Global Journal of Computer Science and Technology: A Hardware & Computation
Volume 14 Issue 1 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Verification of Bangla Sentence Structure using N-Gram

By Nur Hossain Khan, Md. Farukuzzaman Khan, Md. Mojahidul Islam,
Md. Habibur Rahman & Bappa Sarker
Islamic University, Bangladesh

Abstract- Statistical N-gram language modeling is used in many domains like spelling and syntactic verification, speech recognition, machine translation, character recognition and like others. This paper describes a system for sentence structure verification based on Ngram modeling of Bangla. An experimental corpus containing one million word tokens was used to train the system. The corpus was a part of the BdNC01 corpus, created in the SIPL lab. of Islamic university. Collecting several sample text from different newspapers, the system was tested by 1000 correct and another 1000 incorrect sentences. The system has successfully identified the structural validity of test sentences at a rate of 93%. This paper also describes the limitations of our system with possible solutions.

Keywords: *N-gram, sentence structure, corpus, witten-bell smoothing, word error.*

GJCST-A Classification: *D.3.2*

Strictly as per the compliance and regulations of:

Verification of Bangla Sentence Structure using N-Gram

Nur Hossain Khan [α], Md. Farukuzzaman Khan [σ], Md. Mojahidul Islam [ρ], Md. Habibur Rahman [ω] & Bappa Sarker [¥]

Abstract- Statistical N-gram language modeling is used in many domains like spelling and syntactic verification, speech recognition, machine translation, character recognition and like others. This paper describes a system for sentence structure verification based on N-gram modeling of Bangla. An experimental corpus containing one million word tokens was used to train the system. The corpus was a part of the BdNC01 corpus, created in the SIPL lab. of Islamic university. Collecting several sample text from different newspapers, the system was tested by 1000 correct and another 1000 incorrect sentences. The system has successfully identified the structural validity of test sentences at a rate of 93%. This paper also describes the limitations of our system with possible solutions.

Keywords: N-gram, sentence structure, corpus, witten-bell smoothing, word error.

I. Introduction

The goal of Statistical Language Modeling is to build a statistical language model that can estimate the distribution of natural language as accurate as possible. A statistical language model (SLM) is a probability distribution P(s) over strings S that attempts to reflect how frequently a string S occurs as a sentence. By expressing various language phenomena in terms of simple parameters in a statistical model, SLMs provide an easy way to deal with complex natural language in computer. Therefore N-gram based modeling finds extensive acceptance to the researchers working with structural processing of natural language. An n-gram model is a type of probabilistic model for predicting the next item in such a sequence. More concisely, an n-gram model predicts x_i based on $X_{i-1}, X_{i-2}, X_{i-3}, \ldots\ldots\ldots, X_{i-n}$.

In Probability terms, this is nothing but $P(X_i \mid X_{i-1}, X_{i-2}, \ldots\ldots\ldots, X_{i-n})$. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram", size 3 is a "trigram"; and size 4 or more is simply called an "n-gram". For a sequence of words, for example "the dog smelled like a skunk", the trigrams would be: "# the dog", "the dog smelled", "dog smelled like", "smelled like a", "like a skunk" and "a skunk #". N-Grams are typically constructed from statistics obtained from a large corpus of text using the co-occurrences of words in the corpus to determine word sequence probabilities. N-Grams have the advantage of be able to cover a much larger language than would normally be derived directly from a corpus. Open vocabulary applications are easily supported with N-Gram grammars [1]. Within the much application areas, an important application is to assess the probability of a given word sequence appearing in text of a language of interest in pattern recognition systems, speech recognition, OCR Intelligent Character Recognition (ICR), machine translation and similar applications [2]. By converting a sequence of items to a set of n-grams, it can be embedded in a vector space, thus allowing the sequence to be compared to other sequences in an efficient manner. The idea of n-gram based sentence structure verification has come from these opportunities provided by n-grams. Sentence structure verification is the task of testing the syntactical correctness of a sentence. It is mostly used in word processors and compilers. For applications like compiler, it is easier to implement because the vocabulary is finite for programming languages but for a natural language it is challenging because of infinite vocabulary. Three methods are widely used for grammar checking in a language; syntax-based checking, statistics-based checking and rule-based checking. In syntax based grammar checking [3], each sentence is completely parsed to check the grammatical correctness of it. The text is considered incorrect if the parsing does not succeed. In statistics-based approach [4], a corpus is used to train a model. Some sequence will be very common others will probably not occur at all. Uncommon sequences in the training corpus can be considered incorrect in this approach. In rule-based approach [5], a set of hand crafted rules is matched against a text which has at least been POS tagged. This approach is very similar to statistics-based approach, but the rules are developed manually. However, one of the most widely used grammar checkers for English, Microsoft Office Suite grammar checker, is also not above controversy [6]. It demonstrates that work on

Author α: M.Sc. & B.Sc., Dept. of CSE, Islamic University, Kushtia, Bangladesh, e-mail: nur_cse_iu@yahoo.com.
Author σ: Professor, Dept. of CSE, Islamic University, Kushtia, Bangladesh, e-mail: mfkhanbd2@gmail.com.
Author ρ: Assistant Professor, Dept. of CSE, Islamic University, Kushtia, Bangladesh, e-mail: mojahid.cse@gmail.com.
Author ω: Lecturer, Dept. of CSE, Islamic University, Kushtia, Bangladesh, e-mail: habibiucse@gmail.com.
Author ¥: Lecturer, Dept. of CSE, Islamic University, Kushtia, Bangladesh, e-mail: bappacse07@yahoo.com.

grammar checker in real time is not very easy task; so starting the implementation for language like Bangla structural verification of a sentence is a major feat. In our work, an effort has been made to develop system to verify Bangla sentence structure using statistical or more specifically n-gram based method. This is because, this approach does not need language resources like handcrafted grammatical rules, except for a corpus to train the language model (LM). Given the scarcity of language resources for Bangla, proposed approach may be the only reasonable one for the foreseeable future.

II. Techniques Adopted in the Proposed System

In statistical approach we can simply measure the probability of a sentence using n-gram analysis. For example, using bigram probability of the sentence "রহিম ফুটবল খেলে।" is,

P ("রহিম ফুটবল খেলে।") = P (রহিম | <s>) * P (ফুটবল | রহিম) * P (খেলে | ফুটবল)

To estimate the structural correctness of a sentence, we calculate the probability of a sentence using the formula above. If the value of the probability is above some threshold then we consider the sentence to be structurally correct. Now if any of these three words (রহিম,ফুটবল,খেলে) are not in the corpus then the probability of the sentence will become zero because of multiplication. To solve this problem, Witten-Bell smoothing [7] was used to calculate the probability of a sentence in our work. A sample corpus was used in this work that is a part of another corpus under construction in the speech and image processing lab of Islamic University, Bangladesh. We have developed necessary programs to assemble sequences of N tokens into N-grams. Typically N-grams are formed of contiguous tokens that occur one after another in the input corpus. If we consider a bangla sentence "আমরা যে দেশে বাস করি তার নাম বাংলাদেশ", the possible bigrams (N-grams with N=2) are: আমরা যে, যে দেশে, দেশে বাস, বাস করি, করি তার, তার নাম, নাম বাংলাদেশ

Bigram probability, P(আমরা যে দেশে বাস করি তার নাম বাংলাদেশ) = P(আমরা | <s>) * P(যে | আমরা) * P(দেশে | যে) * P(বাস | দেশে) * P(করি | বাস) * P(তার | করি) * P(নাম | তার) *P(বাংলাদেশ | নাম)

and possible trigrams (Ngrams with N=3) are:
আমরা যে দেশে, যে দেশে বাস, দেশে বাস করি, বাস করি তার, করি তার নাম, তার নাম বাংলাদেশ

Trigram probability, P(আমরা যে দেশে বাস করি তার নাম বাংলাদেশ) = P(আমরা | <s1><s2>) * P(যে | <s1> আমরা) * P(দেশে | আমরা যে) * P(বাস | যে দেশে) * P(করি | দেশে বাস) * P(তার | বাস করি) * P(নাম | করি তার) *P(বাংলাদেশ | তার নাম)

Similarly, the possible quad-grams (N-grams with N=4) are:
আমরা যে দেশে বাস, যে দেশে বাস করি, দেশে বাস করি তার, বাস করি তার নাম, করি তার নাম বাংলাদেশ

Quad-gram probability, P(আমরা যে দেশে বাস করি তার নাম বাংলাদেশ) = P(আমরা | <s1><s2><s3) * P(যে | <s1> <s2> আমরা) * P(দেশে | <s1> আমরা যে) * P(বাস | আমরা যে দেশে) * P(করি | যে দেশে বাস) * P(তার | দেশে বাস করি) * P(নাম | বাস করি তার) *P(বাংলাদেশ | করি তার নাম)

After training a model using above concept it was used to design a test system. For the purpose of testing whether a sentence is correct or not, the number of N-grams (2, 3, or 4) in the sentence was counted first. Using all the N-grams of the sentence, we have generated a score for the sentence. If the score is greater than a predefined threshold, the sentence is syntactically correct. On the other hand, if the score is not greater than the threshold, the sentence is syntactically incorrect.

III. Training the N-gram Model

The first step to compute N-grams is counting unigrams. The unigram count and necessary software tools was ready in the laboratory and the work was started from bigram count. After updating the existing software tools bigrams, trigrams and quad-grams were identified, counted and stored in separated disk files. In all cases input to the software was the sample corpus contained in file corpus.txt. The outputs are shown in figure-1(a) & 1(b)

আবার জমজমাট রাজনীতি।
দীর্ঘদিনের জড়তা কাটিয়ে
ফের সরগরম রাজপথ।
আবার জমজমাট রাজনীতি।

Unigram	Frequency
আবার	২
জমজমাট	২
রাজনীতি	২
দীর্ঘদিনের	১
জড়তা	১
কাটিয়ে	৩
ফের	১
সরগরম	১
রাজপথ	১

Figure 1(a) : Samples of first step computation

Bigram	Frequency
আবার জমজমাট	২
জমজমাট রাজনীতি	১
দীর্ঘদিনের জড়তা	২
জড়তা কাটিয়ে	১
কাটিয়ে ফের	১
ফের সরগরম	১
সরগরম রাজপথ	১

Trigram	Frequency
আবার জমজমাট রাজনীতি	২
দীর্ঘদিনের জড়তা কাটিয়ে	১
জড়তা কাটিয়ে ফের	২
ফের সরগরম রাজপথ	১

Quadrigram	Frequency
দীর্ঘদিনের জড়তা কাটিয়ে ফের	১
জড়তা কাটিয়ে ফের সরগরম	২
কাটিয়ে ফের সরগরম রাজপথ	১

Figure 1(b) : Samples of first step computation

Bigram	Probability
আবার জমজমাট	০.০০০৬
জমজমাট রাজনীতি	০.০০০৩
দীর্ঘদিনের জড়তা	০.০০০৬
জড়তা কাটিয়ে	০.০০০৩
কাটিয়ে ফের	০.০০০৩
ফের সরগরম	০.০০০৩
সরগরম রাজপথ	০ ০০০৩

Trigram	Probability
আবার জমজমাট রাজনীতি	০.০০০৪
দীর্ঘদিনের জড়তা কাটিয়ে	০.০০০২
জড়তা কাটিয়ে ফের	০.০০০৪
ফের সরগরম রাজপথ	০.০০০২

Quadrigram	Probability
দীর্ঘদিনের জড়তা কাটিয়ে ফের	০.০০০২৬
জড়তা কাটিয়ে ফের সরগরম	০.০০০১৩
কাটিয়ে ফের সরগরম রাজপথ	০.০০০২৬

Figure 2 : Sample results of second step computation

In the second step of computation, outputs of the first step were used as inputs. A set of program modules were developed to compute bigram, trigram and quad-gram probabilities using N and N-1 gram count. For example, bigram probabilities were calculated by using unigram and bigram counts. The intermediate results of the system as the outputs of the second step are shown in figure-2.

IV. The Test System

For the purpose of testing whether a sentence is correct or not, at first, all the number of bigrams of the sentence was counted. Getting probabilities from the respective models, Witten-Bell smoothing was applied to compute a set of probabilities contained all nonzero values. Multiplying all the bigrams of the sentence, a score for the sentence was generated. If the score is greater than a predefined threshold, the sentence is syntactically correct. The functional block diagram of the system is shown in figure 3. For the trigram or quad-gram models, the same algorithm was followed by replacing only the bigrams with trigrams or quad-grams respectively.

V. Experimental Results and Discussion

In our experiment, 1000 sentences collected from the web edition of a daily newspaper to form a test set. The test set was disjoint from the training corpus. All of these 1000 sentences were structurally correct. Taking these correct sentences as input, the result generated by the test system is shown in table–1. For another experiment, All of these 1000 sentences were modified to make structurally incorrect and presented again as input to the test system. The result generated by second experiment is also shown in table-1.

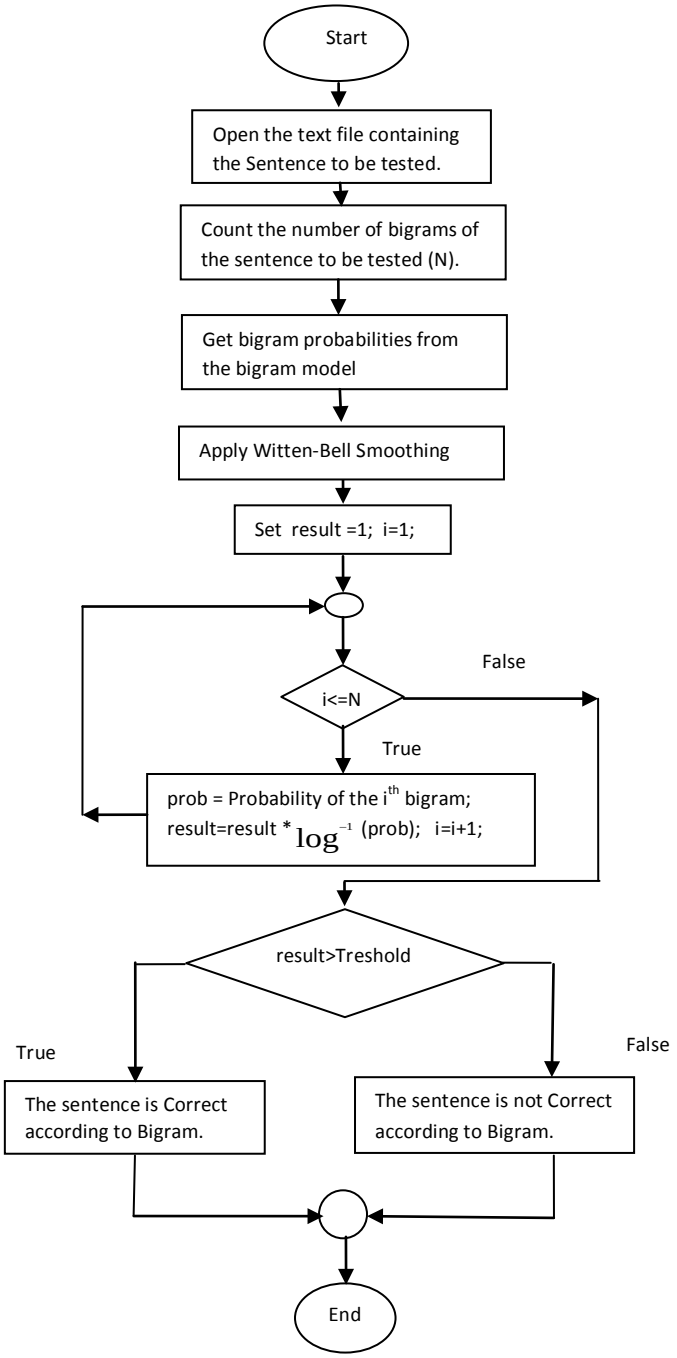

Figure 3 : Block Diagram of the Bigram Model of the System.

Table 1 : The test result with correct and incorrect sentences.

Results with correct sentences			
Models	No. of Sentences	No. of success	Performance
Bigram	1000	900	90%
Trigram	1000	905	90.5%
Quadrigram	1000	907	90.7%
Results with incorrect sentences			
Bigram	1000	950	95%
Trigram	1000	961	96.1%
Quadrigram	1000	963	96.3%
		Average	93.1%

VI. Discussion

The word-error in Bangla can belong to one of the two distinct categories, namely, non-word error and real-word error. A string of characters separated by spaces without a meaning is a non-word. By real-word error we mean a valid but not the intended word in the sentence, thus making the sentence syntactically or semantically ill-formed or incorrect. The developed system can identify both types of errors with an failure rate of 6.9% on average. The major cause of this error is the volume of training corpus. As large as the volume of training corpus so will be success rate.

VII. Conclusion

We have developed a statistical Sentence structure verifier for Bangla, which has a reasonably good performance as a rudiment Sentence verifier. By increasing the volume of training data the performance of the system can be improved and a hybrid system combining both statistical and rule based system can be developed.

References Références Referencias

1. Michael K. Brown, Andreas Kellner, and Dave Raggett, "Stochastic Language Models (N-Gram) Specification", W3C/Openwave, http://www.w3.org/TR/ngram-spec , Access date: 8th Dec. 2010.
2. Wikipedia, "n-gram", http://en.wikipedia.org/wiki/N-gram, Access date: 17th Dec. 2010.
3. Karen Jensen, George E. Heidorn, Stephen D. Richardson (Eds.), Natural Language Processing, the PLNLP approach, 1993. W.-K. Chen, Linear Networks and Systems (Book style).Belmont, CA: Wadsworth, 1993, pp.123–135.
4. Eric Atwell and Stephen Elliott, Dealing with ill-formed English text, The Computational Analysis of English, Longman, 1987.
5. Daniel Naber, A Rule-Based Style and Grammar Checker, Diploma Thesis, Computer Science - Applied, University of Bielefeld, 2003.
6. Sandeep Krishnamurthy, A Demonstration of the Futility of Using Microsoft Word's Spelling and Grammar Check, available online at http://faculty.washington.edu/sandeep/check.
7. Daniel Jurafsky, James H. Martin,"Speech and Language ProcessingAn Introduction to Natural Language Processing: Computational Linguistics and Speech Recognition", Prentice Hall, Englewood Cliffs, New Jersey 07632 , September 28, 1999.

This page is intentionally left blank