REPORT

Generating monolingual dictionary using corpus based paradigm matching

Mentor : Harsh Vardhan GSV

The aim of this project was to expand the bangla dictionary. Bangla dictionaries already exist but they are pretty small right now and need explansion. Our project was a step in this direction. We can easily expand the dictionary, given a corpus.
This project uses an already existing dictionary as a reference dictionary to add new words to the dictionary. The words which are not found in the reference dictionary are analysed and their paradigm table is predicted and hence an entry is generated in the dictionary for them. So, in a way we are expanding the already existing dictionary.

Our approach is corpus-based paradigm matching.

Dictionary acts as a useful tool in many domains of linguistics. If we already know the forms of a word, it becomes easier to annotate the data based on contextual information.

- First, we found a large bangla corpus.
- Our next task was to annotate the corpus. For that, we used the Postagger by LTRC. To suit the input format for the tagger, we had to tokenize the data by a tokenizer, downloaded form ltrc.iiit.ac.in.
- Next task was to extract the Nouns from the tagged data. We wrote a python script for that.
- Given, an input, we first try to look for it in our existing dictionary. If we find an entry for the word in our dictionary, we generate its word forms based on the referenced paradigm table and print them.
- If we encounter a word that we can't find in our existing dictionary, we follow

a simple rule to predict the paradigm table for that word.

Rule: We categorize the words based on their endings, i.e. we say that if a word ends with a sound, say 'a', we can assume that its paradigm table will match with one of the already existing tables for words that end with 'a'. This simple rule shortens the list of tables that we need to try out fot our word.

- In order to narrow down on one paradigm table for the word, we generate the word forms corresponding to all the paradigm tables and then for each one of them, we go back to the original corpus and start looking for entries of the generated forms in our corpus.

- The paradigm table that has the maximum matchings in the corpus is assumed to be the one for the word and the user is notified that it is most probable that the given word follows that paradigm table.

- In case, where multiple tables get the the same number of matchings in the corpus, we display all the tables, because that is the maximum narrowing down we could do with our given data.

- In case we cannot match any of the generated forms to the ones in our original corpus, we can say that we cannot predict the paradigm table for the given word.

We found that most of the words matched with one or the other of the already existing paradigm tables, but still there was a large number of words which could not be mapped to a single paradigm table and hence their paradigm tables are still doubtful, but such discrepancies can be resolved if we use a bigger corpus.

Similarly for words which could not be mapped to any of the tables, we can say for sure that given a larger corpus, we will definitely be able to find a paradigm table which describes its word forms.

Thus, this is a **self improving** dictionary. As we expand our corpus, we tend to get more data and hence can predict the paradigm tables better.

A major limitation of our approach is that we consider the set of paradigm tables in our existing dictionary to be exhaustive. This restricts us to the given set of paradigm tables and hence for words which cannot be mapped to one of these tables, we need another set of paradigm tables.

But there's a solution to that too. When we are provided with a substantially large corpus, we may get all the forms for the word in our corpus only and hence we can create a paradigm table with the help of a native speaker for that language.

Usefulness of a paradigm based monolingual dictionary

A paradigm is a set of all forms of a word in a language. Since many words have same word forms of each type, many words can fall in the same paradigm.

It's useful in a way that we have to write code for only one of the words belonging to a paradigm and for the rest of the words following that paradigm only a one-line code that marks the association is enough.