POS Tagging of Konkani, A Resource Scarce Indian Language

Anonymous ACL submission

Abstract

Supervised Machine Learning Models have achieved state-of-the-art results in POS Tagging for resource rich languages. Resource rich languages have the luxury of availability of huge annotated corpora. However, this is not the case with many Indian languages. Konkani is one such resource poor language with 7.4 million [1] native speakers, where there is a scarcity of available annotated data. Lack of tagged data has led to the exploration of unconventional methods like Active Learning, Self Training with Hidden Markov Models, Clustering using word embeddings and Graph Clustering. Out of all these, Graph Clustering, Active Learning, and Self Training performed comparatively better than other methods. Active Learning is found to be the most efficient method which significantly reduces the dependence on manually annotated labeled data.

1 Introduction

POS tagging, also known as word-category identification, is the process of assigning a part of speech (POS) tag to a word, based on both its definition and its context. Common models used for POS tagging include Hidden Markov Models (HMM), Conditional Random Fields(CRF) and Long-Short Term Memory (LSTM) Networks. These are Supervised Machine Learning Models, and rely on an abundance of labeled data. Konkani is one of the many resource poor languages in India. Developing accurate POS tagged Konkani data will facilitate the advancement of Natural Language Processing (NLP) research in Konkani. On exploring Active Learning, Self Training, and Clustering, the methods that work well for POS Tagging of Resource Poor Languages can also be understood.

[1] https://en.wikipedia.org/wiki/Konkani_language

2 Related Work

Brants et al. Brants (2000) established that an HMM using trigrams performs at least as well as other known methods at POS tagging. In (Gadde and Yeleti, 2008), the TNT tagger, and Conditional Random Fields are used to generate POS tags for Hindi with an accuracy of 91.35% and Telegu with an accuracy of 91.23%. This paper discusses the effect of introducing features like Root of the word, and the Gender, Number and Person of the word to TNT.

K-Means Clustering and Graph Clustering are forms of unsupervised machine learning. In (Biemann, 2006), the graph G consists of a representation of all sentences of the dataset. The square of the adjacency matrix A of graph G is such that the element at position (i, j) is the number of nodes common to nodes i and j. All words of the same POS class having a large number of common nodes, while two words of different POS classes will have a very small number of common nodes. This Graph is then used to cluster English words into POS Tag Classes with 88.8% weighted average cluster purity.

When there is very little training data, and a large amount of testing data, semi-supervised learning models like Self Training an Active Learning can be used. In (Settles and Craven,

2008), different methods used to select data points for active learning intervention are discussed.

In (Rani et al., 2016) (to be henceforth referred to as the POSTagger), a data mining approach of POS Tagging is implemented, where a large amount of untagged data and a small amount of tagged data is available. The tagged data set is used to learn possible tags for a given context. These context based association rules are then grouped by their POS tags to form clusters of rules for each tag. Using this tagger for Hindi, English, Tamil, Telegu, and Bengali, accuracies between 79% and 85% were achieved.

In (Mishra et al.), POS tagging of resource poor Indian Languages is achieved by using a parallel alignment of Hindi data, and the data of the resource poor language. The features of the Hindi data are then projected onto the target language, and used for the POS tagging of the target language.

3 Approach

Due to the absence of any publicly available dataset in Konkani, we scraped data from online Konkani sites (which site?).

This data, however, was unlabeled. The only features available are the words of the sentences. For seed data, we used the annotated Konkani data that was a part of Indian Languages Corpora Initiative (Jha, 2010) [2]. The konkani data was obtained by translating Hindi sentences and manually POS tagging the words. However, the problem with using translated data is that it may not always be correct in the target language. The translations tend to focus more on word level alignments with the source language rather than being natural in the target language. Nevertheless, this data is used for the initial training of the POS tagger.

3.1 POS Tagging

POS Taggers require labeled data to train on. The bigger the annotated corpus size, the better the accuracy of the POS taggers. Tagged test-sets are used for predicting the POS tags and evaluate the performance of the taggers.

[2] sanskrit.jnu.ac.in/projects/ilci.jsp?proj=ilci

As a seed set, 500 sentences (6291 words) of Konkani data are used for training, and 500 sentences (7673 words) are used as a part the testing dataset. The scraped data is used as untagged data as it is not influenced by the source language. However, as the tagged data is influenced by the source language, its structure may differ from the untagged scraped data, and this may lead to erroneous tagging. To check the effect of using untagged, scraped data, the model is run using the remaining translated konkani data that is not used in the training and testing datasets as the untagged data. Initially, just the words are used in the sentence. To provide the model with more information, the words are then appended with their prefix and suffix, in the form prefix_word_suffix. To experiment further, only the suffixes of the words are used, to limit the number of instances of unseen words.

3.2 Self Training

Given data, a Hidden Markov Models (HMM) can efficiently generate POS tags for the test data. To handle the 0 probabilities of unknown words, an SVM is trained. Therefore for unknown words, the zero probabilities are replaced with the probabilities predicted by the SVM for the given word. The SVM model was therefore built solely on suffixes of length 5 of the Konkani words. To handle the 0 probabilities due to missing bigrams and trigrams, Add One Smoothing is attempted. This method of Smoothing was applied to the probability matrices, and the Viterbi algorithm was run to find the best sequence of tags.

500 sentences of the Machine Translated Konkani data are manually validated and used with the HMM to build the initial model. Thereafter, 100 sentences at a time are self trained on the model to see the effect of the additional data on the accuracy of the model.

3.3 Active Learning

Active Learning is a method by which anomalous/erroneous data points from the predicted data are identified, and only those points are manually checked for errors. Identification of these points is done by checking the difference between the highest predicted probability and the second highest predicted probability

ID	Sentence
1	बरें कितें,वायट कितें, तें ताकाह्या तेभायर तो...
2	मेंदवाचो अंत्यमस्तिष्क हो भाग मनशाचें गिन्यान ...
3	"'नागेश करमली"' चो जल्म ५ फेब्रुवारी १९३३ वर...

Table 1: Extracted Untagged Data.

of the data points. The higher the difference, the more confident the model of its prediction. Thus, the 500 validated sentences of the Translated Konkani Data are used for training. The 500 sentences of the test data set are split into 20 groups of 25 sentences each. Each group is tested on, and the predicted tags with the lowest confidence values are checked for errors. The modified data is then appended to the training set, and the model is retrained. The CRF++ Toolkit (Kudo, 2005) was used for this model.

3.4 Clustering

Due to the dearth of tagged data, Unsupervised and semi-supervised learning methods have to be relied upon. Clustering of the scraped data was attempted using the K Means algorithm. As the scraped data had no tags, words common to both the scraped data and the translated data were found, and the tags available were transferred from the translated data to the scraped data. Clustering was initially attempted using the suffixes (upto length 5) of the words, as words belonging to the same Part Of Speech class happened to have similar suffixes. It was then done using word vectors as features. Fast Text embeddings, and GloVe embeddings were fetched and used for clustering. After clustering, for all the words in a cluster which have tags present, the tag that is encountered the maximum number of times is assigned to the entire cluster.

Graph Clustering is done by building an adjacency matrix of all the sentences of the scraped untagged konkani dataset. The square of this matrix (A^2) is then found. Let *maxrow()* be a function that operates rowwise on the matrix, and sets the highest element to 1, and the rest to 0. Let I_n be the identity matrix of dimensions nxn. The following algorithm is used to cluster the graph.

$D^0 = I_n$

$i = 0, j = 1$

do

$D^i = maxrow(D^i)$

$D^j = D^i * A$

$i = i + 1, j = j + 1$

$while D^j != D^i$

This algorithm is run until the matrix D does not change with more iterations. For a node i, if the largest element in row i of matrix D is in column j, i has been clustered into class j. Using the words common to the scraped Konkani data and the translated Konkani data, some words of the dataset were tagged. Within each cluster, the tag that occured the most across the cluster was set as the POS tag of the cluster.

4 Results Analysis

4.1 POS Tagging

The POSTagger was used to attempt the POS tagging of Konkani. It tags words as Nonetags and notvaltags if it cannot find a valid tag for it. The accuracies obtained when using the entire word dataset, the prefix_word_suffix dataset, and the suffixes only dataset, listed in the table below are calculated by first considering the entire test dataset (Accuracy), and then just the words that were not tagged as Nonetags and notvaltags (Accuracy).

4.2 Self Training

The HMM for Konkani data using trigrams was initially trained on 500 validated sentences, and tested on 100. The predicted labels for the 100 sentences were then added to the training dataset, and the model was retrained. This was done with 4 sets of 100 sentences of testing data. As it can be seen from the table, initially, with the addition of self trained data, the accuracy took a drop. But, the accuracy started increasing at the addition of the last 100 sentences, and the overall decrease in accuracy was not significant.

Word Structure	Untagged Data Used	General Accuracy	Acc
Word	Scraped Konkani Dataset	9	38
Word	Translated Konkani Dataset	9	40
prefix_Word_suffix	Scraped Konkani Dataset	9	38
prefix_Word_suffix	Translated Konkani Dataset	9	40
suffix	Scraped Konkani Dataset	9	34
suffix	Translated Konkani Dataset	9	35

Table 2: Accuracies obtained with the POSTagger.

If, after testing on each set of 100 sentences, the predicted POS tags are manually checked and corrected before appending the predictions to the training data set, the accuracies remain almost similar. Thus, it is unnecessary to invest time in manually checking the predicted tags while self training the POS Tagging Model.

4.3 Active Learning

For Active Learning, the CRF++ Tool is used. It is trained on the 500 sentences. After that, 25 sentences at a time from the test set are predicted upon. The data points for which the difference between the probability of the first and the second highest predicted tags is lesser than 0.1 are checked, and the predictions are added to the training data set and the model is retrained. Over the 20 groups of 25 sentences each, the accuracy initially decreased, and then slowly picked up again. The accuracy of the total (predicted, and checked) tags of the words of the 500 sentences was 69.93%.

All the figures reported in Table 4.3 are in terms of sentences. Active Learning reduces the time spent in manual checking of the data, as only the samples with low confidence need to be checked.

4.4 Clustering

All the available Konkani data (that translated from Hindi, as well as the scraped data) is clustered into 31 groups, in an attempt to tag data using K-Means clustering. Using the suffixes of the words as the feature, the clusters are formed. However, the clusters aren't well formed, almost clusters are predominantly occupied by Nouns. Clustering was also attempted using fast text word embeddings. However, this also resulted in all clusters being dominated by nouns.

As this form of clustering proved to be unreliable, graph clustering was attempted. Graph Clustering implemented to cluster Scraped Konkani data. 11000 sentences were tagged using this method, and from the 40000 unique words present, 39985 were tagged with 34 unique POS tags. This method worked more efficiently than the others as it was able to find 28 POS tags, as opposed to tagging every cluster as a Noun due to the large number of nouns in the dataset. To test this method, it was tested on 500 sentences of the tagged translated konkani parallel data, and this resulted in an accuracy of 69.9%.

5 Conclusion

Self Training, Active Learning, a POSTagger, and Clustering were explored for Konkani POS Tagging. K Means Clustering did not work well with abundant untagged data as most of the words were tagged with the most frequently occurring POS Tag, Nouns. Comparatively, Graph Clustering partitioned the data into 34 POS Tag Clusters and works accurately even with scarcity of data. Self Training with HMM had a higher accuracy than Active Learning with CRF++, but, Active Learning in general ensures more accuracy, as the data points with low confidence are checked, with lesser human effort than boot-strapping with validation. The POSTagger was unable to perform well due to the dearth of good quality data. The analysis of the semi-supervised learning methods for tagging of konkani data gave valuable insights that K Means Clustering with limited data can be outperformed by Graph Clustering, Active learning, and Self Training. Graph Clustering obtained an average accuracy of 69.9%, however, it managed to partition the words into 34 POS Tag clusters, as opposed to the models whose results were

Training Dataset Size	Self Training	Active Learning
500	77.28	75.28
600	78.54	68.52
700	75.88	61.60
800	76.46	60.57
900	77.69	60.99

Table 3: Accuracies obtained with Self Training and Active Learning

dominated by nouns. Active Learning and Self Training have both, reduced the manual effort required to annotate data, as well as generated Konkani POS Tagged data with at least 77% accuracy.

References

Christian Biemann. 2006. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems.

Thorsten Brants. 2000. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics.

Phani Gadde and Meher Vijay Yeleti. 2008. Improving statistical pos tagging using linguistic feature for hindi and telugu. *Proc. of ICON*.

Girish Nath Jha. 2010. The tdil program and the indian langauge corpora intitiative (ilci). In *LREC*.

Taku Kudo. 2005. Crf++: Yet another crf toolkit. *http://crfpp. sourceforge. net/*.

Pruthwik Mishra, Vandan Mujadia, and Dipti Misra Sharma. Pos tagging for resource poor indian languages through feature projection.

Pratibha Rani, Vikram Pudi, and Dipti Misra Sharma. 2016. A semi-supervised associative classification method for pos tagging. *International Journal of Data Science and Analytics*, 1(2):123–136.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics.