CRFPOST: Part-of-Speech Tagger for Filipino Texts using Conditional Random Fields

John Francis T. Olivo
Department of Computer Science
Polytechnic University of the Philippines
Manila, Philippines
olivo.jfrancis@gmail.com

Prince Julius T. Hari
Department of Computer Science
Polytechnic University of the Philippines
Manila, Philippines
princejulius230@gmail.com

Michael B. dela Fuente
Department of Computer Science
Polytechnic University of the Philippines
Manila, Philippines
mbdelafuente@pup.edu.ph

ABSTRACT

Classifying and tagging words into different lexical classes, as a fundamental process in language processing, is necessary to be addressed given the constant evolution of language, and in this case is the Filipino language. As a part of this effort, the researchers introduce in this paper a Linear-chain Conditional Random Fields (CRF) Part-of-Speech Tagger for Filipino texts with CRF providing an edge in sequence labelling as compared to generative models and other classifiers. The tool developed utilized a tag set containing 218 POS tags (69 basic and 161 compound) and Filipino text corpus with 15,166 sentences randomly picked from Wikipedia and translated to Filipino by students under linguist supervision. After experimentation, the researchers show that there is a 90.59% accuracy rate for tagging Filipino texts using CRF for POS tagging. Despite CRFPOST's utilization of word and tag sequence features produces a high performance in tagging, there are still improvements for future work. Recommendations are the inclusion of linguistic tools such as morphological analyzer and named entity recognition for better performance.

CCS CONCEPTS

• Computing Methodologies ~ Natural Language Processing
• Computing Methodologies ~ Language resources

KEYWORDS

Conditional Random Fields; Part-of-Speech Tagging; Filipino

ACM Reference format:

John Francis T. Olivo, Prince Julius T. Hari and Michael B. dela Fuente. 2019. CRFPOST: Part-of-Speech Tagger for Filipino Texts using Conditional Random Fields. In *Proceedings of 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI'19). Sanya, China, 6 pages.* https://doi.org/10.1145/3377713.3377788

ACAI '19, December 20–22, 2019, Sanya, China

ACM ISBN 978-1-4503-7261-9/19/12…$15.00
https://doi.org/10.1145/3377713.3377788

1 Introduction

Natural Language Processing (NLP) is a field in the study of Computer Science that links the understanding of natural language with technology [1] and the application and utilization of computer techniques to linguistic analysis [2]. The development of language-specific tools and resources requires funding and are often time-consuming which poses as a serious problem for most of the world's languages [3].

The work of [4] provided insights into the trends, directions, and future work for Filipino language resources listing down attempts for the construction of lexicons, morphological information, grammar, and corpora. Additionally, existing language resources for Filipino as documented by [5] included lexicons (or vocabulary), word corpora, tag sets, and grammar rules. Yet the Filipino language is considered to be a resource-poor language [6] and when compared to other languages, has free word-order pattern [7], [8] thus needing a different approach in the construction of language tools specifically focusing on its morpho-syntactic structure and analysis and requiring a language resource with syntactic (and in some cases includes semantic) annotations.

As the Filipino language is constantly changing and evolving, this work would like to address the problem of language processing by focusing on the fundamental process of tagging parts of speech. A Part-of-Speech (POS) tagger according to [9] as cited by [10], is a computer software for classifying and tagging words into its word classes or lexical categories. The process is necessary in achieving the purpose of any NLP systems [11] ranging from machine translation, information retrieval and/or extraction, and document classification, summarization, routing and indexing aside from the fact that there is a continuous evolution in the Filipino language, POS tagging remains to be of great problem to address.

Various systems and algorithms have been employed to produce better performing POS tagger for the Filipino language. Previous works utilized Hidden Markov Models (HMM) [12], memory based approach [13], template-based approach [14], rule-

based approach [15], and support-vector machines (SVM) [16]. There has been little work on the application of Conditional Random Fields (CRF), specifically Linear-chain CRF, for Filipino part-of-speech tagging. It was shown that CRF offers advantages for sequence labelling as compared to generative models and other classifiers [17].

This research focuses on the implementation of a Conditional Random Field-based Part of Speech Tagger for Filipino texts aiming to provide adequate solution to the lack of language tools, language resources, as well as the problem of continuous evolution of the language. This paper is organized as follows: section 2 shows the related works to this research, section 3 presents the methodology for constructing CRFPOST along with the language resources discussion, then section 4 presents the Results and Discussion, followed by Conclusion and Recommendations in section 5.

2 Related Works

Recent works from the Polytechnic University of the Philippines – Department of Computer Science involving natural language processing have focused mainly on the tasks of machine translation (for language and for solving word problems), sentiment analysis, text analysis, named entity recognition, and corpus building for the Filipino language. Not many have taken to consider the problem of part-of-speech tagging with the possible thought of it as a minimal problem to consider in the field. Yet this process is essential to majority of language processing researches.

The De La Salle University's Center for Language Technologies have produced works on the improvement of the approaches for Filipino Part-of-Speech tagging which made a good accuracy for the task. As cited by Nocon and Borra [1], the following POS taggers are implemented using different approaches: PTPOST4.1 by [12] uses a probabilistic part-of-speech tagger for Tagalog implementing HMM, Viterbi algorithm, and lexical and contextual probabilities; and MBPOST [13] implements a memory-based approach to tagging. Latest research works in Filipino POS taggers include the SMTPOST [1] which implements an unconventional statistical machine translation approach for POS tagging; HPOST in Gramatika [18] harnesses the combined statistical machine translation, rule-based, and regex-based approaches; and FSPOST [10] that utilizes the trainable POS tagger of Stanford implementing Maximum Entropy approach.

But as previously mentioned by Nocon and Borra [1], the evolution of the Filipino language requires constant updates on the tools and their resources with consideration to the following factors: data contents, software usability, performance, and scalability.

One approach showing encouraging performance in a number of natural language processing applications (such as POS tagging, shallow parsing or NP chunking and named entity recognition) is Conditional Random Fields. In the work of Adafre [19], CRF was applied to the tasks of Amharic word segmentation and POS tagging using an annotated corpus of 1000 words. The paper explored related languages such as Arabic and Hebrew for the recent development in the morphological analysis and machine learning approaches and apply them to the Amharic language. The Amharic language belongs to the Semitic family of languages that shares a number of common morphological properties with Arabic and Hebrew. According to Adafre, CRF allows to integrate large set of features easily and the morphological features help in predicting Amharic POS tags. Given the size of the data and the large number of unknown words in their test corpus (80%), an accuracy of 84% for Amharic word segmentation and 74% for POS tagging was the result of the experimentation indicating the applicability of CRFs for a morphologically complex language like Amharic.

In the study of Ekbal, Haque, and Bandyopadhyay [20], they developed a part of speech (POS) tagger for Bengali using the statistical Condition Random Field. Bengali is one of the widely used languages all over the world. It is the seventh popular language in the world, a secondary language in India and national language of Bangladesh. Their CRF-based part-of-speech tagger, along with the word level suffix features, a lexicon and a HMM based Named Entity Recognizer to tackle the unknown words, has been trained on a corpus of 72,341 word forms. This 26-POS tagged training corpus was obtained from the NLPAI_Contest06 and SPSAL2007 contest data. The NLPAI_Contest06 data was tagged with a tag-set of 27 POS tags and had 46,923 word forms and converted into the 26-POS tagged data by defining appropriate mapping. Out of 72,341 word forms, around 15K POS tagged corpus has been selected as development set and the rest was for training their POS. The results show the effectiveness of the proposed CRF-based POS tagger with an accuracy of 90.3%.

According to Sutton & McCallum [21] many tasks involve predicting a large number of variables that depend on each other as well as on other observed variables. The structured prediction methods are essentially combinations of classification and graphical modelling. Combining the ability of graphical models to compact model with the ability of classification methods to perform prediction using large sets of input describes conditional random fields, which is a popular probabilistic method for structured prediction.

Conditional Random Fields are a probabilistic framework for labeling and segmenting structured data, such as sequences, trees and lattices [22]. It was used in many areas, including natural language processing, computer vision, and bioinformatics [21]. CRFs offer several advantages over hidden Markov models and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models as shown in the paper of Lafferty, McCallum, & Pereira [23]. CRF can be pictured as a finite state model with un-normalized transition probabilities. However, unlike some other weighted finite-state approaches, CRFs assign a well-defined probability distribution over possible labeling, trained by maximum likelihood or MAP estimation (rules) [23].

There are models in machine learning that are widely used to solve many different problems, and a popular example for this is

Hidden Markov Models [22]. HMMs have gained a lot of popularity due to their robustness and accuracy. HMM is a generative model and gives the output directly by modeling the transition matrix based on the training data and the results can be improved by providing more data points, but there is no direct control over the output labels. It learns the transition probabilities on its own based on the training data provided. Moreover, if more data points are provided, the improvement of the model include wider variety. CRF on the other hand is a discriminative undirected probabilistic graphical model which outputs a confidence measure [22]. This is really useful in such cases to know how sure the model is about the label produced. In the field of Natural Language Processing (NLP), linear chain CRF is used in different segmentation and sequence tagging tasks such as keywords extraction, named entity recognition, sentiment analysis, part-of-speech tagging, and speech recognition [24]. Moreover, the most similar method for CRF is the Maximum-entropy Markov Model (MEMM) which is also a discriminative probabilistic graphical model. However, MEMM has so called "label bias problem" which is not found with CRF distinguishing the difference between the two.

3 Methodology

In the development of the Part-of-Speech Tagger for Filipino Texts using Conditional Random Fields (CRFPOST), the necessary resources and tools where used and built.

3.1 Language Resources

Text Corpus

The corpus utilized in this paper is the same as the corpus utilized in the FSPOST, HPOST, and SMTPOST. This corpus contains a total of 15,166 sentences composed of 406,509 tokens (54,583 of these tokens are unique). These are sentences which are randomly picked from Wikipedia which are all in English and translated by students under linguists' supervision.

The corpus is divided into a ratio of 80:20 for training and testing of the model where the training set constitutes of 12,133 sentences and 3,033 sentences for testing.

Tag set

This research utilized the MGNN1 tag set which is also utilized for the corpus annotation. The MGNN tag set contains a total of 230 tags for different parts of speech (69 basic tags and 161 compound tags). The compound tags are a combination of one or more basic tags currently available in Filipino.

[1] Accessible in http://goo.gl/dY0qFe

3.2 CRFPOST Modules

Sentence Segmentation

The sentences, or sentence blocks, are preprocessed (see Figure 1). The inputs are passed to a Sentence Segmentation module to detect sentence boundaries and separate each sentence. The input sentences boundaries are detected by the criteria of starting with a capital letter, doesn't start with a number (if ever a number is in the first word it must be spelled out), a space marks boundary for each word, and must end with a period or other punctuation marks that signify end of a sentence.

Given the data set from SMTPOST, a sentence is detected by searching for the end of line ("\n") for each line containing one sentence in the data set.

Word Tokenization

The segmented sentences are then passed to a Tokenization module to detect the tokens (functional units) in each sentence. The tokenized sentences are then stored in a storage file. Since the dataset is already normalized, word pairs (separated with space or a dash) are connected with an underscore such that the system will not separate words which are supposed to be together.

In testing new input sentences, the sentence boundaries are detected by checking the presence of period (given the previous token is not a numeral), or other punctuation marks such as exclamation points and question marks.

The tokenized sentences and the sentences are then utilized as inputs for the Part of Speech Tagging module. This module utilizes Conditional Random Field statistical model implemented using CRF++ toolkit, specifically a Linear Chain Conditional Random Field, to compute the probability of each tag sequence in each sentence. The training using CRF++ will produce a tagger model which contains the probabilities calculated from the training set. This tagger model will be utilized for testing.

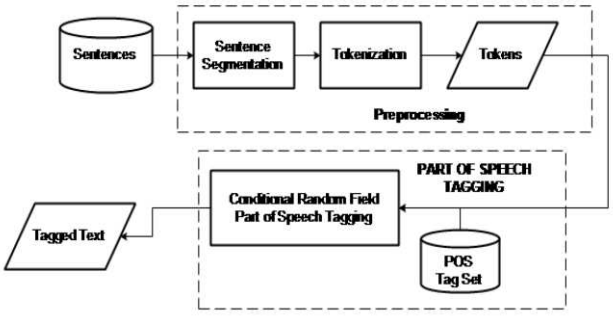

Figure 1: CRFPOST Architecture

Linear-chain Conditional Random Fields

This is to provide an overview on the process of CRF modeling, specifically Linear chain CRF. The sentences, tokens, and tags from training corpora are analyzed in feature functions that compute for certain rules and return a real number that is either a 0 or a 1. These values are multiples to a weight, initially at

random, and are used in computing for the scores of each tag sequence. The formula for the score computation is given below:

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} w_j f_j(s, i, l_i, l_{i-1})$$

The computed scores are stored in a vector. The computed score will be utilized to calculate for the probabilities of each tag sequence. The formula for the probability is given by:

$$p(l|s) = \frac{\exp[score(l|s)]}{\sum_{l'} \exp[score(l|s)]}$$

$$= \frac{exp\left[\sum_{j=1}^{m} \sum_{i=1}^{n} w_j f_j(s, i, l_i, l_{i-1})\right]}{\sum_{l'} \exp\left[\sum_{j=1}^{m} \sum_{i=1}^{n} w_j f_j(s, i, l_i, l_{i-1})\right]}$$

The calculated probabilities are stored as well in a vector. The values are re-computed by using gradient descent to adjust the weights which are initially at random. The gradient descent is given by the formula:

$$\frac{\partial}{\partial x} \log p(l|s) = \sum_{j=1}^{m} w_j f_j(s, i, l_i, l_{i-1})$$

$$- \sum_{l'} p(l'|s) \sum_{j=1}^{m} w_j f_j(s, j, l'_j, l'_{j-1})$$

For each training data, go through each feature function, and calculate the gradient of the log probability of the training example with respect to the weight.

$$w_i = w_i \ \alpha \left[\sum_{j=1}^{m} w_j f_j(s, i, l_i, l_{i-1}) - \sum_{l'} p(l'|s) \sum_{j=1}^{m} w_j f_j(s, j, l'_j, l'_{j-1}) \right]$$

The formula above utilizes α which is the learning rate. The first term is the expected probability while the second term is for the predicted value. This computation is repeated for a certain number of iterations. Once, the probabilities are in place, the model can now be tested by using a decoding algorithm to find the optimal labelling for any input sentences not found in the training data.

CRF++ Toolkit

The CRF++ is a "simple, customizable, and open source toolkit implementation" of Conditional Random Fields which is utilized for segmenting/labeling sequential data.2

The CRF++ toolkit is coded in C++ and allows the users to redefine feature sets. The encoding component utilizes fast training using LBFGS (Limited Memory BFGS), a quasi-newton algorithm for large scale numerical optimization problems. Improvements in the current version of CRF++ utilized includes a more efficient memory resource usage both for encoding and decoding, can perform n-best outputs, can perform single-best MIRA training, and can output marginal probabilities for all candidates.

[2] https://taku910.github.io/crfpp/

4 Results and Discussion

4.1 Comparison of Accuracy

For this study, a dataset which contains a total of 15,166 Filipino sentences from which 80% (12,133 sentences) were utilized for training and 20% (3,033 sentences) was utilized for testing. A total of 406, 509 tokens where generated from which 54,583 tokens are unique. The dataset was annotated manually, under the supervision of a linguist, and contains two types of files that contain (a) the Filipino sentences and (b) the POS tag sequence for each specific sentence. This is the same dataset utilized in the works of SMTPOST [1], HPOST [18], and FSPOST [10].

The research results will be compared to previously mentioned POS Taggers namely: SMTPOST [1], HPOST [18] which was used in Gramatika and an improved version of SMTPOST, and FSPOST [10]. Currently, FSPOST holds the spot as the POS tagger with highest accuracy compared to the Gold standard. All the three taggers, as well as CRFPOST, are tested on the same corpus and utilized the same tag set, MGNN tag set which consists of 218 tags (69 basic tags and 149 currently used compound tags).

Figure 2 shows the results for the testing with FSPOST, a Stanford POS Tagger trained on Filipino sentences, obtaining an accuracy of 96.15%, HPOST a rule-based tagger with an accuracy of 91.63%, followed by CRFPOST at 90.59% accuracy for initial training and testing, and SMTPOST a statistical machine translation-based tagger with 89.11% accuracy.

4.2 Choosing the Model Parameters

Utilizing CRF++ allows its users to model using different parameters that serve as basis on how the training is done. There are four parameters that control the training condition:

- Regularization Algorithm parameter: this allows the user to choose which regularization algorithm will be utilized. CRF++ provides two algorithms which are CRF-L2 and CRF-L1 regularization algorithms. The default is the CRF-L2 which, generally speaking, is better than CRF-L1. Setting this parameter is done using *-a CRF-L2 or -a CRF-L1*
- Fitting value: this parameter allows to have balance between overfitting (overtraining) and underfitting (undertraining) of the model. The fitting value is set using *-c <value>*
- Cut-off Threshold value: this is for setting the cut-off threshold for the features. CRF++ utilizes training features are no less than the value specified in the cut-off threshold value. This is specified using *-f <value>*
- Number of threads: this can be utilized if the computer where the training is done has multiple CPUs which make the training faster by allowing multi-threading. This is done by *-p <value>*

CRFPOST is trained on the dataset on a unigram template. This is specified using the parameters *-f 1 -c 1.5 template.tmp*. The template file specifies the templates for the features of the model. This is the file used for describing features for training and testing.

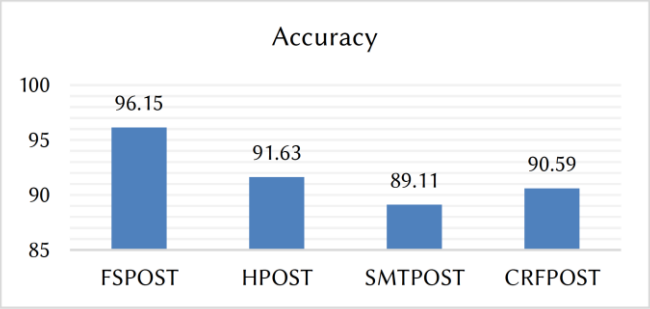

Figure 2: Visualization for Accuracy Comparison

4.3 Error Breakdown and Analysis

To further analyze the results presented in the previous section, the researchers took the tagged results and took the top 10 POS tags with the highest count and distribution patterned from the analysis approach conducted in FSPOST by Go and Nocon [10]. This information is presented in Table 1: Most Frequent Tags.

It can be seen from Table 1 that NNC tag for common nouns has the most numbers of tag in the test corpus with a frequency of 10447 and a distribution percentage of 13.61% meaning that it has the greatest number of words and tags in the test set. This is followed by NNP which is for proper nouns, having a frequency of 7140 and a distribution percentage of 9.59%, CCB is the next top tag which is for one type of conjunctions in Filipino that has a frequency of 5071 and a distribution of 6.27%. Then CCT, tag for another type of conjunctions, follows CCB with a frequency of 4903 and 6.05% distribution percentage. After CCT is the CCP tag which is for another Filipino conjunction which had a frequency of 4029 and 4.97% distribution percentage. This are the top five in the most frequent tags.

Table 1: Most Frequent Tags

POS Tag	Frequency	%Distribution
NNC	10447	13.61%
NNP	7140	9.59%
CCB	5071	6.27%
CCT	4903	6.05%
CCP	4029	4.97%
DTC	3954	4.85%
PMC	3920	4.81%
PMP	3039	3.73%
DTCP	2545	3.12%
PMS	2379	2.93%

Table 2: POS Tags with Most Errors

POS Tag	Error Freq.	Total Freq.	Recall
FW	1094	2902	62.3019
NNP	688	7828	91.2110
NNC	659	11106	94.0663
JJD	593	2111	71.9090
VBW	480	974	50.7187
VBTR	409	1337	69.4091
VBTS	389	1894	79.4615
VBOF	307	675	54.5185
JJD_CCP	224	1438	84.4228
RBD	204	277	26.3538

Table 2: POS Tags with Most Errors shows the most mistagged label after test results of using the CRFPOST. The table also presents the number of times that the tag is wrongly used, the total number of its occurrences in the data set, and the recall percentage. The tag for foreign words, FW, tops for the most mistagged in the whole results. FW achieved a total of 1094 errors with a recall of 62.3019% which is pretty high recall for a tag with almost half wrongly used because of 2902 total occurrences in the data set. This is followed by NNP, which is for proper nouns, that has an error frequency of 688 out of a 7828 with a recall of 91.211% which is still high. After NNP, the NNC achieved highest error frequency with 5071 out of 11106 occurrences garnering a recall of 94.0663%. The figure, Figure 3: Visualization for Most Erroneous POS Tags, below presents the most erroneous POS tags. These errors are due to few occurrences of these tags within the data set for the training, leaving the model to mis tagged multiple entries of this label in the data for test evaluation.

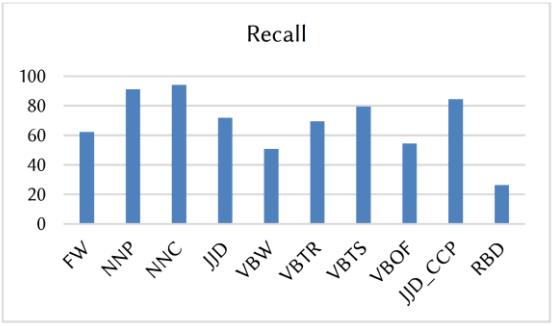

Figure 3: Visualization for Most Erroneous POS Tags

5 Conclusion and Future Works

The implementation of Linear Chain Conditional Random Fields for Part-of-Speech tagging of Filipino texts generated a 90.59% accuracy. In the comparison of POS taggers, there is very little difference between HPOST and CRFPOST, and high values between CRFPOST and SMTPOST as well as with CRFPOST

and FSPOST. The breakdown of errors in tags show that FW, NNP, and NNC has the highest number of mistags due to the low frequency of occurrence in the training data.

Also, this might be due to the ordering of words wherein most of the FW words are common nouns and since CRF is a probabilistic model, it sees that after a specific word succeeded by a FW, a NNC or an NNP is the most probable succeeding tag. NNC and NNP also has a higher frequency throughout the training set, thus, resulting to a higher probability of the NNC or NNP tag to be in the tag sequence other than FW which is for foreign words. The researchers conclude that with better data values and parameters, the CRF model for POS tagging will outperform the other models of POS tagger compared in this paper.

Future works are the inclusion of linguistic tools such as morphological analyzer and named entity recognition for the enhancement of the performance of the tagger to extract more features of words and sequences in the data for training. Also, the utilization of other algorithms found in CRF++ such as CRF L1 for regularization algorithm that can enhance its performance, because CRF L2 is the default as well as the use of MIRA (Margin-infused relaxed algorithm). Moreover, changing the values for cut-off threshold and fitting value is of essence for producing a more customizable and better performing tagger.

ACKNOWLEDGEMENTS

The work will not be accomplished without the expertise and assistance of the people in the linguistic field namely Prof. Perla S. Carpio, Prof. Mary Joy A. Castillo, Prof. Alvin M. Ortiz, Dr. Ricardo Nolasco, Prof. Mayluck Malaga, and Mr. Francisco Muyana. Much thanks are also given to Prof. Nicco Nocon for providing insights and sharing resources to the betterment of this research.

REFERENCES

[1] N. Nocon and A. Borra, "SMTPOST: Using statistical machine translation approach in Filipino part-of-speech tagging," in *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation, PACLIC 2016*, 2016.

[2] R. R. K. Hartmann and G. James, *Dictionary of Lexicography*. New Fetter Lane, London: Routledge by Taylor and Francis Group, 1998.

[3] J. Tiedemann, Ž. Agić, and J. Nivre, "Treebank Translation for Cross-Lingual Parser Induction," 2015, doi: 10.3115/v1/w14-1614.

[4] R. E. Roxas, C. Cheng, and N. R. Lim, "Philippine Language Resources: Trends and Directions," in *Proceedings of the 7th Workshop on Asian Language Resources*, 2009, p. Pages 131-138, doi: 10.3115/1690299.1690318.

[5] S. B. Chu, "Language Resource Development at DLSU-NLP Lab," 2009.

[6] P. Baumann and J. Pierrehumbert, "Using Resource-Rich Languages to Improve Morphological Analysis of Under-Resourced Languages," *Proc. Ninth Int. Conf. Lang. Resour. Eval.*, 2014.

[7] P. Kroeger, "Phrase Structure and Grammatical Relations in Tagalog," 1993.

[8] D. Alcantara and A. Borra, "Constituent structure for Filipino: Induction through probabilistic approaches," in *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation, PACLIC 22*, 2008.

[9] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, Inc. ©2009, 2009.

[10] M. P. V. Go and N. Nocon, "Using Stanford Part-of-Speech Tagger for the Morphologically-rich Filipino Language," in *31st Pacific Asia Conference on Language, Information and Computation (PACLIC 31)*, p. pages 81–88.

[11] D. D. Miguel and R. E. O. Roxas, "Comparative Evaluation of Tagalog Part-of-Speech Taggers," in *4th National Natural Language Processing 2007*, 2007, pp. 74–77.

[12] K. Go, "PTPOST4.1 Probabilistic Tagalog Part of Speech Tagger," De La Salle University, Manila, 2006.

[13] R. J. Raga and R. Trogo, "Memory-based Part-of-Speech Tagger," De La Salle University, Manila, 2006.

[14] C. K. Cheng and V. S. Rabo, "TPOST: A Template-based, N-gram Part-of-Speech Tagger for Tagalog," *J. Res. Sci. Comput. Eng.*, vol. 3, no. 1, 2004.

[15] G. K. Fontanilla and H. W. Wu, "Tag-Alog: A Rule-based Part-of-Speech Tagger for Tagalog," De La Salle University, Manila, 2006.

[16] C. D. E. Reyes, K. R. S. Suba, A. R. Razon, and P. C. J. Naval, "SVPOST: A Part-of-Speech Tagger for Tagalog using Support Vector Machine.," in *Proceedings of the 11th Philippine Computing Science Congress*, 2011.

[17] H. M. Wallach, "Conditional random fields: An introduction," *Neural Comput.*, 2004, doi: 10.1162/jmlr.2003.3.4-5.993.

[18] M. P. Go, N. Nocon, and A. Borra, "Gramatika: A grammar checker for the low-resourced Filipino language," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2017, doi: 10.1109/TENCON.2017.8227910.

[19] S. F. Adafre, "Part of speech tagging for Amharic using conditional random fields," 2005, doi: 10.3115/1621787.1621797.

[20] A. Ekbal, R. Haque, and S. Bandyopadhyay, "Bengali Part of Speech Tagging using Conditional Random Field," pp. 131–136, 2007.

[21] C. Sutton and A. McCallum, "An Introduction to Conditional Random Fields," *Found. Trends® Mach. Learn.*, vol. 4, no. 4, pp. 267–373, 2012, doi: DOI:10.1561/2200000013.

[22] P. Joshi, "What Are Conditional Random Fields," 2013. .

[23] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *ICML '01 Proc. Eighteenth Int. Conf. Mach. Learn.*, 2001, doi: 10.1038/nprot.2006.61.

[24] N. Nikitinsky, "Conditional Random Fields (CRF): Short Survey," 2016. .