SVPOST: A Part-of-Speech Tagger for Tagalog using Support Vector Machines

Kevin Rainier Sinogaya Suba
University of the Philippines
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Abigail Razon
University of Birmingham
10 PUBLICATIONS **10** CITATIONS
SEE PROFILE

SVPOST: A Part-of-Speech Tagger for Tagalog using Support Vector Machines

Camille Dominique E. Reyes, Kevin Rainier S. Suba
Abigail R. Razon, Prospero C. Naval, Jr.
Computer Vision and Machine Intelligence Group
Dept. of Computer Science
College of Engineering
University of the Philippines Diliman
cvmig@engg.upd.edu.ph

ABSTRACT

This paper explores the use of Support Vector Machines and bi-grams in the Part-of-Speech Tagging of Filipino words through the creation of SVPOST. The great potential of this approach is seen as it was able to achieve a highest accuracy rating of 81%, which is 2% higher than the highest accuracy achieved by currently available POS taggers for Filipino. Experiments were conducted in order to determine the effect of SVM parameters, such as its kernel or gamma, in the accuracy of the tagger.

1. INTRODUCTION

Natural Language Processing of the Filipino language is still a young field of research with relatively fewer studies compared to other languages. Thus, there is a need to conduct further research of this field in our country and in order to do so, a fundamental task in NLP, which is Part-of-Speech tagging, needs to be addressed. Many other NLP applications, such as grammar checking, require Part-of-speech tagging prior to further computations and analysis, therefore, there is a need for a Part-of-Speech tagger that is highly accurate, robust, and flexible. The use of Support Vector Machines offer a promising opportunity to create a POS Tagger that guarantees the accuracy, robustness and flexibility desired. Thus, in this paper, a new approach to Filipino Part-of-Speech tagging namely, Support Vector Machines, is discussed.

2. PART-OF-SPEECH TAGGING

Part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, refers to the process of marking up the words in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context ,i.e. its relationship with adjacent and related words in a phrase, sentence, or paragraph [1].

It is considered a fundamental problem in the field of Natural Language Processing (NLP) as most NLP applications require some form of POS tagging prior to further data processing and analysis.

In the past decades, there have been numerous literatures regarding POS tagging in languages other than Filipino; however, it can be considered that the field of NLP research in the Philippines is still at its infancy and while there have been several attempts to create a POS Tagger for Filipino in recent years, none of these have involved the use of Support Vector Machines.

3. EXISTING PART-OF-SPEECH TAGGERS FOR TAGALOG

There are several existing Part-of-Speech Taggers for Tagalog, and in 2007, Miguel [8] did a comparative evaluation of four existing Tagalog POS Taggers namely TPOST,MBPOST, PTPOST, and Tag-Alog.

3.1 TPOST: A Template-Based, n-gram Part-Of-Speech Tagger for Tagalog

In 2004, Cheng & Rabo [2] designed a Template-based POS Tagger for Tagalog using n-gram analysis achieving an accuracy rating of 70% for unseen data. They noted that existing algorithms for POS tagging for English, such as Baum-Welch, aren't easily adaptable for Tagalog because of the differences between subject-verb agreements between the two languages. Furthermore, they stated that the standard tagset for English, the Penn TreeBank Tagset, is not completely applicable to all languages, including Tagalog.

3.2 MBPOST: Memory-Based Part-Of-Speech Tagger.

MBPOST is a Memory-Based POS Tagger implemented by Raga and Trogo in 2006[9]. MBPOST utilizes a collection of sentence templates, predefined words and feature value pairs for its training phase. MBPOST then checks its testing data for words that are predefined along with a similarity metric for determing a word's tag. It also utilizes sentence templates for disambiguation. MBPOST achieved an accuracy of 77% for unseen data when stemming is not implemented[8].

3.3 PTPOST: Probabilistic Tagalog Part-of-Speech Tagger.

PTPOST, implemented by Cortez et al. in 2005[3], is a Probabilistic POS Tagger for Tagalog that utilizes the concepts of Hidden Markov Models, the Viterbi algorithm and lexical and contextual probabilities to determine the correct Part-of-Speech tag of a word. PTPOST achieved a highest accuracy rating of 78.3% for unseen data when stemming is not implemented[8].

3.4 Tag-Alog: A Rule-Based Part-Of-Speech Tagger For Tagalog.

Tag-Alog is a Rule-Based Tagalog POS Tagger implemented by Fontanilla and Wu in 2006[5]. It utilized a database of word-tag pairs with tag refinement using a patch list. Tag-Alog achieved an average accuracy of 72.5% and was noted to perform poorly with regards to tagging words that are not part of its database[8].

4. SUPPORT VECTOR MACHINES

Support Vector Machines (or SVM) are a set of kernel-based, supervised learning methods mainly used for classification and regression analysis that analyze data and recognize patterns. SVM has been succesfully applied to numerous practical problems, including those under NLP [4] and in 2003, Giménez & Márquez created an SVM POS Tagger used to tag words in the English and Spanish language. Their tagger achieved an overall accuracy of 97% with a 94% accuracy on ambiguous words[6].

5. DATA AND DATA PREPROCESSING

Our data consists of a corpus of 122,318 tagged words composed of 14,270 unique words and symbols while our tagset consists of 64 tags. Our corpus consists of of texts from different domains such as Business, Entertainment and Rizal Novels.

SVPOST utilizes the use of bigram patterns and from our data' we generated 74,479 unique bigram patterns.

Prior to feeding our data onto SVPOST, the data is formatted in a way such that each text is broken down into sentences and each sentence is broken down into tokens separated by spaces. We do not utilize a spellchecker nor do we use a stemmer; we assume that all our data is correct i.e. with regards to spelling, grammar etc.

5.1 SVM Methodology

5.1.1 Data Preprocessing

The data is first further processed onto csv (Comma separated value) files. A table composed of each tag in the tagset used is created to hold and index each tag (e.g. the second element in the table corresponds to the 2nd tag in the tagset). This index was then used as a reference in the creation of the input file. The tagset table used for SV-POST, which was based on the revised tag set of Cheng & Rabo [2] is shown below.

Given a tag, the indeces corresponding to the 2 tags before and after it were and placed into the csv file. The tag which corresponds to the word being processed is then appended to the csv file. A sample of the data in the csv file is shown in the image below.

5.1.2 Separation of Data

The data (as seen in Figure 3) is then separated into two: A matrix which contains the indeces corresponding to the two tags before and after a given tag, and a vector which contains the given tag.

Table 1: Tagset Table

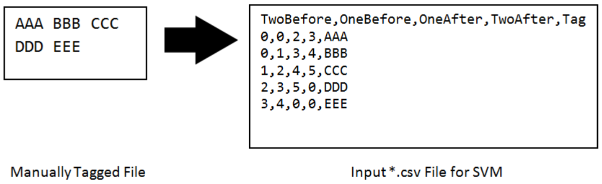
Index	Tag	Index	Tag
1	CCA	33	PRQP
2	CCB	34	PRS
3	CCP	35	PRSP
4	CCR	36	RBB
5	CCT	37	RBD
6	CDB	38	RBF
7	DTC	39	RBI
8	DTCP	40	RBJ
9	DTP	41	RBK
10	DTPP	42	RBL
11	JJC	43	RBM
12	JJCC	44	RBN
13	JJCN	45	RBP
14	JJCS	46	RBQ
15	JJD	47	RBR
16	JJN	48	RBS
17	LM	49	RBT
18	NNC	50	RBW
19	NNP	51	TS
20	NNPA	52	VBAF
21	PMC	53	VBH
22	PME	54	VBN
23	PMP	55	VBOB
24	PMQ	56	VBOF
25	PMS	57	VBOL
26	PRC	58	VBRF
27	PRF	59	VBS
28	PRI	60	VBTF
29	PRL	61	VBTP
30	PRO	62	VBTR
31	PRP	63	VBTS
32	PRQ	64	VBW

5.1.3 SVM Proper

SVPOST is implemented using the software R with the package *e1071*. The training data is then fed into the SVM and a model is made. This model is then used to evaluate the tags of the test and validation sets.

6. SIMULATION DETAILS AND RESULTS

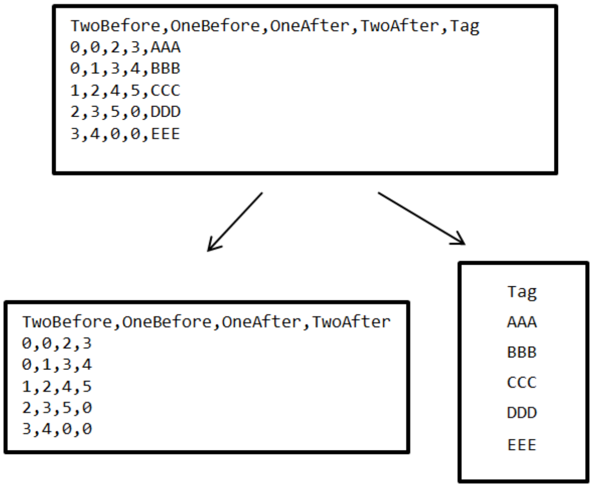
Our experiments concentrated in comparing the tags generated by our system and the tags provided in our corpora



```
AAA BBB CCC
DDD EEE
```

```
TwoBefore,OneBefore,OneAfter,TwoAfter,Tag
0,0,2,3,AAA
0,1,3,4,BBB
1,2,4,5,CCC
2,3,5,0,DDD
3,4,0,0,EEE
```

Figure 1: Sample Input File Conversion



```
TwoBefore,OneBefore,OneAfter,TwoAfter,Tag
0,0,2,3,AAA
0,1,3,4,BBB
1,2,4,5,CCC
2,3,5,0,DDD
3,4,0,0,EEE
```

```
TwoBefore,OneBefore,OneAfter,TwoAfter
0,0,2,3
0,1,3,4
1,2,4,5
2,3,5,0
3,4,0,0
```

```
Tag
AAA
BBB
CCC
DDD
EEE
```

Figure 2: Sample Input File Separation

which were ran on a Mac Pro (2 x 2.26 GHz) with 16 GB of RAM running Mac OSX 10.6.

For our experiements, the data we had were separated using a 60/20/20 ratio for our training, testing, and validation sets prespectively.

For the first experiment, we sought to determine which kernel setting produces the most accurate result. We chose to create SVM models with the following kernel settings:

1. A model with a radial kernel
2. A model with a linear kernel
3. A modet with a polynomial kernel

Outlined in the tables are the parameters, running times and results of the experiment.

Table 2: Parameters of the Models Created

Kernel	Degree	Cost	Gamma	Number of Support Vectors
Radial	1	1	0.25	54282
Linear	1	1	0.25	53711
Polynomial	3	1	0.25	53847

Table 3: Running Time for Each SVM Model

Kernel	Training Time	Testing Time
Radial	1046 seconds	129 seconds
Linear	848 seconds	75 seconds
Polynomial	1068 seconds	119 seconds

Based on the results, the model with a radial kernel was able to produce the best results with an accuracy of 76.80% for our test set and an accuracy of 78.24% for our validation set. Although it produced the best results, this model took the longest time to train and took the longest time to test the tags of the test and validation sets. However, it's accuracy is about 6% higher than the linear kernel model and 10% higher than the polynomial kernel model.

Table 4: Accuracy ratings based on Kernel Variation using SVPOST and bi-grams

Kernel	Test Set	Validation Set
Radial	76.80%	78.24%
Linear	70.07%	69.60%
Polynomial	66.25%	68.12%

Given that in the previous experiment, a model with a radial kernel produces the most accurate results, we wanted to find out if changing the gamma, which is related to the data dimension of the model, has an effect in the accuracy of the SVM tagger. The gamma is obtained through the following equation:

$$gamma = \frac{1}{data\ dimension}$$

We chose to create three models with a raidal kernel with the following gamma configurations:

1. A radial model whose gamma is .25
2. A radial model whose gamma is .5
3. A radial model whose gamma is .125

Outlined in tables 5,6 and 7 are the parameters, running time and results of the experiment.

Table 5: Parameters of the Models Created

Kernel	Degree	Cost	Gamma	Number of Support Vectors
Radial	1	1	0.25	54282
Radial	1	1	0.5	54165
Radial	1	1	0.125	54314

Table 6: Running Time for Each SVM Model

Gamma	Training Time	Testing Time
0.25	1066 seconds	129 seconds
0.5	1052 seconds	126 seconds
0.125	1248 seconds	145 seconds

Based on the results, the radial model with a gamma of .5 was able to produce the best results with an accuracy of 79.50% on our test set and an accuracy of 81.37% when ran on our validation set.

Table 7: Results based on using a Radial Kernel, Bi-grams and Gamma Variation

Gamma	Test Set	Validation Set
0.25	76.80%	78.24%
0.5	79.50%	81.37%
0.125	73.30%	74.06%

7. CONCLUSIONS

Based on the two experiments conducted, the following conclusions were generated:

1. The use of SVMs seem to be a promising approach in Part-of-Speech tagging producing an accuracy as high as 81%

2. An SVM model with a radial kernel produces the most accurate results though it takes a longer time to build the model and generate the tags of the test data

3. A larger gamma produces more accurate results given an SVM model with a radial kernel

SVPOST's accuracy was along the lines of the accuracies of the POS taggers mentioned in Section 3. It was even able to surpass the tagger with thehighest accuracy among the previous taggers by about 1%. This shows that using Support Vector Machines is a promising approach when it comes to Part-of-Speech tagging.

8. FUTURE WORK

Future works that can be done for this project consists of the following:

1. Increase the dataset in order to accomodate more words and to cover more domains

2. Further tweaking of certain SVM parameters like its gamma to further increase the model's accuracy

3. Implementation of SVM using binary inputs in order to achieve higher accuracies and, if possible, lessen training time

9. ACKNOWLEDGEMENT

We would like to thank Dr. Rowena Cristina L. Guevara of the Institute of Electrical and Electronics Engineering for guiding us in the development of the project.

We would also like to acknowledge Dr. Rachel Edita O. Roxas from the Center for Language Technologies of De La Salle University for letting us use their data.

10. REFERENCES

[1] Brill, E. *Part of Speech Tagging.* Handbook of Natural Language Processing, Microsoft Research (2000), USA, Redmond, Washington.

[2] Cheng C., Rabo, V. *TPOST: A Template-based, n-gram Part-of-Speech Tagger for Tagalog.* MSCS Thesis. De La Salle University- Manila, 2004.

[3] Cortez, A., Navarro, D.J., Tan, R., Victor A. *PTPOST: Probabilistic Tagalog Part-of-Speech Tagger.* De La Salle University-Manila, 2005.

[4] Cristianini, N., Shawe-Taylor J. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[5] Fontanilla, G. K., Wu, H.W. *Tag-Alog: A Rule-Based Part-Of-Speech Tagger For Tagalog.* De La Salle University, Manila, 2006.

[6] Giménez, J.,Márquez, L. *Fast and Accurate Part-of-Speech Tagging: The SVM Approach Revisited.* Proceedings of the International Conference RANLP - 2003 (Recent Advances in Natural Language Processing), pages 158 - 165. September, 10-12, 2003. Borovets, Bulgary.

[7] Giménez, J.,Márquez, L. *SVMTool: A general POS tagger generator based on Support Vector Machines.* Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), vol. I, pages 43 - 46. Lisbon, Portugal, 2004.

[8] Miguel, D., Roxas, R.. *Comparative Evaluation of Tagalog Part of Speech Taggers.* Proceedings of the 4th National Natural Language Processing Research Symposium, 2007.

[9] Raga, R. Jr., Trogo, R. *MBPOST: Memory-Based Part-Of-Speech Tagger.* De La Salle University-Manila, 2006.