

CSE 431

Automated Part Of Speech Tagger for Bangla

**Submitted
By**

**M. Hammad Ali
Id: 02201076**

**Fahim Muhammad Hasan
Id: 03101057**

Problem Statement

The objective of our project was to implement an automated Bangla Part-Of-Speech (POS) tagger. We could implement the algorithms we learnt as part of the course material, or look into the topic further for any algorithm that could be more advanced or provide better performance than the ones we already know of. We used the python programming language and the NLTK LITE language library for the purpose of the implementation. Final submission entailed a working program that would be able to tag an untagged Bangla corpus to a reasonable degree of accuracy.

Literature Review

In this section we highlight some of the work that has already been done in this area, in Bangla as well as in other languages. We start with a generic discussion of certain algorithms, and then highlight more specific information that we found during the literature review phase of the project.

In general, there are three different approaches to the problem of assigning each word of a text with a tag. These are the statistic approach, the rule-based approach and the transformation-based learning approach. Rule-based taggers incorporate the use of a set of hand-written rules in order to try and assign a tag to each word. These rules could specify, for instance, that a word following a determiner and an adjective must be a noun. Of course, this means that the set of rules must be properly written and double-checked by human experts. The statistical approach uses a training corpus to pick the most probable tag for a word. One instance of such an approach is the Hidden Markov Model. The transformation-based approach combines the rule-based approach and statistical approach. It picks the most likely tag based on a training corpus and then applies a certain set of rules to see whether the tag should be changed to anything else. It saves any new

rules that it has learnt in the process, for future use. One example of an effective tagger in this category is the Brill tagger.

We reviewed some papers on the Brill tagger and implementations of the Hidden Markov Model. Most of these implementations were for languages like English and Dutch, but there were also some for languages like Hindi, Tamil and Bangla. We also went through some papers that discussed possible methods of POS tagging using not language-specific but universal features shared by several languages, although due to time and scope restrictions we did not get to explore this issue in proper depth. Our main focus within these papers was on possible hints we could find on how to implement an HMM tagger for Bangla, since the source code found within the NLTK LITE module was not transparent enough. Unfortunately, the theoretical knowledge that we gathered was identical to what we had already as part of coursework, and not much was covered about the implementation of the algorithm. So at that point, we decided to try and enhance the Brill tagger to a greater degree of accuracy.

Our solution

As a starting point for our project, we were given a code written by our classmate Jahangir Alam. His code basically enabled us to use NLTK to access a Bangla corpus in UTF-8 format, and use certain built-in methods to manipulate the corpus. For the implementation of this code, he had to use the brown corpus methods as reference. Using Jahangir's code and the examples provided in the NLTK tutorial, we then used a set of words from a tagged Bangla corpus to train a unigram and bigram tagger. These taggers were then applied on an untagged Bangla text and the accuracy checked using built-in methods that we learnt during the lab work. At this point, we were not sure of the quality of the training corpus or the effectiveness of taggers trained using this corpus. At first, the unigram tagger attained an accuracy of 69% and the bigram tagger managed 76%. The first transformation-based Brill tagger implementation got an accuracy rate as high as

80%. After this phase, the accuracy was not too satisfactory and we had certain problems that we needed to solve. For one, the Brill tagger template we were using assigned a default tag of none, while we wanted the default tag to be NP. So we had to resolve that problem. We would also have to start considering the possibility of a hybrid solution and evaluate it using NLTK's accuracy libraries.

By the second phase, we had managed to improve the accuracy by discarding the unnecessary rules. The unigram tagger now averaged 80%, the bigram tagger 82% and the Brill tagger managed 85%. We had also managed to use the hybrid solution in NLTK. At this point, we detected some discrepancy in the corpus that we had been using so far for training purposes. So we decided to manually edit the corpus and clarify whatever inconsistency there was originally, with the hope that a better quality training corpus would push up the accuracy rate considerably higher. We would also have to keep in mind the possibility of a working implementation of HMM, although the literature review we had done up to this point had not given us any satisfactory directions towards that end. Lastly, we were asked by our supervisor to take a similar-sized English corpus and evaluate its performance using the NLTK libraries. This would let us compare the performance of our taggers to English taggers that use the same size of training corpus.

In the next phase, we first reported on the accuracy comparison between the English and the Bangla corpus. Below we give the accuracy figures at this stage of the work:

Using a 3000+ word Bangla corpus that we had been using so far:

Unigram – 76%

Bigram – 77%

Brill tagger – 86%

Using a 3000+ word corpus from the Brown corpus:

Unigram tagger – 72%

Bigram tagger – 72%

Brill tagger – 86%

So we could now claim that for a similar sized corpus, our work on Bangla tagging was nearly as effective as the existing solutions for the English language. Following this, we wanted to try certain experiments, as follows:

After editing the Bangla corpus a little:

Unigram – 77%

Bigram – 77%

Brill tagger – 87%

On using a reduced tagset:

Unigram – 86%

Bigram – 87%

Brill tagger – 91%

At this phase, then, we were confident that with a well-tagged corpus of reasonable size, we would be able to hit around 95% level of accuracy. Since there still did not seem to be sufficient material to attempt an implementation of the Hidden Markov Model, we thus decided to focus our attention on improving the quality of the corpus.

After we had manually checked and tagged the training corpus to purge it off the discrepancies we had mentioned before, we ended up with one of 2400 words. On using this corpus for training purposes, we arrived at the following accuracy figures:

Unigram – 65%

Bigram – 65%

Brill tagger – 82%

Now, our immediate target is to try and increase the size of the training corpus to around 5000 words. With that corpus, we will evaluate the taggers one last time. Whatever accuracy figures we reach using that corpus will be our final submission within the scope of this project.

Analysis of our solution

In this section we evaluate our solution from a neutral perspective, trying to highlight if and why it is better than other work done in this area, as well as mention areas that could use some more work and thus pose a future area of research.

The amount of work that we have done is actually much more than we had expected to achieve within the short span of time we had to work on this project. The amount of accuracy achieved was much more than we expected to achieve. In many cases, our solution seemed to be working better for the Bangla language than corresponding solutions for the English language. The size of corpus was always one obstacle in our task. If we had a bigger corpus tagged to a higher degree of accuracy, we would probably be able to achieve a greater degree of accuracy. Of course, there is a threshold to just how much we can achieve by that approach. One can only process so much data, and simply increasing the size of the corpus without adding any fine tunings in the code would not really work beyond a certain point. Reducing the number of tags in the tagset also helped increase the percentage accuracy, but any rich language cannot be adequately tagged with distinctions too broad. One has to go into the matter deeper, and in this case that means a greater number of tags must be allowed. So, the probability of getting a tag right has to be increased in ways that do not rely too heavily on a bigger corpus or a smaller tagset.

Another drawback of our solution is that we could not get to the point where we could present an adequate implementation of the Hidden Markov Model. The literature we

found on the topic, as already mentioned, was theoretically adequate but not sufficient, at least at this level of our work, to manage a practical work on the topic.

Future work

In the future, we would want to do some more research on the topic and find out where we stand in terms of the percentage accuracy achieved so far. We want to find ways to achieve the threshold accuracy achieved by the Brill tagger, since that has worked remarkably well for us up to this point. However, we would also want to move ahead and try a working implementation of the HMM algorithm, something that we could not really approach within this semester. From there on, it would most likely be further efforts to try and improve the performance of the HMM solution, tweaking the code here and there to move the accuracy rate higher up. As an interesting aside, we might want to look into the work that is being done in order to get to a generic approach to POS tagging that depends on universal language constructs and not language-specific details. Finally, our last goal towards this end would be to try and single out any one or a class of algorithms that turns out to be the best as far as the Bangla language is concerned. How much of this work we can get done, only time will tell. Nevertheless, whatever amount we achieve will set a good foundation upon which further work in this area can proceed.

Reference

- Daniel Jurafsky, James H. Martin – Speech and Language Processing (Chapter 8: Word Classes and Part-of-speech Tagging)
- Eric Brill, 1992. A simple rule-based part of speech tagger
- Eric Brill, Some Advances in Transformation-Based Part of Speech Tagging

- Eric Brill, A Report of Recent Progress in Transformation-Based Error Driven Learning
- Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu, A Hybrid Model for Part-of-Speech Tagging and its Application to Bengali
- Manish Shrivastava, Part of Speech Tagging of Indian Languages using Hidden Markov Model
- Goutam Kumar Saha, Amiya Baran Saha and Sudipto Debnath, Computer Assisted Bangla Words POS Tagging
- Doug Cutting, Julian Kupiec, Jan Pedersen and Penelope Sibun, A Practical Part-of-Speech Tagger
- Eric Brill, Unsupervised learning of disambiguation rules for part of speech tagging
- Manish Shrivastava, Part of Speech Tagging of Indian Languages using Hidden Markov Model