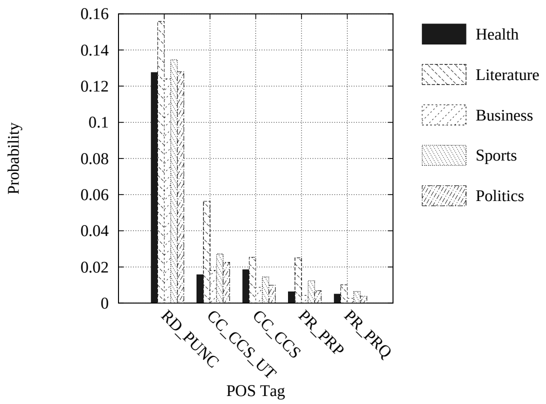

Figure 3: POS Tag distributions in different domains.

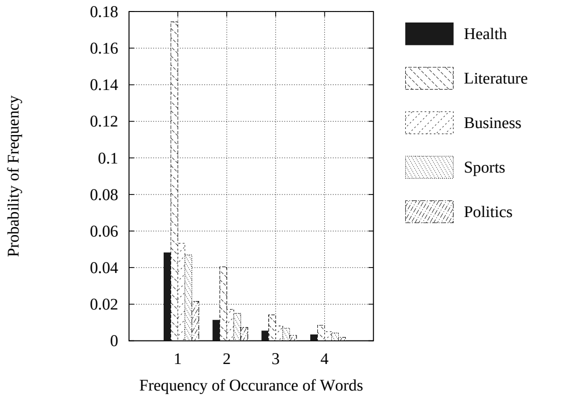

Figure 4: Frequency of word distribution across different domains.

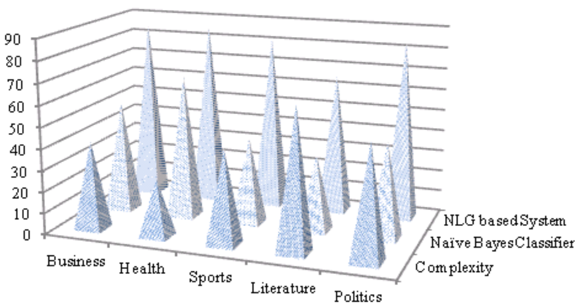

Figure 5: Complexity measure and Precision score obtained by NLG based grammar checker and Naïve Bayes classifier systems.

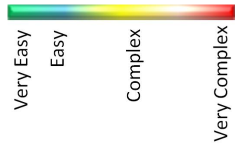

Figure 6: Colour bar indicates different level of complexity.

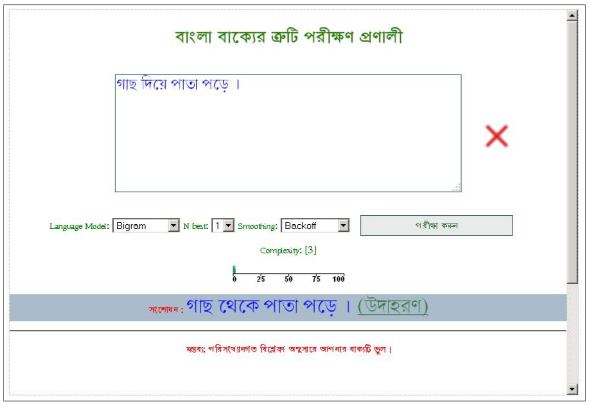

Figure 7: Screenshot of integrated online NLG based Grammar Checker showing low complexity score for a given input sentence.

alerts the user to select the system's suggestion. Figure 7 and 8 show screenshots of our integrated NLG based grammar checkers showing suggestions for "Easy" and "Complex" erroneous sentences.

4 Conclusions

In this paper, we have introduced a novel Complexity Measurement Metric (CMM) for reliable grammar correction and to alleviate the need of standard test corpus for evaluation of grammar checkers across various languages. The CMM follows active learning methodology with expert user interaction. The work reported in the paper is part of a bigger initiative driven by the need to build robust NLP tools for morphologically rich and relatively free word order Indian languages like Bangla. The implementation reported is limited by the fact that human feedback is critical to the learning process. In a practical setting, however, several users would use the system and simultaneously contribute to the process of making it more robust. The system performance is expected to gradually stabilize as the seed set of representative examples accumulate. This is true with most instance based learners, the "competence" of Case Based Reasoning systems (Leake and Wilson, 2011), for example, is expected to show only marginal improvements beyond a point. The tool developed has been made available for use online (http://14.139.223.144/testComplexity/) and we are now in a position to monitor its use over time. NLP systems based on similar ideas can be used to successfully complement CALL systems which often generate improper suggestions to the user. In that context, a competence framework would be useful in alerting the learner in situations where it is not confident. Though the work reported is exclusively focussed on robust grammar checking in the context of Indian languages, the idea of estimating the complexity of a task based on a set of appropriate features and the active learning framework that exploits the resulting complexity measure may have interesting implications for other NLP tasks. We have recently launched an exercise to evaluate the complexity of sentiment classification using a set of features which are quite different from the ones used in grammar checking and plan to use an idea

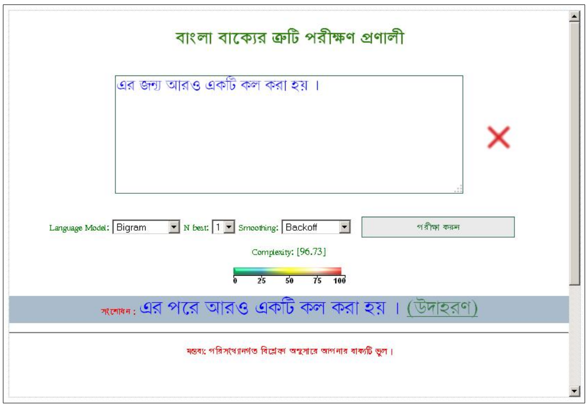

Figure 8: Screenshot of integrated online NLG based Grammar Checker showing high complexity score for a given input sentence.

inspired by the CMM framework reported in this paper to facilitate active learning in this context. Future work should also lead to exploring issues pertaining to evaluation of other grammar checking systems using the proposed methodology. We are also planning to examine the parameters in our regression model to have insights into which features are more central in determining complexity. At a later stage we may also need to study interactions between the features more closely.

References

Md. Jahangir Alam, Naushad UzZaman, and Mumit Khan. 2006. N-gram based statistical checker for Bangla and English. In *Proceedings of International Conference on Computer and Information Technology*, Dhaka,Bangladesh.

Lyle F. Bachman. 1976. *Fundamental Considerations in Language Testing*. Oxford University Press, Oxford.

Ellis R. Bareiss, Bruce. E. Porter, and Craig C. Wier. 1990. PROTOS: An Exemplar-Based Learning Apprentice. In *Machine learning and uncertain reasoning*, pages 1–13. Academic Press Ltd., London, UK.

Roberto Basili and Fabio M. Zanzotto. 2002. Parsing Engineering and Empirical Robustness. *Natural Language Engineering*, 8(2-3):97–120.

James J. Bradac, Robert A. Davies, and John A. Courtright. 1977. The Role of Prior Message Context in Evaluative Judgments of High and Low Diversity Messages. *Language and Speech*, 20(4):295–307.

Martin Chodorow, Markus Dickinson, Ross Israel, and Joel R. Tetreault. 2012. Problems in Evaluating Grammatical Error Detection Systems. In *COLING*, pages 611–628.

Peter Clark. 1987. PROTOS - A Rational Reconstruction. Technical report, Turing Institute, Glasgow.

Michael Daller. 2010. Guiraud's Index of Lexical Richness. In *British Association of Applied Linguistics*.

Fernanda Ferreira, Kiel Christianson, and Andrew Hollingworth. 2001. Misinterpretations of Garden-Path Sentences: Implications for Models of Sentence Processing and Reanalysis. *Journal of Psycholinguistic Research*, 30(1):3–20.

Stephen J. Gaies. 1980. T-Unit analysis in Second Language Research: Applications, Problems and Limitations. *TESOL Quarterly*, 14(1):53–60.

Mandeep Singh Gill and Gurpreet Singh Lehal. 2008. A grammar checking system for panjabi. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 149–152.

Andrew R. Golding. 1995. A Bayesian Hybrid Method for Context-Sensitive Spelling Correction. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 39–53.

Robin L. Hill and Wayne S. Murray. 1998. Comma and Spaces: The Point of Punctuation. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing*, New Brunswick, New Jersey.

Kellogg W. Hunt. 1965. *Grammatical Structures Written at Three Grade Levels*. NCTE Research report, USA.

Ross Israel, Joel R. Tetreault, and Martin Chodorow. 2012. Correcting Comma Errors in Learner Essays, and Restoring Commas in Newswire Text. In *HLT-NAACL*, pages 284–294.

Saisuresh Krishnakumaran and Xiaojin Zhu. 2007. Hunting Elusive Metaphors using Lexical Resources. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, FigLanguages '07, pages 13–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bibekananda Kundu and Subhash Chandra. 2012. Automatic Detection of English Words in Benglish text. In *Proceedings of 4th International Conference on Intelligent Human Computer Interaction*, pages 319–322.

Bibekananda Kundu, Sutanu Chakraborti, and Sanjay Kumar Choudhury. 2011. NLG Approach for Bangla Grammatical Error Correction. In *Proceeding of 9th International Conference of Natural Language Processing*, ICON'11, pages 225–230, Chennai,India. Macmillan Publisher.

Bibekananda Kundu, Sutanu Chakraborti, and Sanjay Kumar Choudhury. 2012. Combining Confidence Score and Mal-rule Filters for Automatic Creation of Bangla Error Corpus: Grammar Checker Perspective. In Alexander F. Gelbukh, editor, *CICLing (2)*, volume 7182 of *Lecture Notes in Computer Science*, pages 462–477. Springer.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automatic Grammatical Error Detection for Language Learners. Synthesis Lectures on Human Language Technologies*. Morgan Claypool.

David Leake and Mark Wilson. 2011. How Many Cases Do You Need? Assessing and Predicting Case-Base Coverage. In *Proceedings of the 19th international conference on Case-Based Reasoning Research and Development*, ICCBR'11, pages 92–106, Berlin, Heidelberg. Springer-Verlag.

Steffen Lohmann, Katrin Tomanek, Jrgen Ziegler, and Udo Hahn. 2010. Getting at the Cognitive Complexity of Linguistic Metadata Annotation: A Pilot Study Using Eye-Tracking. In *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society (CogSci 2010)*, Austin, TX. Cognitive Science Society.

Ortega Lourdes. 2003. Syntactic Complexity Measures and Their Relationship to L2 Proficiency A Research Synthesis of College Level L2 Writing. *Applied Linguistics*, 24(4):492–518.

S. K. Mangal. 2012. *Statistic in Psychology and Education*. PHI, India.

Douglas R. McCallum and James L. Peterson. 1982. Computer-Based Readability Indexes. In *Proceedings of the ACM '82 conference*, pages 44–48, New York, USA. ACM.

John McCarthy. 1986. Applications of Circumscription to Formalizing Common-Sense Knowledge. *Artificial Intellegent*, 28(1):89–118.

Harry G. McLaughlin. 1969. SMOG Grading - A New Readability Formula. *Journal of Reading*, 12(8):639–646.

Tom Mitchell. 1997. *Machine Learning*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Manjira Sinha, Sakshi Sharma, Tirthankar Dasgupta, and Anupam Basu. 2012. New Readability Measures for Bangla and Hindi Texts. In *COLING (Posters)*, pages 1141–1150.

Kevyn C. Thompson. 2011. Enriching Information Retrieval with Reading Level Prediction. In *Proceedings of SIGIR 2011 Workshop on Enriching Information Retrieval*, Beijing, China.

Yingxu Wang, Robert C. Berwick, Xiangfeng Luo, and Jingsheng Lei. 2012. A Formal Measurement of the Cognitive Complexity of Texts in Cognitive Linguistics. In *Cognitive Informatics Cognitive Computing*, pages 94–102.