SMTPOST: Using Statistical Machine Translation Approach in Filipino Part-of-Speech Tagging

Nicco Nocon
De La Salle University
2401 Taft Avenue, Malate, Manila City
1004 Metro Manila, Philippines
noconoccin@gmail.com

Allan Borra
De La Salle University
2401 Taft Avenue, Malate, Manila City
1004 Metro Manila, Philippines
allan.borra@dlsu.edu.ph

Abstract

The field of Natural Language Processing (NLP) in the country has been continually developing. However, the transition between Tagalog to the progressing Filipino language left tools and resources behind. This paper introduces a Statistical Machine Translation Part-of-Speech (POS) Tagger for Filipino (SMTPOST), with the purpose of reviving, updating and widening the scope of technologies in the POS` tagging domain, catering to the changes made by the Filipino language. Resources built are comprised mainly of a tagset (218 tags), parallel corpus (2,668 sentences), affix rules (59 rules) and word-tag dictionary (309 entries). SMTPOST was tested to different tagsets and domains, producing 84.75% as its highest accuracy score, at least 3.75% increase from the available Tagalog POS taggers. Despite SMTPOST's utilization of Filipino resources and good performance, there are room for improvements and opportunities. Recommendations include a better feature extractor (preferably a morphological analyzer), an increase in scope for all of the resources, implementation of pre- and/or post-processing, and the utilization of SMTPOST research to other NLP applications.

1 Introduction

Natural Language Processing (NLP) is a field in computer science where it connects human language with technology. In the Philippines, NLP applications and resources have been continually expanding. Specifically, a project conducted by De La Salle University (DLSU), Manila in the span of three years developed numerous NLP products: from language resources such as lexicons, word corpora, tagsets and grammar rules, to tools such as Morphological Analyzers, Part-of-Speech (POS) Taggers, Grammar Checkers and Machine Translators (Chu, 2009). These outputs enabled DLSU to produce research papers and extended applications not only for the Filipino language, but also to English, marking these works as well-established at that time.

Focusing on POS tagging[1], Chu (2009) featured taggers from Miguel and Roxas' (2007) comparative study. These POS taggers were implemented on different approaches: PTPOST4.1 (Go, 2006) an extension from past PTPOST researches (Cortez et al., 2005; Flordeliza et al., 2005), is a probabilistic tagger implementing the Hidden Markov model, Viterbi algorithm, lexical and contextual probabilities; MBPOST (Raga and Trogo, 2006), a memory-based tagger; Tag-Alog (Fontanilla and Wu, 2006), a rule-based tagger; TPOST (Cheng and Rabo, 2004), a template-based tagger; and adding to the list, SVPOST (Reyes et al., 2011), a Support Vector Machines tagger. Despite developments of POS taggers in the country, the Filipino language's evolution requires constant updates on the tools and their resources. Without these updates, the products become outdated in the following factors: data contents, software usability, performance and availability. This paper addresses those issues through experimentation and creation of a new tagger using Statistical Machine Translation (SMT) for

[1] The process of indicating the Part-of-Speech (i.e. Nouns, Pronouns, Verbs, Adjectives, etc.) of a given word. In this case, the tagging process is automated.

the Filipino language. This research is also intended to provide aid in the understanding of Filipino POS, establish a Filipino tagset and support NLP products or processes (i.e. grammar checker, language parsing, speech processing, information retrieval, etc.) in their tasks.

In choosing an approach, the use of Hidden Markov Models, Viterbi Algorithm, and Machine Learning (Support Vector Machines, Perceptron, and the likes) has been recurrent to foreign languages. As a challenge and motivation for this research, instead of implementing widely used approaches, it has been set to start up new ventures on a potential tagger – ending up with selecting Statistical Machine Translation. SMT as a tagger is uncommon; as specified in its name, it is mainly used in translating one language to another. However, it is not limited to be used that way. Oda et al.'s (2015) work, used SMT for generating English and Japanese pseudo-codes from a given source code, intended to aid code understanding. Other samples are from the work of Mizumoto et al.'s (2011) Japanese error correction and Nocon et al.'s (2014) Filipino shortcut words normalizer. These examples, provided results that proved using SMT in different areas is feasible by supplying two types of data labeled as source (to be transformed) and target (transformed into).

As a data-driven approach, the method for this research leverages SMT by using pairs of word features (source) and POS tag counterparts (target), and translated Filipino Wikipedia data as input for training; while for POS tagging, words or sentences are accepted as input to be automatically transformed into features to match the generated model from training.

This paper mainly focuses on elaborating the creation of Statistical Machine Translation Part-of-Speech Tagger (SMTPOST). It is outlined in the following order: first is the methodology section in which the construction of SMTPOST is discussed; followed by test results and discussions, including analysis of SMTPOST's performance against other existing taggers; next, conclusion and recommendations; and finally, the list of references used.

2 Methodology

In order to create the Filipino Statistical Machine Translation Part-of-Speech Tagger (SMTPOST), the necessary resources and tools were built.

2.1 Language Resources

MGNN Tagset

From the Rabo Tagset (Cheng and Rabo, 2004), tag codes were modified and POS sub-categories were added such as common noun abbreviation, preposition, semi-colon tag and compound (combination of two or more POS) tags. An example for a compound tag, given the word *bagong* 'new', it has the frequency adverb (RBW) *bago* 'new' and the ligature (CCP) *-ng,* resulting to the compound RBW_CCP tag. The MGNN Tagset[2] consists of 218 tags, with 69 basic and 149 (currently used) compound tags.

Corpora

The parallel corpus used was collected from Wikipedia, containing Filipino word and POS tag pairs, with a total of 2,668 sentences or 70,312 (14,575 distinct) words. The parallel corpus was divided into two parts: training and testing data, following 80 (2,134 sentences/55,428 words) to 20 (534 sentences/14,884 words) ratio, respectively.

Additional corpora were gathered from TPOST (i.e. Biblical Text and Children Storybooks) for testing purposes. The numbers designated from their work's training and testing were followed.

All of these data were collected in English and then translated into Filipino by university students whom were supervised by a linguist in the specified language field. There were no specific rules in translating as long as they are consistent (sentences may be in predicate-subject or subject-predicate form), to apply Filipino conversational style and terminologies in the data. The POS counterpart was manually tagged using the MGNN Tagset for Wikipedia and Biblical Text (1) corpora, and Rabo Tagset for Biblical Text (2) and Children Storybooks. Taken from TPOST, Biblical Text (1) and (2) have the same word entries but differ in their POS tag counterparts.

Affix Rules

59 affix rules from Bonus (2003) were used as basis for feature extraction. Rules per affix: prefix, infix, and suffix are distributed in 42, 2 and 15 rules, respectively.

Word-Tag Dictionary

A dictionary containing 309 word and POS tag pair entries (updated from TPOST's predefined

[2] Can be accessed in http://goo.gl/dY0qFe

words) include word samples from each category. It acts as a database for determining words that have POS tags. TPOST used this resource in providing tags, but in this research, it was only used to mark words that are already in the dictionary as part of the feature extraction.

2.2 SMTPOST

SMTPOST's processes follows the framework shown at Figure 1. Processes with the '*' mark were done beforehand and are excluded during the tagging process.

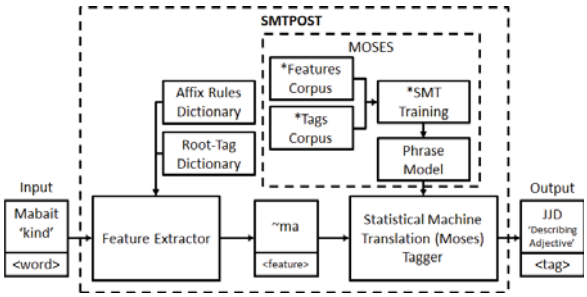

Figure 1. SMTPOST Framework

Feature Extractor (FEX)

FEX takes out word features (affixes) from a given text and inserts marker/s before the found affixes. Following TPOST's structure for extracting and marking features (see Table 1) with the addition of *:A* marker for abbreviations, *kumakain* 'eating' will result into *@um$ka* or in English *+ing*.

Word Feature Structure	
(((#<PDW>)* [:<Capitalized>](~<Prefix>)* (@<infix>)* (+<Suffix>)* ($<DuplicatedCharacters>)*) [-] or *<word>)	
Feature Code	**Description**
#	Predefined Word
:F	1st letter Capitalized
:FS	1st word of Sentence
:A	Abbreviations
~	Prefix
@	Infix
+	Suffix
$	Duplicated Characters
-	Hyphen
*	No Features, whole word

Table 1. FEX Structure and Markers

The algorithm for extracting features was based from Cheng and Rabo's TPOST (2004), migrated from 2004 Java Server Pages into 2016 Java – intended to eliminate dependencies on other POS tagger programs. It utilizes the affix rules and word-tag dictionary to aid in the marking of word features. Using it on training and testing words passes the extracted affixes on as input for SMTPOST. Given this, the input data for SMTPOST is generalized instead of literal words.

Statistical Machine Translation (SMT)

SMT is a translation technique which uses statistical models as its heuristics. By setting a parallel corpus as training input, it determines the patterns and matches of both words and phrases, together with their probabilities. The SMT tagger was implemented using Moses [3] (including SRILM and GIZA++), a well-known and online available SMT tool.

In Moses, there are two main components namely, training and decoding. For training, it requires a set of data to learn from the source and target data. In this research, a Wikipedia parallel corpus was used; but before feeding the data to Moses, it underwent cleaning. Unnecessary characters (e.g. Äì1916 → 1916) and duplicate entries were omitted. At the same time, cleaning involves word correction (e.g. k0lumna → kolumna) and fixing tagging errors such as typographical errors (e.g. JJCC → JJC), incorrect tags (e.g. '.' = PMC → PMP) and tag casing (e.g. PRI_cCP → PRI_CCP).

The cleaned data was originally a word-tag parallel corpus. To generalize the data, word features were generated by running FEX to the words counterpart, producing the feature-tag parallel corpus. This monolingual feature-tag parallel corpus serves as the main data for training, setting word features as source and POS tags as target data.

Following the training pipeline, feeding the data into Moses generated the phrase-model. It contains phrase-table rules (features mapped with tags and their probabilities) with a total of 297,633 lines to be used in POS tagging.

Decoder on the other hand is the tagging proper. It uses the output of training (phrase-model) and sentence/s to be tagged. The accepted input for SMT are extracted features based from input sentence/s and by supplying them, SMT will be able to decode and determine the POS tag – SMTPOST's final output.

[3] http://statmt.org/moses/

Tagging using SMTPOST				
Domain	**Tagset**	**Training Sentences**	**Testing Sentences**	**Accuracy**
Wikipedia	MGNN	2,134	534	84.75%
Biblical Text (1)	MGNN	107	34	77.20%
Biblical Text (2)	Rabo	107	34	84.63%
Children Storybooks	Rabo	68	34	68.72%
Tagging using TPOST (Cheng and Rabo, 2004)				
Wikipedia	MGNN	2,134	534	23.33%
Biblical Text[4]	Rabo	107	34	81.65%
Children Storybooks[4]	Rabo	68	34	61.00%

Table 2. Testing Summary and Results

3 Results and Discussion

SMTPOST was tested through the following domains and tagsets (see Table 2). Additional information on the table are results using TPOST (Cheng and Rabo, 2004), for it is the closest one to the system – in terms of data and process. Data with at most 141 sentences or 2,658 (637 distinct) words per domain were reflected from the same reference in order to enable this research in showing the performance of SMTPOST based on TPOST's testing using different types of corpora.

On the first part of the table, results showed SMTPOST's 84.75% tagging accuracy, where in a total of 14,869 words, the number of correctly tagged, incorrectly tagged and untagged instances are the following: 12,601, 1,577 and 691, respectively. Biblical Text (1) against (2) fell from the line of 8 to 7, with 7.43% difference. This score was the effect of tag specifics and variations using MGNN Tagset, which enhanced the detail in capturing how words are used in a sentence – a deeper POS categorization for a certain word. To illustrate this point, given the words *akin* 'mine' and *aking* 'my', MGNN tags the two words as PRSP (possessive subject pronoun) and PRSP_CCP (possessive subject pronoun with the ligature *-ng*), respectively. On the other hand, Rabo Tagset will simply tag them both as PRSP. Based on the example above, it differentiates independent from dependent possessive pronouns through their single or combined POS tags than generalizing all that falls under a single POS sub-category. With this statement, even if Biblical Text (2) is close to the highest, the use of MGNN Tagset was favored than of Rabo's because of its well tag description for a word and was applied to the training of a modernized Filipino data. About the Children Storybooks domain, it performed poorly with 68.72%. The reason for this is that the data heavily contained proper and common nouns, resulting into a large number of words without features; unlike Wikipedia, the preceded case together with its limited training data prevented both the feature extractor and statistical heuristics from pulling up its accuracy score.

On the second part of the table, the Wikipedia corpus was tagged using TPOST and TPOST's testing results on Biblical Text and Children Storybooks were taken directly from the source for cross-referencing. Tagging the Wikipedia corpus produced 23.33% accuracy, exceedingly low as opposed to the other testing and domains. The testing revealed that similar to Children Storybooks, the Wikipedia corpus contains a heavy amount of nouns and complex terminologies (multiple affixes) which makes it difficult to tag and TPOST was unable to handle its complexity; thus exhibiting SMTPOST's exceptional tagging capabilities.

Comparing results from the two taggers, SMT showed that its results between the same tagset and domain surpassed the template-based approach. It implies that even though both uses generalized data, the use of probabilities in tagging is superior than TPOST's scoring heuristics. Furthermore, evaluation in terms of tagging speed was conducted to both taggers. On the same machine, TPOST tagged 534 sentences for 2 hours and 50 minutes while SMTPOST tagged them for only 26 seconds. Although TPOST's computations are simpler than SMTPOST, TPOST's scoring system were done during the tagging process; whereas, SMTPOST's computations were done during the training process, making the tagging similar to a lookup. Taking an ambiguous word for instance, both taggers will gather the candidate phrases (neighboring words) that will help distinguish the correct tag. After collecting the candidates,

[4] Results taken from Cheng and Rabo (2004) reference.

TPOST will compute how much each candidate fit with the ambiguous word; in contrast, SMTPOST searches for the candidate with the highest probability. Hence, the testing results showed that SMTPOST performs well when it comes to the correctness of its tag while maintaining its decent tagging speed in the process.

Aside from TPOST, SMTPOST was compared to other POS taggers shown at Table 3.

POS Tagger	Data Composition	Accuracy
PTPOST4.1	120,000 words (Miguel and Roxas, 2007)	78.30%
MBPOST		77.00%
Tag-Alog		72.50%
TPOST		70.00%
SVPOST	122,318 words (Reyes et al., 2011)	81.00%
SMTPOST	**70,312 words**	**84.75%**

Table 3. Comparison of POS Taggers

From the given scores, with just 70,312 words, SMTPOST's score exceeded the other taggers by at least 3.75%. The reason for this is other taggers used words for their training, whereas SMTPOST used features which are words in their generalized forms. The effect of using generalized data mainly widens the scope of the tagger, lessening out-of-vocabulary (OOV) words or words unrecognized by the system. For example, in SMTPOST's training data, *kumain* 'ate', *sumayaw* 'danced' and *tumalon* 'jumped' all contains the infix *-um-((@um)*. SMTPOST then creates a rule that whenever an extracted feature is *@um*, it will tag VBTS or past tense verb. When FEX process a word like *tumakbo* 'ran', it will output *@um* and through SMTPOST, it will be tagged as VBTS. Note that SMTPOST considers the probabilities of neighboring features or tags, and features may match words more than the previous examples given, for which both improves the tagging output.

Given the presented results, gaining the highest score among the other taggers demonstrated the utilization of SMT for tagging, at the same time the implementation of Filipino language, generation of word features and accurate generalizations as the basis for tagging were a success. To produce such results, SMTPOST is found to have its own set of advantages and disadvantages. One of the advantages is related to its tagging process, which makes use of generalized data instead of literal ones. Choosing this type of data extends the tagger's scope and lessens the instances of OOV words. Applying statistics as basis is equally as important, for it uses frequency and probability to determine the correct tags, even for phrases and ambiguous words. Moreover, SMTPOST has been tested with different domains, so adaptability is not a problem when its training data is modified.

On the other hand, disadvantages include a weak feature extractor and the lack of training data, hindering SMTPOST to tag complex feature combinations. Common features such as *~mag* 'to …' (future tense), *~nag* '-d or -ed' (past tense) and *+ng* (a word with the ligature *-ng*) are helpful triggers in determining tags for any given word as long as those features appear in it. However, when mixed with additional features, the word features become complicated, thus resulting into errors. An example word feature *~mag~ka~sing+an* from *magkasingkahulugan* 'synonymous'; where the *~mag* prefix feature is present, but joined by other features such as the prefixes *~ka*, *~sing*, and suffix *+an*. Its distinctness made it out-of-vocabulary and as a result made SMTPOST unable to label a POS out of it. In relation to this, OOV features also appear on occurrences of nouns, foreign words, abbreviations and numbers (e.g. *:F*osaka*, **sweldo* 'salary', **box*, *:A*ceo*, **2016*) due to their empty word features – they are marked as "no features" or "whole word". In this case, SMTPOST's data failed to capture these types of words because they were already whole (or in their root) form and not affected by the generalized data. Nevertheless, these uncaptured words come with definite marker patterns which can be resolved through the use of pre- or post-processing tools, hinting on the usage of regular expressions or increase in language resources (pointing out to the corpora and word-tag dictionary).

Overall, in spite of imperfectly extracting word features, the accuracy of the system is high. Acknowledging this, certain and common patterns of words in Filipino were captured by the tagger, making different word variations with the same features most likely fall into one POS category.

4 Conclusion and Recommendations

SMTPOST proved that an unconventional Statistical Machine Translation approach can be used as a Part-of-Speech tagger in Filipino; addressing the factors about existing taggers' data contents, software usability, performance and availability. With 70,312 words from Wikipedia, its highest accuracy score produced 84.75%, at

least 3.75% higher than the other existing taggers. Despite SMTPOST's high accuracy, there are some improvements needed. Recommended for future works are the following: use of a morphological analyzer for feature extraction; increase in scope for all of the resources, aiming at least 100,000 words for the parallel corpus and inclusion of other local and/or foreign languages; utilization of resources built by SMTPOST to other NLP applications; data checks for SMT, to make sure the correctness of the given word-tag pair data; software solutions for lessening complex feature and OOVs; implementation of additional techniques for pre- or post-processing; and finally, usability and availability extensions by using SMTPOST in a NLP software application or deploying it into the web as a service.

Acknowledgments

This research work is supported by the Philippine Council for Industry, Energy and Emerging Technology Research and Development (PCIEERD) of the Department of Science and Technology (DOST), Philippines as part of their research program entitled "Interdisciplinary Signal Processing for Pinoys: Software Applications for Education (ISIP:SAFE)".

References

Bonus, E. (2003). A Stemming Algorithm for Tagalog Words. De la Salle University, Manila.

Cheng, C. K., & Rabo, V. S. (2004). TPOST: A Template-Based, N-gram Part-of-Speech Tagger for Tagalog. Journal Research in Science, Computing and Engineering (JRSCE), 3(1).

Chu, S. (2009). Language Resource Development at DLSU-NLP Lab. The School of Asian Applied Natural Language Processing for Linguistics Diversity and Language Resource Development ADD-4: Language Resource Technology, Bangkok, Thailand, February 23-27, 2009.

Cortez, A., Navarro, D.J., Tan, R., & Victor A. (2005). PTPOST: Probabilistic Tagalog Part-of-Speech Tagger. De La Salle University, Manila.

Flordeliza, J., Go, K., & Miguel, D. (2005). PTPOST4.0: Probabilistic Tagalog Part of Speech Tagging. De La Salle University, Manila.

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

Go, K. (2006). PTPOST4.1 Probabilistic Tagalog Part of Speech Tagger. Class Project. De La Salle University, Manila.

Miguel, D., & Roxas, R. (2007). Comparative Evaluation of Tagalog Part-of-Speech Taggers. In Proceedings of the 4th National Natural Language Processing, De La Salle University, Manila, Philippines, June 14-16, 2007.

Mizumoto. T., Mamoru, K., Nagata, M., & Matsumoto, Y. (2011). Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In Proceedings of the 5th International Joint Conference on Natural Language Processing, pp. 147-155, Chiang Mai, Thailand, November 8-13, 2011.

Nocon, N., Cuevas, G., Magat, D., Suministrado, P., & Cheng, C. (2014). NornAPI: An API for Normalizing Filipino Shortcut Texts. In Proceedings of the International Conference on Asian Language Processing 2014, pp. 207-210, Kuching, Sarawak, Malaysia, October 20-22, 2014. doi: 10.1109/IALP.2014.6973494

Oda, Y., Fudaba, H., Neubig, G., Hata, H., Sakti, S., Toda, T., & Nakamura, S. (2015). Learning to Generate Pseudo-code from Source Code using Statistical Machine Translation. In proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), pp. 574-584, Lincoln, Nebraska, USA, November 9-13, 2015.

Raga, R. Jr., & Trogo, R. (2006). Memory-Based Part-Of-Speech Tagger. De La Salle University, Manila.

Reyes, C. D. E., Suba, K. R. S., Razon, A. R., & Naval, P. C. Jr. (2011). SVPOST: A Part-of-Speech Tagger for Tagalog using Support Vector Machines. In Proceedings of the 11th Philippine Computing Science Congress, Ateneo de Naga University, Philippines.