Statistical Morphological Analyzer of Hindi

Arpita Das
MS CSE (201407536)

Harish Yenala
MS ECE (201407613)

***Abstract*— Statistical morph analyzers have proved to be highly accurate and easier as compared to rule based approaches. The morph analyzer (SMA++) is an improved version of statistical morph analyzer (SMA) described in Malladi and Mannem (2013). SMA++ predicts the gender, number, person, case (GNPC) and the lemma (L) of a given token. The SMA in Malladi and Mannem (2013) is modified by adding some rich machine learning features. The feature set was chosen specifically to suit the characteristics of Indian Languages.**

I. INTRODUCTION

Morphological analysis for Indian Languages is defined as the analysis of a word in terms of its lemma or root word(L), gender (G), number (N), person (P), case (C) and TAM label (tense, aspect and modality). The tool which predicts Morph Analysis of a word is called a Morph Analyzer (MA).Most of the techniques are rule-based which are difficult to maintain. Statistical Morph Analyzer (SMA) is an MA which uses machine learning to predict the morph information. Using the training data and the feature-set, statistical models are formed. These models help to predict the morph-analysis of the test data. This works for all words, including out of vocabulary words. SMA is language independent. SMA++ is a modification over SMA by adding some context-dependent features (which are important for Indian Languages).Indian languages are lexically and grammatically similar. Gramatically, there are many similarities.

Indian languages are

1. Synthetic
2. derivational
3. inflectional
4. Indian Languages predominantly have subject-object-verb (SOV) word order. They show agreement among words.

We captured all these characteristics, by building a robust feature set.

II. METHODOLOGY

A. Feature Set

The feature-set was chosen specifically to suit the Indian Languages. The following are the features used:

(i) **Suffixes** :
Indian languages show inflectional morphology. The inflectional morphemes carry the G, N, P and C of a word. These morphemes generally occur in the form of suffixes. Hence, to capture the inflectional behaviour of ILs we considered the suffixes as a feature for the ML task.

(ii) **Previous morph tags and next morph tags :**
Agreement is an important characteristic of ILs. Through agreement, GNPC of a token may percolate to the other tokens. An example to this is, if the subject (noun) is masculine, then the verb form should also be masculine. To capture agreement, we considered features which carried the GNPC of the neighbouring words. Previous morph tags feature captures predicted morph tag of previous 3 tokens. Next morph tags feature captures the set of morph tags of the next token, if found in the training corpus.

(iii) Word Forms :
ILs are morphologically rich languages. Words carry rich information regarding GNPC. To capture this characteristic we considered three features relating to word forms. Word present captures the word form of the present token. Word previous captures the word form of the previous token.Word next captures the word form of the next token.

(iv) Part of Speech (POS) :
POS is one of the of the fundamental ML feature of any NLP task. Based on the POS of the word, the set of possible inflections can be found. For example, verbs have a set of inflections and nouns have another set. To capture such information we included POS in the feature-set.

The **Support Vector Machine (SVM)** (using linear classifier) was used for the ML task.

B. Choosing Class Label

For the ML task, the class-labels for G, N, P, C were chosen from the training data itself. For lemma, the class-labels were formed based on the edit-distance operations required to convert the given token to its lemma.
The Algorithm is explained using an example. Consider the token crying. The lemma for crying is cry.

Step 1: The token and its lemma are reversed. crying becomes gniyrc and cry becomes yrc.
Step 2: Note the edit operations required to convert reversed token to the reversed lemma. To convert gniyrc to yrc we need to delete the characters at the 1st, 2nd and 3rd indices. Hence the edit operationswould be [d 1, d 2, d 3], where 'd' represents delete operation.
Step 3: The set of edit operations would form the class-label. [d 1, d 2, d 3] would be the class-label and would be added to the set of class-labels

III. EXPERIMENT

The training and test data are taken from the HTB (Hindi TreeBank)
For **Gender Classification** we have taken the following features:
0-Suffix,1-Previous Gender,2-Next Gender ,3-Previous wordform,4-next wordform, 5-post position,6-present word form,7-POS
For **Number Classification** we have taken the following features:
0-Suffix,1-Previous Number,2-Next Number ,3-Previous wordform,4-next wordform, 5-post position,6-present word form,7-POS
For **Person Classification** we have taken the following features:
0-Suffix,1-Previous Person,2-Next Person ,3-Previous wordform,4-next wordform, 5-post position,6-present word form,7-POS
For **Category Classification** we have taken the following features:
0-Suffix,1-Previous Category,2-Next Person Category,3-Previous wordform,4-next wordform, 5-post position,6-present word form,7-POS
For **G,N,P,C Classification using combination of all features** we have taken the following features:
0-suffix ,1-prev gender, 2-next gender, 3-prev person, 4 next person,5-prev number ,6-next number,7-prev category ,8-next category,9-prev root,10-next root,11-current root ,12-Postposition 13-POS
For **Lemma Classification using combination of all features** we have taken the following features:
0-suffix ,1-prev gender, 2-next gender, 3-prev person, 4 next person,5-prev number ,6-next number,7-prev category ,8-next category,9-prev root,10-next root,11-current root ,12-Postposition 13-POS

IV. RESULTS

The feature-set, which was specifically chosen for Indian Languages, should contribute to high accuracies. With even the small training data we had access to ,we got an accuracy of 50 to 52% when we tried to predict the G,N,P,C of a token with specific contextual features. When we used the combination of all features the accuracy improved to 55% for the same training and the test data. We also tried increasing the context information as features that is instead of taking one previous morph tag or word form we tried for bi-gram and tri-gram versions of them, the use of bi-gram increased the accuracy to 56.5% but no significant change was observed for tri-gram or higher n-grams.

V. CONCLUSION

i. SMA++ out performs other SMAs. For Hindi, the L+G+N+P+C accuracy was 56.5% using the small set of data.This show that SMA++ is a marked improvement over the SMA in Malladi and Mannem (2013) .

ii. SVM with linear kernel gave the best result for classification. We have tried using the sigmoid and RBF kernel but they showed no significant improvements. The Bernoulli Classifier gave the second best result.

iii. The use of bigram improved the results but there was no change in accuracy by using trigram or higher order ngrams.

iv. The use of combined features (that is combination of all contextual G,N,P,C information) led to high accuracy .

REFERENCES

[1] Statistical Morph Analyzer for Indian Languages. Saikrishna Srirampur, Ravi Chandibhamar, Radhika Mamidi Dublin, Ireland, August 23 2014.
[2] Grzegorz Chrupała. 2006. Simple data-driven context sensitive lemmatization. Procesamiento del Lenguaje Natural, 37:121–127.