Journal of Industrial Information Technology and Application Vol. 2 No. 4

ISSN(Online) : 2586-0852

An Enhanced Part-of-Speech Tagger for English

Mary Destiny M. Abellana [1)], Robert R. Roxas [2,*)]

[1,2)] Dept. of Computer Science, College of Science, University of the Philippines
Cebu, Philippines

Abstract. This paper presents a Part-of-Speech (POS) Tagger that can accurately tag words, especially the homographs in a sentence. Our POS Tagger aims to overcome the problems we encountered when using other well-known Taggers for the English Language. Our POS Tagger uses a POS tagged corpus, which contains tag sentences that serve as the syntax as well as the semantic rules and a dictionary (POS-Word Mapping Dictionary) containing all the possible POS tags of a word. Our POS tagger scored an accuracy of ninety-six percent (96%) for our simple test, which is significantly higher than the two Taggers that we compared with. Our POS Tagger achieved the purpose of addressing the issues found in the other Taggers by correctly tagging the homographs.

Keywords; POS tagger; homograph; tagset

1. Introduction

Part-of-speech tagging is a process in which the system accepts sentences as an input and produces the part-of-speech tags of the words as an output of those sentences [1, 2]. The most commonly used methods of POS tagging are rule-based tagging and stochastic tagging [3]. Rule-based tagging is dependent on a dictionary and a set of rules to produce the possible tags, while stochastic tagging produces tags based on the sequence of tags with the highest probability [4]. Stochastic tagging method has been gaining appeal because it is less complex to construct and requires less amount of effort to maintain unlike rule-based tagging. Stochastic tagger, however, requires a large POS tagged

[*] Corresponding author: robert.roxas@up.edu.ph

Received: 2017.10.20; Accepted: 2018.3.15; Published: 2018.12.30

training corpus to produce the possible tags but requires less work to construct [3]. There are several existing POS Taggers [5, 6, 7] available for use such as Stanford Tagger and OpenNLP Maxent Tagger, which are both stochastic taggers. When using the mentioned POS Taggers, however, neither of the two POS Taggers were accurate enough for simple sentences and failed to correctly tag some singular verbs, which have homographs that are plural nouns. Hence, we believe that the development of a new tagger that can overcome that obstacle is necessary.

This paper presents our proposed POS Tagger. Section 2 describes the background of the study. Section 3 discusses the design of the new tagger showing how an input sentence is processed to produce the possible tags and the components used to construct our POS Tagger. Section 4 discusses the results of the tests showing the accuracy of the two mentioned taggers and our POS Tagger.

2. Related Works

Research in the area of POS tagging is a complex but interesting endeavor. It has not yet reached the 100% accuracy rate [1]. Previous researches employed different approaches just to increase the accuracy rates. Approaches include like the Maximum Entropy (Maxent) [2], Virtual Evidence and Negative Training [6], and Ontology-based Approach [7], just to name a few.

When using the Stanford Tagger, it has problems identifying singular and present verbs preceded by a noun phrase containing a noun and an adjective. It sometimes POS tagged the singular verb as a plural noun instead. The OpenNLP Maxent Tagger has a higher score than the Stanford Tagger in a test, but it still exhibited errors produced by the Stanford Tagger.

A study done by Brill [3] developed a rule-based tagger, which has no regard for context. The performance improves incrementally as it recognizes the weaknesses and addresses them. The tagger initially assigns each word its most likely tag by examining a large corpus. Brill's "initial tagger" has two methods of improving performance for words that were not in the training corpus. The first method was to identify capitalized words, which tend to be proper nouns, in which the tagger attempts to fix tagging errors. The second method was to assign a tag to a word, which is most common to words with the same three-letter ending. Such a work demonstrated that stochastic approach is not the only viable approach to POS tagging. Rule-based approach is also promising.

This paper presents a POS tagger that uses a POS tagged corpus in conjunction with rule-based approach. This was motivated by our experience in using a well-known POS

Tagger for the English Language, which cannot correctly tag some simple sentences containing homographs.

3. Our Approach

We developed a POS Tagger that can accurately tag and overcome the problems we encountered when using the Stanford Tagger and OpenNLP Maxent Tagger, even for simple sentences. Our POS Tagger must be able to correctly identify the part-of-speech of a word and more importantly, the part-of-speech of a homograph. Our POS Tagger requires a POS tagged corpus and a dictionary (POS-Word Mapping Dictionary) containing all the possible POS tags of a word.

Our POS Tagger depends on a corpus containing POS tagged sentences. These tagged sentences will serve as the "syntax rules" for determining the possible tag sequence of a given sentence based on the number of occurrences of a particular tag sequence found in the corpus. The same corpus can be used as the "semantic rules" that will be used to calculate the probability of a tag sequence to be the tag sequence corresponding to the given input sentence. The corpus uses the Brown Corpus tagset because of the large amount of tags available, which also caters to the specificity of the tag. The corpus contains simple sentences, which have a sequence of POS tags that are commonly occurring. For example, POS sequence "NNS-VB-NN" is a simple order of tags and is included in the corpus. Figure 1 shows the phases on how a sentence is processed.

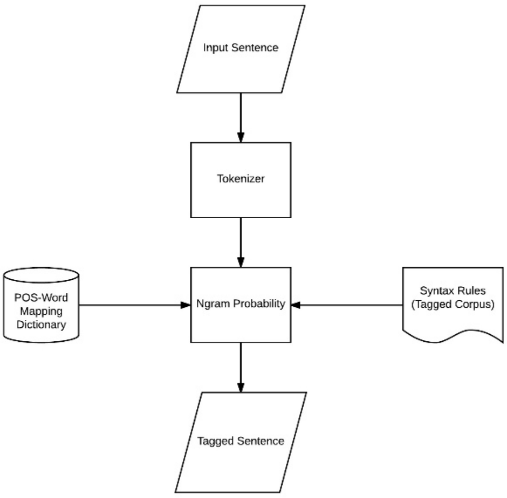
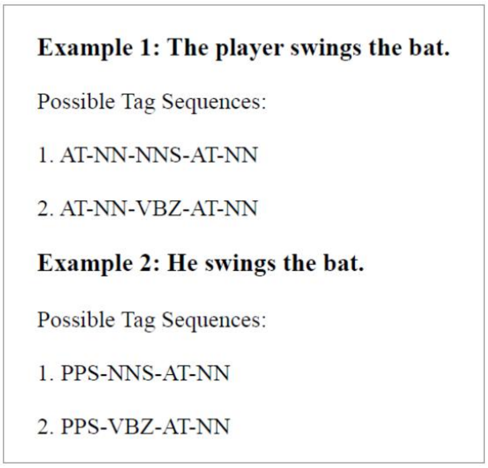


Fig 1. The schematic diagram of our POS Tagger

The process of tagging a sentence happens in four steps as follows:

- The tagger accepts a string as an input sentence.
- It then passes the input sentence to the Tokenizer. The Tokenizer splits the sentence into a list or a sequence of individual tokens.
- For every token (i.e., a word) in the sentence, it is searched in the database (POS-Word Mapping Dictionary) and all the POS tags mapped into the word are returned and then assigned them to that particular word. One word may have more than one POS tags, which occurs in homographs.
- The tagger then generates all the possible combinations of tag sequences for the given sentence based on the assigned tags.

Figure 2 shows two example sentences containing the homograph "swings." Example 1 would produce two possible tag sequences because the word "swings" can either be a plural noun with the corresponding tag "NNS" or a singular verb in the 3rd person with the corresponding tag "VBZ." The same is true for Example 2, which also has two possible tag sequences using the same reason given for Example 1.



Example 1: The player swings the bat.

Possible Tag Sequences:

1. AT-NN-NNS-AT-NN

2. AT-NN-VBZ-AT-NN

Example 2: He swings the bat.

Possible Tag Sequences:

1. PPS-NNS-AT-NN

2. PPS-VBZ-AT-NN

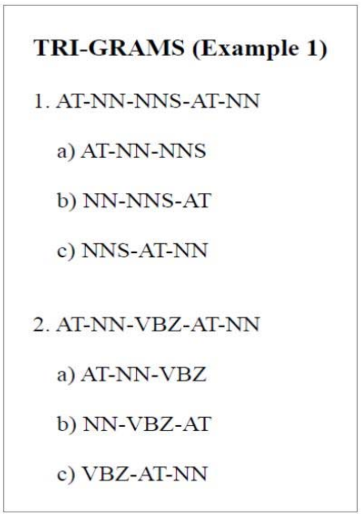
Fig 2. Possible tag sequences of the given examples

When the number of combinations (i.e., tag sequences) is greater than one, the tagger then proceeds to assign a probability to every combination. In calculating the probability, the tag sequence is first split into tri-grams and may include a bi-gram or a uni-gram, if the size of the tag sequence is less than 3. This is accomplished through a sequence of steps:

1. Get the length of the tag sequence.

2. If the length of the tag sequence is less than three, return the bi-gram, if length is equal to two or the uni-gram, if length is equal to one. Otherwise, if the length is equal to or greater than three, get the first 3 tags that form a trigram. The tri-gram is added into the group of tri-grams. Exclude the first tag of the current tag sequence, and treat the remaining tag sequence as the new tag sequence.

3. Repeat step 2 if the length of the new tag sequence is greater than three. Otherwise, treat the new tag sequence as the last tri-gram and should be included to the group of tri-grams.



TRI-GRAMS (Example 1)

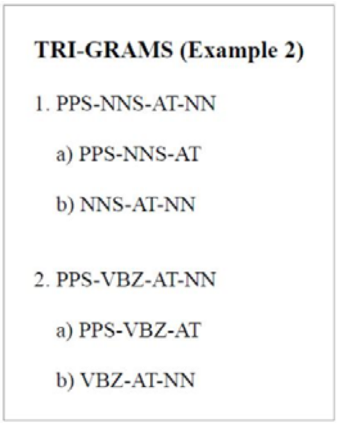
1. AT-NN-NNS-AT-NN
   - a) AT-NN-NNS
   - b) NN-NNS-AT
   - c) NNS-AT-NN

2. AT-NN-VBZ-AT-NN
   - a) AT-NN-VBZ
   - b) NN-VBZ-AT
   - c) VBZ-AT-NN

Fig 3. The resulting n-grams (tri-grams) for Example 1 of Figure 2

Each n-gram probability is calculated using the chain rule, and all the probability of all the n-gram in a combination is multiplied together to get the probability of the combination. The combination with the highest probability is used as the tag of the sentence. For Example 1 shown in Figure 2, the resulting tri-grams are shown in Figure 3. After computing the probability of the resulting n-grams, (trigrams in this example), it would result to selecting the 2nd tag sequence because in the POS tagged corpus, the case where a noun (NN) is followed by a verb (VBZ) is higher than a noun (NN) followed by a plural noun (NNS), and also the case where the verb (VBZ) is followed by an article (AT) is also higher than a plural noun (NNS) followed by an article (AT).

In the case of Example 2 of Figure 2, The resulting tri-grams are shown in Figure 4. Our tagger would select the 2nd tag sequence also. This is because the case where a pronoun (PPS) is followed by a verb (VBZ) has higher probability than a pronoun (PPS) followed by a plural noun (NNS).



TRI-GRAMS (Example 2)

1. PPS-NNS-AT-NN
   a) PPS-NNS-AT
   b) NNS-AT-NN

2. PPS-VBZ-AT-NN
   a) PPS-VBZ-AT
   b) VBZ-AT-NN

Fig 4. The resulting n-grams (tri-grams) for Example 2 of Figure 2

4. Results and Discussion

An experiment comparing the Stanford Tagger, OpenNLP Maxent Tagger, and our POS Tagger was made. Table 1 shows the accuracy of the mentioned three POS Taggers. A total number of twenty-five simple sentences was used to test the accuracy of the three POS Taggers with some of the sentences containing a noun followed by homograph, which can be both a plural noun and a singular verb. For example, the sentence “He swings the bat.” contains a singular verb “swings.” The word “swings” is also a plural form of the noun “swing.” The scenario will test the taggers whether or not it can correctly recognize the verb as a verb rather than a plural noun.

TABLE I. POS TAGGERS AND TEST SCORES

POS Taggers	Score
Stanford POS Tagger	68%
OpenNLP Maxent Tagger	88%
Our POS Tagger	96%

For the sample sentences that we used for testing, the Stanford Tagger scored an accuracy of sixty-eight percent (68%). In sentences containing singular verbs that are homographs to plural nouns, the tagger failed to tag the verb used as a singular present tense verb and tagged the word as plural noun instead. The OpenNLP Maxent Tagger using the maximum entropy model scored an accuracy of eighty-eight percent (88%). But this Tagger still encountered the same problem encountered when using the Stanford Tagger. Our POS Tagger scored an accuracy of ninety-six (96%) by correctly tagging the verbs that were tagged incorrectly by the other two Taggers.

5. Conclusion and Future Work

The development of a POS tagger has been described in this paper. Our POS Tagger scored an accuracy of ninety-six percent (96%) using the sample sentences, which contained homographs. The accuracy of our POS Tagger is heavily attributed to the use of the dictionary for the word mapping and the POS tagged corpus, which contains POS tagged sentences. Although our POS Tagger was only able to handle simple sentences, the algorithm is a good start to derive new ways of POS tagging. This study can also be a stepping stone in researching for a tagger that is not necessarily a purely stochastic nor purely rule-based but also on other methods of POS tagging, more specifically on hybrid POS tagger to widen the boundaries of POS tagging.

References

[1] C. D. Manning, "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?," in Gelbukh A.F. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2011. Lecture Notes in Computer Science, vol 6608. Springer, Berlin, Heidelberg, 2011, pp. 171-189.

[2] K. Toutanova and C. D. Manning, "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger," in 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 2000, pp. 63-70.

[3] E. Brill, "A Simple Rule-Based Part of Speech Tagger," in Proceedings of the Third Conference on Applied Natural Language Processing. Trento, March 1992, pp. 152-155.

[4] F. M. Hasan, "Comparison of different POS Tagging techniques for some South Asian Languages. An undergraduate thesis, BRAC University, Dhaka, Bangladesh, 2006, unpublished.

[5] B. G. Patra, K. Debbarma, D. Das, and S. Bandyopadhyay, "Part of Speech (POS) Tagger for Kokborok," in Proceedings of COLING 2012: Posters, COLING 2012, December 2012, pp. 923-932.

[6] S. M. Reynolds and J. A. Bilmes, "Part-of-Speech Tagging using Virtual Evidence and Negative Training," in Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language

[7] Processing (HLT/EMNLP), October 2005, pp. 459-466.

[8] M. Sukhareva and C. Chiarcos, An Ontology-based Approach to Automatic Part-of-Speech Tagging Using Heterogeneously Annotated Corpora," in Proceedings of the Second Workshop on Natural Language Processing and Linked Open Data, September 2015, pp. 23-32.