Support Vector Machines based Part of Speech Tagging for Nepali Text

Tej Bahadur Shahi
Tribhuvan University
Central Department of Computer Science and Information Technology (CDCSIT)

Tank Nath Dhamala
Tribhuvan University
Central Department of Computer Science and Information Technology (CDCSIT)

Bikash Balami
Tribhuvan University
Central Department of Computer Science and Information Technology (CDCSIT)

ABSTRACT

Optimal part-of-speech tagging have great importance in various field of natural language processing such as machine translation, information extraction, word sense disambiguation, speech recognition and others. Due to the special nature of the Nepali language, Tagset used and Size of the corpus (training data), getting accurate part-of-speech tagger is one of the challenging task. This study is oriented to build an analytical machine learning model based on which it can be possible to determine the attainable accuracy. To complete this task, the support vector machine based part-of-speech tagger has been developed and tested for various instances of input to verify the accuracy level. The SVM tagger construct the feature vectors for each word in input and classify the word into one of two classes (One Vs Rest).

General Terms

Machine Learning, Natural Language Processing, Classification, Part of Speech (POS) Tagging.

Keywords

Support Vector Machine, POS Tagging, HMM, Supervised Machine Learning

1. INTRODUCTION

In general, Tagging is the process of assigning any label to a linguistic unit or token. The linguistic unit may be word, phrase, sentence etc. In this work the tagging refers to the process of assigning part of speech (POS) tag to a word. The computer programs designed to automatically assign the POS tag to a word in natural language text, are called taggers. The outline of process is shown as in figure 1.1.

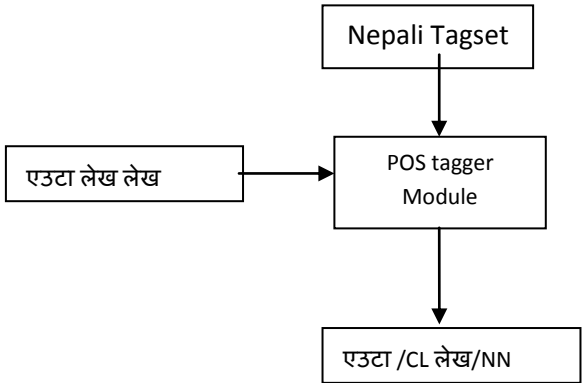

Figure1.1: POS tagging Example

Nepali is morphologically rich language [1] and one has to consider many features to build a language model for such language. The POS tagging approaches like rule based and hidden Markov Model (HMM) cannot handle many features. The support vector machine based POS tagger has been implemented in [2] for a Bengali language which is also morphologically rich and shown the outstanding performance. In [2] rich feature set has been used to model the language characteristic. In [3] SVM based tagger was proposed which is efficient, portable, scalable and trainable. Support vector machine (SVM) are recently developed supervised learning method having good performance and generalization [4]. SVM has been successfully applied in text classification and shown that SVM can handle large features and is resist of over fitting [5].

1.1 Support Vector Machine

In their basic form, SVM construct the hyperplane in input space that correctly separate the example data into two classes. Hence SVM is a binary classifier. This hyperplane can be used to make the prediction of class for unseen data. The hyperplane always exist for the linearly separable data [4].

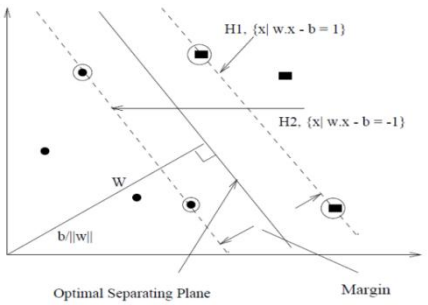

Figure1.2: Support Vector Machine hyperplane

2. RELATED WORK

The Nepali National Corpus (NNC) from NELRALEC (Nepali Language Resources and Localization for Education and Communication) project, which contain 14 million Nepali words. It consists of speech corpus, spoken corpus, core sample (CS), general collection, and parallel data. Some part of it was first manually tagged (One hundred and sixty texts from the NNC–CS were annotated manually using this tagset with 112 tags). This data then served as the basis for the training of an automatic tagger. The Nepali English parallel

corpus annotated with 43 POS tag developed at Madan Puraskar Pustakalaya (MPP) contains nearly 88000 words [6].

NELRALAC tagset is the first work in developing Nepali tagset which consist of 112 tags. This tagset has been compiled with reference to widely published grammars of Nepali. This tagset was used to tag (Nepali National Corpus) NNC manually and semi manually. As [1] showed that error rates of annotation could be much higher with a large tagset, the reason primarily being the chances of assigning incorrect tags to the words out of confusion while manually annotating the training data itself.

The *Unitag* has been developed or customized for Nepali language and was used for semi automatic tagging of Nepali National Corpus under the NERLAC project. The tagset used is NERLAC tag set with 112 tags. *Unitag* was originally developed for Urdu language by hardie et [8]. It consists of a powerful morphological and lexical analysis system, and twin disambiguation modules, one based on hand-written rules and the other using a probabilistic system based on a Markov model. After tagging, the corpus was manually reviewed and the correction was done. Since the tagset used was which large, it introduced the more error in tagging.

In [1] , the TnT has been used as POS tagger with the 43 tags and training corpus of medium size as one of the pipelined module for computational grammar analyzer. First order Markov model has been implemented in [8] which use the same POS tagset as in [1] and reports the good accuracy (91%) for known word.

3. RESEARCH METHODOLGY

In this work, we proposed the support vector based model for POS tagging and uses the feature set as described in section 3.4 to create feature vector and then uses the SVMlight [5] tool to learn and classify the POS for a particular word.

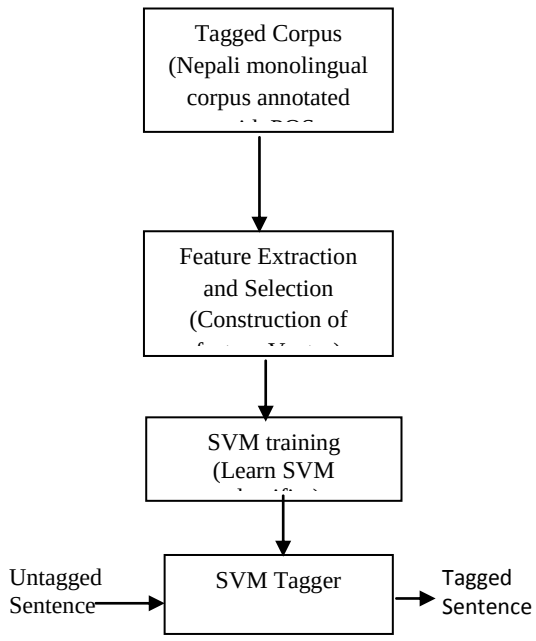

Figure 3.1: System flowchart

The SVMLight software package consists of two main components, namely the model learner (SVMlearn), the tagger (SVMclassify).

SVMlight is a binary classifier and it only work for two class classification but the POS tagging is a multi classification problem. Here each POS tag represent a class and in Nepali language there are 43 tag used for this purpose. To use the SVMlight tool for POS tagging purpose, the One Vs rest method of binarization of problem is used as described in section 3.3.

3.2 Training data format

Training data must be in column format, i.e. a word per line corpus in a sentence by sentence fashion. The column separator is the only one blank space. The word is expected to be the first column of the line. The tag to predict takes the second column in the output. Following is a sample of the training data:

सुश्री/NN
हाग/NNP
एलियान्टी/NNP
को/PKO
भूमिका/NN
खेल्नुहुन्छ/VBF

3.3One Vs Rest classification

This strategy is based on idea of building one classifier per class. To train N different binary classifiers, each one trained to distinguish the examples in a single class from the examples in all remaining classes. When it is desired to classify a new example, the N classifiers are run, and the classifier which outputs the largest (most positive) value is chosen.

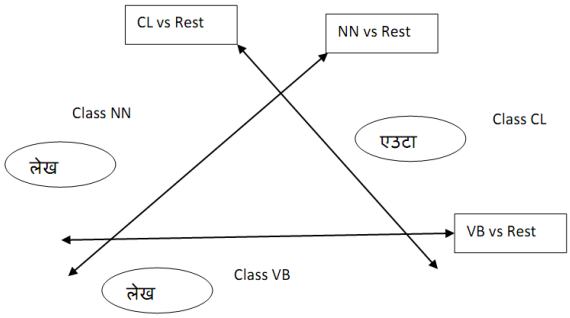

Figure 3.2: One Vs Rest classification illustration

3.4 Feature Set

The features used in the experiment are tabulated in the table 4.1.

Word Feature	w_{-3}, w_{-2}, w_{-1}, w_0, w_1, w_2, w_3
POS Feature	p_{-3}, p_{-2}, p_{-1}
Word Bigrams	(w_{-3},w_{-2}),(w_{-2},w_{-1}),$(w_{-}$

	$_1$,w_0),(w_0,w_1),(w_1,w_2)
POS Bigrams	(p_{-3}, p_{-2}) (p_{-2}, p_{-1})
Word Trigram	(w_{-3},w_{-2},w_{-1}), (w_{-2},w_{-1},w_0), (w_{-1},w_0,w_1), (w_0,w_1,w_2), (w_1,w_2,w_3)
Ambiguity Classes	a_0, a_1, a_2,a_3
Maybe's	m_0, m_1, m_2, m_3

Table 3.1: Feature set used

3.5 Feature Vector Construction

The process of feature vector construction for a word "**भने"(vane)** is descried in this section as an example.

प्यालेस्टाइनी<JJ> ओलम्पिक<NN> कमिटि<NN>का<PKO> एक<CD> अधिकारी<NN>ले<PLE> **भने<VBNE>** कमिटि<NN>ले<PLE> सब<JJ>भन्दा<VBO> पहिले<PLE> १९७९<CD> मा<POP> सदस्यता<NN>का<PKO> लागि<POP> निवेदन<NN> दिएको<VBKO> थियो<VBX>

The dictionary entry for target word "**भने"** is:

भने 126 4 VBKO 1 VBNE 1 VBO 1 VBF 4

Some of the features along with their ids for the target word **भने /VBNE** are:

w(-3)_is_एक 71

w(-2)_is_अधिकारी 72

w(-1)_is_ ले 73

w(0)_is_ **भने 37**

w(1)_is_कमिटि 74

w(2)_is_ले 75

w(3)_is_सब 76

p(-3)_is_CD 77

p(-2)_is_ NN 78

p(-1)_is_PLE 79

a(0)_is_VBKO-VBNE-VBO-VBF 42

m(0)_may_be VBNE 43

m(0)_may_be_VBO 7

m(0)_may_be_VBF 45

a(1)_is_NN-VBO 80

m(1)_may_be_NN 81

m(1)_may_be_VBO 82

……

……..

The feature vector for target word **भने is**

+1 71:1 72:1: 73:1 37:1 74:1 75:1 76:1 77:1 78:1 79:1 42:1 43:1 7:1 43:1 7:1 45:1 80:1 81:1 82:1……

4. RESULTS AND DISCUSSION

Test data is prepared form the original corpus. It consists of 10775 tokens from the original corpus. This part of data is not used during training period. Since the count of tokens should be in whole number, some consideration has made about the percentage of testing data to make it whole number. The test data contains total of 10775 randomly selected tokens out of which 82% are unambiguous, 13% are unknown tokens and 5% are of ambiguous. This is shown in the following pie chart.

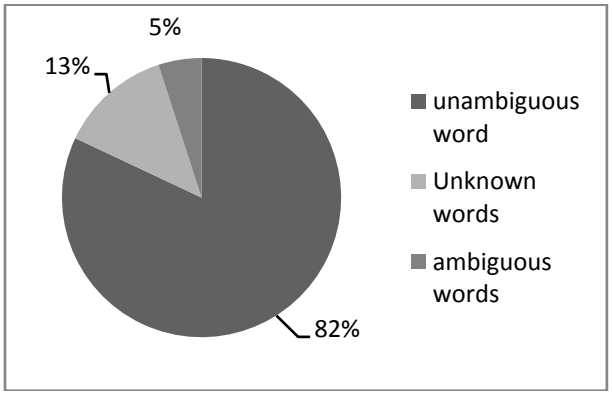

Figure 4.1: Statistics of Sample Test

Accuracy measurement of tagger

The accuracy is measured with matching two file: one is test file and another is manually tagged test gold file. The correctly tagged tokens are those which match in both files and the remaining tokens are tagged incorrectly. For this a matcher program is written which sequentially reads the both file and match the line by line and increment the count if both line matched in both files.

On the test file of 11147 *words, the overall accuracy is calculated as*

$$accuracy = \frac{Total\ word\ correctly\ tagged}{total\ word\ in\ test\ file} = \frac{10050}{10775}$$
$$= 93.27\%$$

The detail comparison of tagger performance with ambiguous and unambiguous word is tabulated below (In table 4.1).

Table 4.1: Comparison of performance of tagger [9]

	No of tokens	Accuracy	Error
Ambiguous word	538	490/538 (91.07%)	48/538(5.88)
Unambiguous words	8819	8290/8819(94.01%)	265/8819(1.07 %)
Unknown words	1418	1270/1418(89.56)	141/558(9.94%)

Validation and Evaluation

Cross validation technique is used to validate the measured accuracy of tagger. The general k- cross validation is a technique which divides the whole corpus into ten parts and nine part(90%) of data is used for training and remaining one part is used for testing. The process is repeated for ten times taking each of ten parts as testing instances.
Here the 10-cross validation is adapted in which the whole corpus is divided into 10 portions sequentially and in each iteration of program, the 9 folds are used for training and the remaining 1 fold is used for testing.

And in other respect, the learning nature of tagger is evaluated with the different size of training data. The size of training data is gradually increased and the performance of tagger is observed. The result so found is presented in the table 5.3.

Table 4.2: The tagging accuracy for different training

Training data size	Accuracy(in percentage)	
	SVM	TnT
10000	71%	61%
20000	79%	69%
40000	85%	72%
80000	90%	90%
100000	92%	91%

The corresponding learning curve is shown in figure

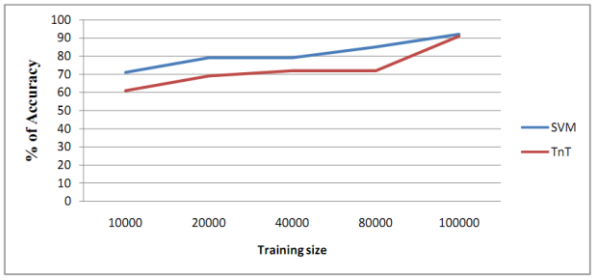

Figure 4.2: Learning Curve for Nepali Tagging

The learning curve shows the gradual increment in accuracy for the large size of training data for the HMM tagger and it performs not very well for the small set of training data since the bigram probabilities are not found sufficiently in the case of small size data and most of the case it becomes zero. But the SVM tagger does not depends upon the probabilities rather it depends upon the features extracted from the training and testing data. So it performs very well for the small training data size as well as for large data size.

4.1 Results

The overall accuracy is slightly better for SVM then other two taggers as shown table 5.4. This is because of rich pattern set provided for SVM. TnT perform well for known word but it gives lower accuracy for unknown word, it is because there is no mechanism to deal with unknown word for Nepali language.

Table 4.3: Accuracy of different taggers for Nepali text

Tagger	Accuracy		
	Known Words	Unknown Words	Overall
TnT	92%	56%	74%
SVM	96.48%	90.06%	93.27%

5. FUTURE WORK

In this research work, the SVM based POS tagger is built which uses the dictionary as a primary resources. This dictionary is collected from the FinalNepaliCorpus which contains only 11147 unique words. The performance of tagger is dependent on this dictionary and so in future; such dictionary may be built using the information on news sources and available Nepali raw text with the help of morphological analyzer and part of speech acquisitions techniques.

The limitation of the SVM tagger built is the speed. It is found to be slow in training than other tagger. Since the SVM based POS tagger uses the different set of features to construct the feature vectors, the empirical analysis to find the optimal set of features may be the future work which may concentrate on speed optimization of tagger.

6. REFERENCES

[1] B. Prasain, LP. Khatiwada, B.K. Bal, and P. Shrestha. Part-of-speech Tagset for Nepali, Madan Puraskar Pustakalaya, 2008.

[2] A. Ekbal and S. Bandopadhya, Part of Speech Tagging in Bengali Using Support Vector Machine, In: Proceeding of IEEE 2008.

[3] Jesus Giménez and Lluís Márquez . SVMTool: A General POS Tagger Generator Based on Support Vector Machines, In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004). Lisbon, Portugal, 2004.

[4] Valdimir Vapnik, Corinna Cortes, Support Vector Networks, machine learning 20, 273-297, Kunwer Acedemic Publisher 1995.

[5] T. Joachims, Making Large-Scale SVM Learning Practical. Advances in Kernel Methods Support Vector Learning, B., Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[6] B.K. Bal, and P. Shrestha, Reports on Computational Grammar Madan Puraskar Pustakalaya, Patan Dhoka, Lalitpur, Kathmandu.

[7] A. Hardie, The Computational Analysis of Morphosyntactic Categories in Urdu, (PhD Thesis, Department of Linguistics and Modern English Language, Lancaster University, 2003)

[8] M. R. Jaishi., Hidden Markov Model Based Probabilistic Part Of Speech Tagging For Nepali Text, (Masters Dissertation, Central Department of Computer Science and IT ,Tribhuvan University 2009, Nepal).

[9] T.B.Shahi, Support Vector Machine Based POS Tagging For Nepali Text, (Masters Dissertation, Central Department of Computer Science and IT 2012 ,Tribhuvan University, Nepal).