# Brill's Rule Based Tagger for Hindi

Abhigyan Ghosh 20171089

International Institute of Information Technology

## Project Description

Rule based taggers depends on dictionary or lexicon to get possible tags for each word to be tagged. Hand-written rules are used to identify the correct tag when a word has more than one possible tag. Disambiguation is done by analysing the linguistic features of the word, its preceding word, its following word and other aspects.

Brill's tagger is one such rule based tagger which was implemented for English. We now take that same idea and try to apply it for Hindi.

## Theory

Given an input string, Brill's tagger first assigns the most frequent tag to each word. If a particular word is not present in the corpus, it assigns a tag based on the most probable tag for that word based on its morphology. For example: ‘’. Then it applies a given set of context based rules to try and improve on the correctness of the tags. The tags are chosen from a set of rules that may or maynot make linguistic sense.

## Method

### Transforming given data to retrievable format

The set of tagged sentences is first broken down down into word-tag pairs and then counted for their number of occurrences with each tag and sorted in descending order to find the tag that occurs with a given word the maximum number of times. So for each word we can just run a linear search for the first tag with the given word in the new sorted data set.

### Linear Search to find the tag

Since the entries in the new data set are sorted, we can then just run a linear search to search for each entry in the dataset and as soon as we find a search for the word, we return the corresponding tag which will automatically be the most frequently used tag for that word.

### Assigning POS

When the corresponding tag has been received from the file, we add it to an array of POS tags. We also maintain an array of words in the sentence. If no POS was returned on search, we go in for assigning tags based on basic word morphology. We look at the last few characters of the word and try to match it with commonly used gender or number inflections, common verb endings, etc. For example: suppose 'खेलता' was not present in our dataset but 'ता' suffix is usually found in verbs, so it would be assigned a verb. If none of the patterns match, then we just assign the word a the proper noun tag.

### Applying contextual rules

To apply contextual rules, we can just translate each rule as a manipulation of array elements. For example: To implement the rule:

VM VAUX PREV-TAG-IS VM

we can convert it to the following code snippet:

```
if POS[i]=="VM" and POS[i-1]=="VM"
then
     POS[i]="VAUX"
```

This makes it considerable easier to translate the rules to separate functions.

**Finding rules from the errors**

To find some rules, we first look at all the errors and then check for the tags before and after each incorrect assignment. We then look at the tags before and after it and check for its consistency.

**Finding consistent rules**

From the previous method we get a list of rules. But the rules in which the original context is same are ignored as those cases are ambiguous for our tagger. For example we cannot have rules of the type in which:

1. X Y PREV-TAG A NEXT-TAG B
2. X Z PREV-TAG A NEXT-TAG B

i.e. `A|X|B->A|Y|B and A|X|B->A|Z|B`

where a,b is the context and x is the original tag and y is the new tag. So we remove all such rules from the list of rules. Also rules which are statistically unimportant can be ignored.

## Results

We ran the tagger for the first **10%** of the provided Hindi corpus.

For each test we calculate the score by the following formula:

$$Score = 1 - (Total\ number\ of\ Incorrect\ Tags\ \div Total\ number\ of\ words)$$

**Outcome 1**

**When no rules were applied and the tagger was run to give the best-fit tag:**

On a given total of 26514 words, 1956 tags were assigned incorrectly. This gives us a total score of **92.622765** which is pretty high.

**Outcome 2**

**On applying the rule ANY V_VAUX PREV-TAG V_VM NEXT-TAG V_VAUX**

On a given total of 26514 words, 1910 tags were assigned incorrectly. This gives us a total score of **92.796258** which is little improvement.

**Outcome 3**

**On applying the rule N_NN N_NNP PREV-TAG N_NNP OR NEXT-TAG N_NNP (Simple try for NER)**

On a given total of 26514 words, 2233 tags were assigned incorrectly. This gives us a total score of **91.578034** which is our not so successful attempt on NER.

**Outcome 4**

**On applying the rule V_VM V_VAUX PREV-TAG V_VM OR V_VAUX**

On a given total of 26514 words, 1891 tags were assigned incorrectly. This gives us a total score of **92.8679188** which is little improvement.

**Outcome 5**

**On applying the rule V_VAUX V_VM NEXT-TAG CC_CCS AND PREV-TAG V_VM**

On a given total of 26514 words, 1985 tags were assigned incorrectly. This gives us a total score of **92.5133892** which is a slight decrease.

**Outcome 6**

**On applying the rule DM_DMD PR_PRP NEXT-TAG PSP PREV-TAG PSP**

On a given total of 26514 words, 1948 tags were assigned incorrectly. This gives us a total score of **92.6529381** which is a slight increase from the original.

**Outcome 7**

**On applying the rule V_VAUX V_VM NEXT-TAG RD_PUNC PREV-TAG N_NST**

On a given total of 26514 words, 1949 tags were assigned incorrectly. This gives us a total score of **92.6491665** which is a slight increase from the original.

**Outcome 8**

**(This rule cannot be represented by Brill's rules as it involves changes on )**

**DM_DMD|V_VAUX|CC_CCS -> PR_PRP|V_VM|CC_CCS**

On a given total of 26514 words, 1944 tags were assigned incorrectly. This gives us a total score of **92.39088** which is a slight increase from the original.

## Analysis

Surprisingly, the tagged data had about 92.6% accuracy without any rules being applied. On further applications of context-rules, the percentage of error was reduced. For each rule in isolation but rules in combination sometimes led to increase in percentage of error due to conflicting rules. The maximum achieved correctness was about 92.9697518% on applying the rules in combination:

1. **V_VM V_VAUX PREV-TAG V_VAUX OR V_VM**
2. **V_VAUX V_VM PREV-TAG N_NST NEXT-TAG RD_PUNC**
3. **DM_DMD|V_VAUX|CC_CCS -> PR_PRP|V_VM|CC_CCS**
4. **DM_DMD PR_PRP PREV-TAG PSP NEXT-TAG PSP**

## Conclusion

From the above experiments we can conclude that Brill's tagger can be used effectively not only for English but for Hindi too. But the improvement that can be achieved over the statistical normal is very low (less than 0.5%).

## References


- Eric Brill. (1992). A Simple Rule Based Part of Speech Tagger, In Proceeding of the Third Conference on Applied Computational Linguistics (ACL), Trento, Italy, 1992, pp.112–116.
- Naveen Garg, Vishal Goyal, Suman Preet(2012). Rule Based Hindi Part of Speech Tagger ,Proceedings of COLING 2012: Demonstration Papers, pages 163–174, COLING 2012, Mumbai, December 2012