Prediction of Part of Speech Tags for Punjabi using Support Vector Machines

Dinesh Kumar[1] and Gurpreet Josan[2]
[1]Department of Information Technology, DAV Institute of Engineering and Technology, India
[2]Department of Computer Science, Punjabi University, India

Abstract: *Part-Of-Speech (POS) tagging is a task of assigning the appropriate POS or lexical category to each word in a natural language sentence. In this paper, we have worked on automated annotation of POS tags for Punjabi. We have collected a corpus of around 27,000 words, which included the text from various stories, essays, day-to-day conversations, poems etc., and divided these words into different size files for training and testing purposes. In our approach, we have used Support Vector Machine (SVM) for tagging Punjabi sentences. To the best of our knowledge, SVMs have never been used for tagging Punjabi text. The result shows that SVM based tagger has outperformed the existing taggers. In the existing POS taggers of Punjabi, the accuracy of POS tagging for unknown words is less than that for known words. But in our proposed tagger, high accuracy has been achieved for unknown and ambiguous words. The average accuracy of our tagger is 89.86%, which is better than the existing approaches.*

Keywords: *POS tagging, SVM, feature set, vectorization, machine learning, tagger, punjabi, indian languages.*

Received September 18, 2013; accepted February 28, 2014; Published online December 23, 2015

1. Introduction

Part-Of-Speech (POS) tagging is a task of assigning the appropriate POS or lexical category to each word in a natural language sentence. It is an initial step in Natural Language Processing (NLP) and is useful for most NLP applications and has a diverse application domain including speech recognition, speech synthesis, grammar checker, phrase chunker, machine translation etc. POS tagging can be done using linguistic rules, stochastic models or a combination of both. In the rule based approach, a knowledge base of rule is developed by linguistic to define precisely how and where to assign the various word class tags. This approach has already been used to develop the POS tagger for Punjabi language with nearly an accuracy of 80%. Stochastic taggers are based on techniques like Hidden Markov Model (HMM) [3], Conditional Random Field (CRF) [9], decision trees [13], Maximum Entropy (ME) [12], Support Vector Machines (SVM) [6] and multi-agent system [15]. Out of all these statistical learning algorithms, we have used SVMs for following reasons.

- SVMs have high generalization performance independent of dimension of feature vectors. Other algorithms require careful feature selection, which is usually optimized heuristically, to avoid over fitting.
- SVMs can carry out their learning with all combinations of given features without increasing computational complexity by introducing the kernel function.

Conventional algorithms cannot handle these combinations efficiently. Development of a stochastic tagger requires large amount of annotated corpus. Stochastic taggers with more than 95% word-level accuracy have been developed for English, German and other European languages, for which large labelled data is available. The problem is difficult for Indian Languages (ILs) due to the lack of such annotated large corpus.

2. Related Works

2.1. For Punjabi

Very little work has been carried out in POS tagging for Punjabi. To the best of our knowledge only 02 POS taggers have been proposed so far. A rule-based POS tagging approach was applied for tagging Punjabi text, which was further used in grammar checking system for Punjabi [5]. Their approach was based entirely on the grammatical categories taking part in various kinds of agreement in Punjabi sentences and applied successfully for the grammar checking of Punjabi. This tagger uses handwritten linguistic rules to disambiguate the part-of speech information, which is possible for a given word, based on the context information. Later, HMM has been used for POS tagging to improve the accuracy of this tagger [14]. This POS tagger can be used for rapid development of annotated corpora for Punjabi. There are around 630 tags in this fine-grained tagset. This tagset includes all the tags for the various word classes, word specific tags and tags for punctuations. A neural network has also been used for the prediction of POS tags of Punjabi [7]. In this work authors have used trigram language model for POS tagging. An accuracy of 88.95% has been reported.

2.2. Rest of ILs

SVMs have been successfully applied to various ILs like Kannada, Bengali and Malayalam. For POS tagging of Bengali, SVM has been used SVM [4]. The Bengali POS tagger has been developed using a tagset of 26 POS tags. The system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the various POS classes. The POS tagger has been trained and tested with the 72, 341 and 20K word forms, respectively. Experimental results show the effectiveness of the proposed SVM based POS tagger with an accuracy of 86.84%. A SVM has been used for POS tagging of Malayalam language [2]. A corpus size of 1, 80, 000 words was used for training and testing the accuracy of the tagger generators. An overall accuracy of 94% has been achieved. It was found that the result obtained was more efficient and accurate compared with earlier methods for Malayalam POS tagging. A kernel based POS tagger for Kannada language has been proposed to analyze and annotate Kannada texts [1]. A corpus size of 54,000 words was used for training and testing the accuracy of the tagger.

3. System Design

Figure 1 shows the various components of the proposed system. The use and working of various components is explained in this section.

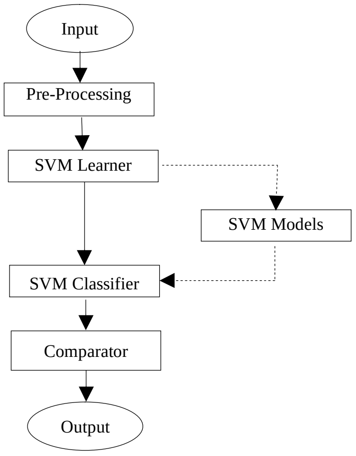

Figure 1. System design with various phases.

- *Input Unit*: The input comprises the manually annotated corpora on the basis of tagset comprising of 38 tags.
- *Pre-Processing*: The annotated corpora given to the Pre-processing unit, where tagged dictionary for each word is extracted and corresponding input is translated to vector form and a training file for each tag is generated.
- *SVM Learner*: The training files generated in Pre-Processing phase are input to SVM-Learner, where a model file for each tag is generated. These model files comprise the support vectors that are required to identify the tag of the text.
- *SVM Classifier*: Finally, the text to be tagged is given as an input to SVM-Classifier in the form of vectors, along with the model files generated in previous phase. The input vectors are compared with each model files and the output is generated, one for each model file.
- *Comparator*: The outputs generated from previous step are compared by the comparator and the output with the highest value is predicted as the tag of the input text.

3.1. POS Tag set

Punjabi words may be inflected or uninflected. Inflection is usually a suffix, which expresses grammatical relationships such as number, person, tense etc., for the proposed tagger, we have used a Punjabi tagset proposed by [8]. The tagset consists of 38 Coarse-grained tags. Table 1 shows the Punjabi POS tagset used for the proposed tagger.

Table 1. PoS tagset developed for Punjabi.

Main Category	**Sub Category**	**POS Tag**
Noun	Common	NN
Noun	Proper	NNP
Noun	Compound	NNC
Noun	Compound Proper	NNPC
Pronoun	All Categories	PRP
Adjective	All Categories	JJ
Cardinal	-	QC
Ordinal	-	QO
Verb	Main	VBM
Verb	First Person	FP
Verb	Second Person	SP
Verb	Third Person	TP
Verb	Present Tense	PT
Verb	Past Tense	PAT
Verb	Future Tense	FT
Verb	Auxiliary	VAUX
Adverb	-	RB
Postposition	-	PSP
Conjunct	Sub-ordinate	CS
Conjunct	Co-ordinate	CC
Interjection	-	INJ
Particle	-	PT
Quantifier	-	QF
Special Symbol	@, #, $, etc.	SYM
Reduplication	-	RDP
Meaningless Words	-	MW
Unknown Words	-	UNW
Question Words	-	QW
Verb Part	-	VP
Sentence Final Punctuation	|,?, !	SFP
Comma	,	COM
Colon, Semicolon	:,;	CSP
Left Brackets	(,[,{	OP
Right Brackets	),],}	CP
Dot	.	DP
Hyphen	-	HP
Single Quote	‘	SQP
Double Quote	“	DQP

In the tag set, person and tense POS sub category tags of Verb POS main category are used in conjunction with verb tag VBM. These tags cannot be used in isolation. e.g., in a sentence, a word which is behaving as main verb with second person and in future tense will be tagged with VBM_SP_FT tag.

3.2. Predicting Tags For Unknown Word

Unknown word class tag has been predicted by Rule-based method [10] and the decision tree-based method [11]. In this paper, we propose a method to predict POS tags of unknown Punjabi words using SVMs. In order to predict the POS tag of an unknown word, the following features are used:

- POS Tag Context: The POS tags of the two words on both sides of the unknown word.
- Word Context: The two words on both sides of the unknown word.

The following example shows how the prediction is done for unknown words. Suppose the training sentence is:

maa <NN> dain <PSP> kadman <NN> vich <PSP> jannat <NNP> hai <VAUX> | <SFP>

The sentence given to SVM for tagging is:

*maa dain **paira** vich jannat hai |*

The words and symbols “*maa*”, “*dain*”,” *jannat*”, “*hai*”, “|” are known words as they are seen in the training data but the word “*paira*” is unknown for the tagger. The features of word w_0 *(paira)* are shown in Table 2. These features are converted to feature vectors and given as an input to SVM Classifier, where it compares the feature vectors with all the feature vectors in all the models. The model that returns the highest value is treated as predicted tag.

Table 2. Neighbouring context for unknown word.

POS Tag Context	t-2=NN	t-1=PSP	t+1=PS	t+2=NNP
Word Context	w-2=maa	w-1=dain	w+1=vich	w+2=jannat

4. POS Tagging Algorithms

In this section, we have discussed the POS tagging algorithms for tagging Punjabi Sentences using SVM. The task of tagging has been divided into vectorization, training and classification. In the vectorization phase, the manually tagged Punjabi file is converted into SVM format. During training, the SVM is trained using formatted input file created in vectorization phase. The output of this phase is the model files for each POS tag. The last phase is the classification phase in which untagged file along with the model files created during the training phase is given as input and the tagged file will be generated as output. Algorithms 1 and 2, explains the procedures of training and classification as implemented in the proposed system. Table 3 shows the type and meaning of different variables used in both the algorithms.

Algorithm 1: Training algorithm.

```
Input: Tagged training file
Output: SVM model files

Begin
    Read training file;
    wc ← No. of words in training file;
    tag[ ] ← Extract POS tags from training file;
    w[ ] ← Extracts words from training file;
    for each tag in tag[ ] do
        Create example file corresponding to each tag;
    end
    for i ←1 to wc do
        Create a feature vector $fv_i$ for each $w_i$ in w[ ]
        Write: +$fv_i$ for $tag_i$ in corresponding tag file;
        Write: -$fv_i$ in remaining (tag[ ] - $tag_i$) tag files
    end
    for each tag in tag [ ] do
        Apply svm-learn on corresponding example files of
        tag to generate SVM Model Files;
    end
end
return Trained SVM Model for each POS tag
```

Algorithm 2: Classification algorithm.

```
Input: Test file
Input: SVM models
Input: DICT file
Output: Tagged files

begin
    Read tEST fILE and SVM mODEL fILES;
    v ← 0;
    wct ← No. of words in test file;
    for i ← 1 to wct do
        Create a feature vector fvi for each wi of wct;
        if $w_i$ is found in DICT file then
            ptag[ ]← ptag[ ] of $w_i$ from DICT file;
        else
            ptag[ ] ←All POS Tags
        end
        if count( ptag[ ]= 1) then
            predictedtag ← ptag[0];
        else
            for each tag in ptag[ ] do
                result ← Apply SVM classifier with ($fv_i$,tag, SVM Model);
                if (result > v) then
                    v = result;
                    predictedtag = tag;
                end
            end
        end
    end
    wi=wi <predicted tag> in tagged file
end
return tagged file
```

Table 3. Variables used in the algorithms and their meaning.

Variable Name	Type: Meaning
wc	Variable: Holds the no. of words in training file
tag[ ]	Array: Holds the POS tags
fv	Variable: Holds feature value
v	Variable: Holds temporary values
wct	Variable: Holds the no. of words in test file
ptag[ ]	Array: Holds predicted tags
w[ ]	Array: Holds words of training files
wt[ ]	Array: Holds words of testing file

5. Experimental Results and Discussions

Experimentation on the proposed SVM based Punjabi tagger is performed using manually tagged Punjabi corpus with 38 tags proposed by [8]. Different sizes of randomly selected training data sets were constructed. During the experimentation different data is obtained during training and testing. In this section, we have discussed the data obtained for different file sizes on the basis of various parameters like training and testing time, accuracy, precision, recall, F-measure. Tag-wise analysis is also discussed in this section.

Figures 2 and 3 shows that as we increase the corpus size (No. of words) during training and testing, the processing time is also increased. During training, SVM generates different models for the tags based on the training data and as we train SVM with big corpus size the processing of the data increases which results in increased training time.

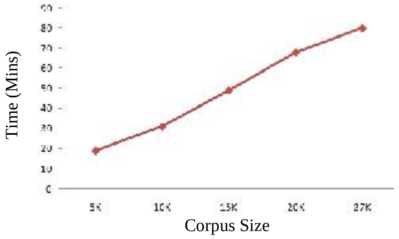

Figure 2. Training time with respect to corpus size.

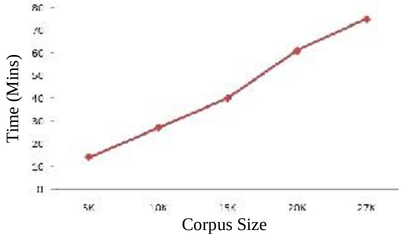

Figure 3. Testing time with respect to corpus size.

During testing, the tags are predicted using model files generated during training phase. So, as the size of model files increases it takes more processing time for the prediction of a tag for a word.

In our case, for a corpus of 5K words it took approximately 19 minutes and it goes up to 80 minutes for a corpus of 27K words on Intel core i3 3.3GHz Processor with 2GB RAM for training and testing. So, a higher configuration machine can be used to reduce training and testing time.

SVM based tagger shows four types of learning: Perfect-learning, near-to-perfect learning, partial-learning and no-learning.

The results shown in Table 4 depict this behaviour. SVM based tagger has shown perfect learning in case of conjuncts (sub-ordinate and co-ordinate), ordinals. Near-to-perfect learning has been obtained on pronoun, postposition and verb auxiliary. The tagger has shown partial-learning on verb main, adverbs, noun, pronouns etc. The tagger fails to learn tag mappings in the case of verb sub-categories like person and tense, interjection etc.

Table 4. Tag-wise accuracy achieved.

POS Tag	Accuracy
CS	100%
QP	100%
CC	99.25%
VAUX	99.18%
PT	89.44%
NN	87.52%
VBM	86.12%
RB	83.03%
JJ	71.70%
PRP	70.83%
INJ	67.86%

Precision, Recall, F-measure and accuracy are the measures to check the behaviour of the tagger. These measures are defined as follows:

$$Precision(P)= TP/(TP + FP) \quad (1)$$

$$Recall(R)= TP/ (TP+FN) \quad (2)$$

$$F\text{-}measure= 2^* (P^*R)/ (P+R) \quad (3)$$

Where True Positive count (*TP*): Number of words tagged as tag$_i$ both in the test data and by the tagger, False Positive count (*FP*): Words tagged as non-tag$_i$ in the test set and as tag$_i$ by the tagger, False Negative (*FN*): Words tagged as tag$_i$ in the test set and as non-tag$_i$ by the tagger and F-measure is a score that combines the two parameters. The values of these measures lie between 0 and 1. As shown in Figure 4, we converted the values obtained using Equations 1 to 3 for this measure to percentage.

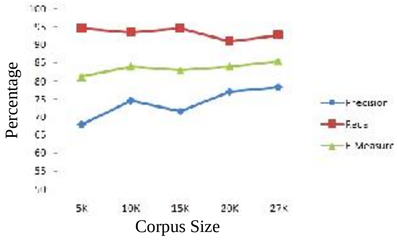

Figure 4. Precision, recall and F-measure at different corpus sizes.

Accuracy is the average number of words correctly tagged in the test data. The accuracy of the tagger is calculated with the help Equation 4:

$$A= (N/ T)^*100 \quad (4)$$

Where *A*: Is the Accuracy, *N*: Is the Number of words tagged correctly, and *T*: Is the Total number of words tagged.

From Figure 5 it is clear that as we increase the corpus size, the accuracy improves.

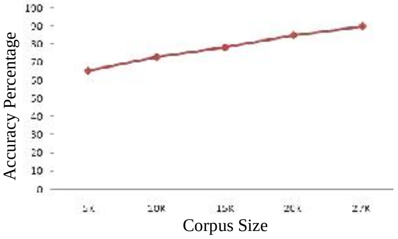

Figure 5. Accuracy achieved with different corpus size.

This is because in case of small training corpus size all the examples related to each and every tag of the tagset are not covered and accuracy in those cases affects the accuracy of the tagger. With the increase in number of examples and the training corpus size the tagger able to predict correct tags and the overall accuracy of the tagger improves.

Cross Validation (CV) is a performance measure that validates the prediction model on the basis of independent data set and also gives an estimate of the accuracy of the prediction model. There are different types of cross-validation techniques viz. k-fold, 2-fold, Leave-one-out CV etc. In this work we have taken the value of k as 5 i.e., we divided the training set into 5 smaller sets of equal sizes except the last. In this approach, a single subset acts as a validation data for testing the model and the remaining (k-1) subsets are used for training data. The process is repeated k times with different subset as validation data. The CV score of the prediction model is the average of the scores computed during k-iterations. The mean score and the standard deviation of the proposed model is 0.87 and 2.5 respectively.

6. Comparison with Existing Taggers

The proposed SVM based tagger has been compared with the existing taggers for Punjabi proposed by [6, 14]. Accuracy of the tagger is the most important parameter to judge the quality of the tagger so we compared the different taggers on the basis of accuracy only. The results shown in Table 5 clearly show that proposed tagger performed better than the already existing taggers for Punjabi.

Table 5. Comparison with exiting Punjabi tagger.

Total words	Technique	Accuracy
26,479	Rule Based	80.61%
26,479	HMM	84.37%
27,000	SVM (Proposed)	89.86%

7. Conclusions

In this paper, we showed that how SVM can be successfully applied to POS tagging of Punjabi Sentences. SVM achieves high accuracy as compared to rule based techniques and HMM techniques. SVMs have the advantage of considering the combinations of features automatically by introducing a kernel function. Feature set used here consisted of four neighbouring words and their tags. Feature set can be extended, to include substrings, identification of a number, delimiter, start of a sentence and end of a sentence can also be used.

Our method does not consider the overall likelihood of a whole sentence and uses only local information compared to probabilistic models. The accuracy may be improved by using some beam search scheme. Initial training of SVMs is slow. It took almost 1.5 hours to train SVM for a corpus of 27,000 words. We have used linear kernel for SVM, other kernels like Sigmoid, Polynomial with different degree can be used for SVM.

Our method outputs only the best answer and does not output the second or third best answer. Further, predictions of unknown words can be incorporated again into training leading to self-learning and enhanced POS tagger. A morphological analyzer can be used before inputting the words for tagging to the tagger and to further improve the accuracy of the system.

References

[1] Antony P. and Soman K., “Kernel based Part of Speech Tagger for Kannada,” *in Proceedings of International Conference on Machine Learning and Cybernetics*, Qingdao, pp. 2139-2144, 2010.

[2] Antony P., Mohan S., and Soman K., “SVM based Part of Speech Tagger for Malayalam,” *in Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing*, Kerala, India, pp. 339-341, 2010.

[3] Charniak E., Hendrickson C., Jacobson N., and Perkowitz M., “Equations for Part-of-Speech Tagging,” available at: http://cs.brown.edu/research/pubs/pdfs/1993/Charniak-1993-EPT.pdf, last visited 1993.

[4] Ekbal A. and Bandyopadhyay S., “Part of Speech Tagging in Bengali using Support Vector Machine,” *in Proceedings of International Conference on Information Technology*, Bhubneswar, India, pp. 106-111, 2008.

[5] Gill M., Lehal G., and Joshi S., “Part of Speech Tagging for Grammar Checking of Punjabi,” *the Linguistic Journal*, vol. 4, no. 1, pp. 6-21, 2009.

[6] Gimenez J. and Marquez L., “Fast and Accurate Part-of-Speech Tagging: The SVM Approach Revisited,” available at: http://nlp.lsi.upc.edu/papers/gimenez03.pdf, last visited 2004.

[7] Kashyap D. and Josan G., ”A Trigram Language Model to Predict Part of Speech Tags Using Neural Network,” *in Proceedings of the 14th International Conference, IDEAL*, Hefei, China, pp. 513-520, 2013

[8] Kumar D. and Josan G., “Developing a Tagset for Machine Learning based Pos Tagging in Punjabi,” *International Journal of Applied Research on Information Technology and Computing*, vol. 3, no. 2, pp. 132-143, 2012.

[9] Laferty J., McCallum A., and Pereira F., “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data,” *in Proceedings of the 8th International Conference on Machine Learning*, San Francisco, USA, pp. 282-289, 2001.

[10] Mikheev A., "Automatic Rule Induction for Unknown-Word Guessing," *Computational Linguistics*, vol. 23, no. 3, pp. 405-423, 1997.

[11] Orphanos G. and Christodoulakis D., "POS Disambiguation and Unknown Word Guessing with Decision Trees," *in Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics*, Stroudsburg, USA, pp. 134-141,1999.

[12] Ratnaparkhi A., "A Maximum Entropy Model for Part-of-Speech Tagging," available at: http://www.aclweb.org/anthology/W96-0213, last visited 1996.

[13] Schmid H., "Probabilistic Part-of-Speech Tagging using Decision Trees," *in Proceedings of International Conference on new methods in language processing*, Manchester, UK, pp. 44-49, 1994.

[14] Sharma S. and Lehal G., "Using Hidden Markov Model to Improve the Accuracy of Punjabi POS Tagger," *in Proceedings of IEEE International Conference Computer Science and Automation Engineering*, Shanghai, pp. 697-701, 2011.

[15] Zribi C., Torjmen A., and Ben Ahmed M., "A Multi-Agent System for POS-Tagging Vocalized Arabic Texts," *The International Arab Journal of Information Technology*, vol. 4, no. 4, pp. 322-329, 2007

Dinesh Kumar is Associate Professor in Department of Information Technology at DAV Institute of Engineering and Technology, Jalandhar, Punjab, India. He has done BTech degree in Computer Science and Engineering, MTech degree in Information Technology and currently, pursuing PhD degree in Computer Engineering from the Punjabi University, Patiala. He is member of IEEE, ISTE and CSI (Computer Society of India). He has more than 12 years of teaching and research experience. He has supervised more than 10 MTech Students in natural language processing, machine learning and computer networks, image processing.

Gurpreet Josan is Assistant Professor in Department of Computer Science at the Punjabi University, Patiala, India. He holds PhD degree in Computer Science from the Punjabi University in addition to MTech degree in Computer Engineering. He has more than 12 years of teaching and research experience. He has supervised many MTech students and is supervising five PhD students in natural language processing, machine learning and computer networks. He also leads and teaches modules at both B.Tech and M.Tech levels in computer science.