N-gram based Language Identification and Rule-based Grammar Checking

Nathaniel Oco, Leif Romeritch Syliongka, Joel Ilao, and Rachel Edita Roxas
De La Salle University
2401 Taft Avenue
1004 Manila, Philippines
+6325360278
{nathan.oco,leif.syliongka,joel.ilao,rachel.roxas}@delasalle.ph

ABSTRACT
Literature suggests that data pre-processing contributes to accuracy. This is true for natural language processing technologies as accuracy is partly dependent on the training data; if the training data contains documents written in a language not part of the domain or has documents with style and grammar errors, it could affect the accuracy of the system. In this paper, we present a system for data pre-processing. It uses (1) Apache Tika's n-gram based language identification to filter documents not part of the domain and (2) LanguageTool's rule-based engine to correct basic style and grammar errors. The required linguistic resources were developed and these are discussed in detail in the paper. As testbed, a 1,000-sentence corpus from the Tagalog Wikipedia was used. An expert validated a 100-sentence subset of the system output and these are the results: the system properly identified Filipino sentences with 89% accuracy, and was able to detect style and errors with 90% accuracy. Analyses reveal that false negatives for document filtering were caused by the presence of proper nouns. This could be addressed by removing them in the input text or automatically through the introduction of named entity recognition. False negatives for style and grammar checking on the other hand are not within the scope of the current rule file. Future work includes the development of an error corpus and learning the rules through statistics.

Categories and Subject Descriptors
[**Document Management and Text Processing**]: Document Management – *Text Editing*; Document Preparation – *Markup Languages*.

[**Artificial Intelligence**]: Natural Language Processing – *Language Resources*.

[**Information Retrieval**]: Retrieval Models and Ranking – *Language Models*.

[**Formal Language and Automata Theory**]: Grammars and Context-Free Languages.

General Terms
Algorithms, Measurement, Languages.

Keywords
N-gram based Language Identification, Category Profiles, Document Profile, Rule-based Grammar checking, Tagger Dictionary, XML Rules.

1. INTRODUCTION
With the quintillion bytes of data generated daily [1], data can be considered as an oil well of information. The vast Internet alone offers a wide range of publicly available documents. These include Wikipedia articles[1], RSS feeds from news websites[2] (e.g. Rappler, Manila Bulletin, Pilipino Star Ngayon), and status updates from news organizations[3] (@cnnbrk, @BBCBreaking). Current natural language processing technologies utilize these documents – either as training data or as input data – in automated processes such as part-of-speech (or POS) tagging [2], automatic corpus building [3], and sentiment analysis. However, accuracy is partly dependent on the training data and the input data; if the training data has documents not part of the domain, or has documents with style and grammar errors, it could affect the accuracy of the system. For instance, modern word forms cannot be tagged using historical word forms as training data [2]. This is also true for other vertical domains. As an example, image processing technologies need to pre–process the training data in order to achieve higher accuracy rates [4].

In this paper, we present a system that can be used for data pre-processing. The system uses (1) Apache Tika's n-gram based language identification to filter documents not part of the domain and (2) LanguageTool's rule-based engine to correct basic style and grammar errors.

The paper is structured as follows: we discuss LID and grammar checking in section 2; Apache Tika's language identification in section 3; LanguageTool's rule-based engine and the different style and grammar errors considered for this paper in section 0; testing and results in section 5; and we conclude our work in section 6.

2. LANGUAGE IDENTIFICATION AND GRAMMAR CHECKING
Language identification (or LID) is the process of determining which language a text input is in [5]. The manual approach normally involves checking each word in the text input against

[1] Wikipedia XML files are available online for download: http://dumps.wikimedia.org/

[2] News articles can be downloaded using an e-book reader. One example of such is Calibre: http://calibre-ebook.com/

[3] Tweets can be downloaded using Twitter 4J: http://twitter4j.org/en/index.html

several dictionaries. The dictionary that has the most number of word matches would equate to the identified language. Even if this is automated [6], the accuracy is low, which is in the range of 75.22% to 76.26%. Supervised learning approaches [7] on the other hand are computationally expensive. A large number of training data equates to maintaining a large number of terms [8]. Even with the aid of computer tools, training is still time-consuming. These word-based approaches are either low in accuracy or are computationally expensive.

Meanwhile, grammar checking according to literature [9], [10] is the process of (1) determining if there is error in a document, (2) locating where the error is, (3) determining the cause of error, (4) notifying the user about the error, and (5) providing suggestions on how to address the error. Early approaches [11], [12] in grammar checking relied on formal grammars. An error is detected if parsing fails. This approach requires a large grammar file to properly work; if the sentence is not covered by the grammar file, it will be marked as with error. This weakness has been augmented in recent studies [13] by supplying hand-crafted rules to handle certain sentence patterns with errors.

In this paper therefore, we seek to address the following research questions: (1) *how can computers be used to filter documents without using word-based approaches* and (2) *how can computers be used to check documents for style and grammar errors without relying on formal grammars*.

3. N-GRAM BASED LID

As mentioned, the use of dictionaries often equates to low accuracy while supervised learning are computationally expensive. Due to these problems faced by word-based approaches, a number of studies [3], [5], [6], [14] have utilized character n-grams to perform language identification. N-grams are n-character slices of a word [14]. As an example, the following n-grams can be generated from the word "word":

- 1-gram: {w,o,r,d}
- 2-gram: {wo,or,rd}
- 3-gram: {wor,ord}
- 4-gram: {word}

N-gram based approaches can be mathematically described using the equation in (1). It normally involves:

- modeling the domain languages and the input into smaller representations called n-gram profiles – the n-gram profiles of the domain languages are called category profiles (*CP*) and the n-gram profile of the input is called document profile (*DP*);
- computing the similarity between the document profile and the category profiles (*S(DP,CP)*);
- the category profile that would yield the highest score is the identified language.

$$\hat{L} = \arg\max_{CP \in \Gamma} S(DP, CP) \quad (1)$$

3.1 N-gram Profiles

An n-gram profile is typically composed of two things: (1) n-grams of different sizes and (2) the number of times they appeared in the training data. Based on literature [15], 3-grams offer a good combination of accuracy and computation. An example n-gram profile composed of 3-grams is shown in Table 1. An underscore ('_') is added at the start or end of a word during training. This is to differentiate a 3-gram that is found at the start, at the middle, or at the end of a word. The first 3-gram in the table indicates that there are 7M words that end in "ng".

Table 1. An example category profile of the Filipino language showing the top 10 3-grams

3-gram	Frequency
ng_	7,675,177
ang	4,575,086
_na	3,083,907
_sa	2,643,366
sa_	2,388,606
_an	2,027,396
_ma	1,975,713
_ng	1,842,424
_pa	1,811,321
an_	1,803,874

There are computer programs that can automatically generate n-gram profiles. For this paper, Apache Nutch[4] was used to generate the top 1,000 trigrams. It already has n-gram profiles for several languages, shown in Table 2, which were also used for this paper. The two-letter digit codes were taken from ISO standards[5].

Table 2. List of languages with n-gram profiles

Code	Language
da	Danish
de	German
en	English
es	Spanish
fr	French
it	Italian
nl	Dutch
pl	Polish
ru	Russian
sv	Swedish

Currently, Apache Nutch does not have an n-gram profile for Filipino (ISO language code: tl) so we generated several for this study. We would also like to know how the size of training data affects accuracy so different corpora were used. These are shown in Table 3: (1) religious and literary texts from the PALITO corpus [16], [17]; (2) general knowledge articles from the Tagalog Wikipedia [18]; and (3) the University of the Philippines Digital Signal Processing corpus [19].

Table 3. Sources of training data and the number of words

Source	Number of words
PALITO Corpus	290K
Tagalog Wikipedia	2M
UP DSP Corpus	31M

[4] Apache Nutch can be downloaded from this website: http://nutch.apache.org/

[5] Codes for the representation of names of languages: http://www.loc.gov/standards/iso639-2/php/code_list.php

3.2 Sum of Differences of Probabilities

There are computer programs that can automatically compute for the similarity between the category profiles and the document profile. For this paper, we used Apache Tika's LID module[6] to perform language identification. Its similarity measure is heavily based on the out-of-place measure [14] and takes into consideration the frequency count. We modified the similarity measure by considering instead the probabilities of the 3-grams in the category profiles and the frequency count of the 3-grams in the document profile. This is shown in equation (2) and an example is given in Figure 1. If a 3-gram present in the document profile is also present in the category profile, the difference between the probability of the 3-gram of the category profile ($P(T|CP)$) multiplied by 10^4 and the frequency count of the 3-gram in the document profile ($F(T|DP)$) is taken and divided by two. Else, the frequency count of the 3-gram in the document profile is taken.

$$S(DP,CP)=\sum_{i=1}^{n}\frac{|P(T|CP)*10^4-F(T|DP)|}{2} \quad (2)$$

Category Profile		Document Profile		Difference / 2
3-gram	P*10^4	3-gram	F	
[a]ng_	259,000	[a]ng_	26	129,487
[b]ang	150,200	[b]ang	14	75,093
_na	119,100	[c]_sa	8	36,046
[c]_sa	72,100	[d]_ma	4	8,248
sa_	42,800	ga_	2	2
_an	25,600	_ag	2	2
[d]_ma	16,500	_ga	2	2
_ng	14,900	_mg	2	2
_pa	14,400	_ta	2	2
an_	14,700	tn_	2	2
		Sum		248,886

Figure 1. Calculating the similarity measure between the category profile and the document profile

4. RULE-BASED GRAMMAR CHECKING

Recent works on grammar checking [20], [21] have focused on capturing sentences with errors instead of correct sentences. This is through pattern matching with the aid of rule-based engines. One example of a rule-based engine is LanguageTool[7]. LanguageTool uses two resources to work: (1) the tagger dictionary and the (2) rule file. An example screenshot of LanguageTool is shown in Figure 2. It can run as an OpenOffice and LibreOffice extension or as a stand-alone program.

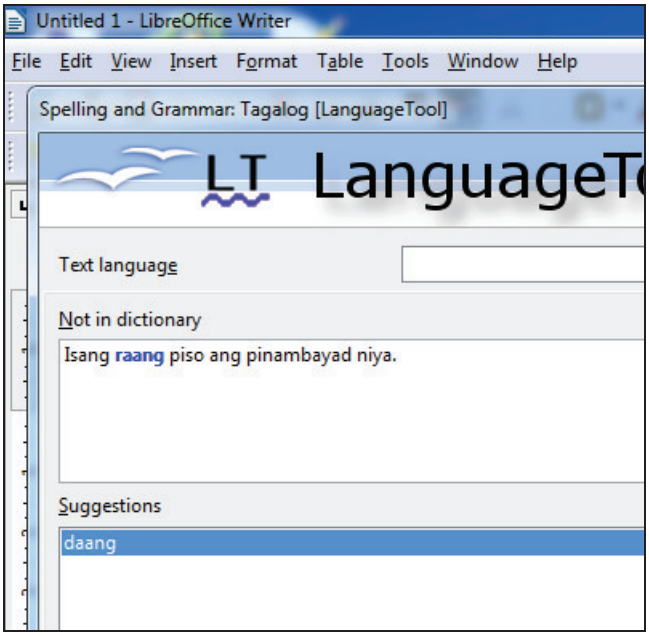

Figure 2. Example screenshot of LanguageTool in LibreOffice

The tagger dictionary is a text file that contains word declarations and their tag. Some examples are shown in Table 4. NPRO is proper noun, NCOM is common noun, and INTR is interjection. The tagger dictionary follows a three-column format where the first column is the token, the second column is the base form of the token, and the third column is the POS tag of the token.

Table 4. Example word declarations

Token	Base form	Tag
Agustino	Agustino	NPRO
ah	ah	INTR
aha	aha	INTR
ahedres	ahedres	NCOM 2
abogadong	abogado	NCOM 1

The rules on the other hand are stored in an XML file contains the patterns to be matched. These patterns could be represented in terms of tokens, regular expressions, and/or POS tags.

LanguageTool checks documents as follows:

- it separates an input into sentences and separates each sentence into tokens;
- tokens are given their tag using the declarations in the tagger dictionary;
- the tokens, together with their tag, are matched against the rule file;
- if a pattern matches, the user is notified and feedback with possible linguistic explanation or suggestion is provided.

Currently, LanguageTool supports English and other languages including Filipino. Foundations for the Filipino component of LanguageTool have been established in related literatures [9], [10]:

[6] Apache Tika can be downloaded from this website: http://tika.apache.org/

[7] LanguageTool is available online and can be downloaded as an OpenOffice and LibreOffice extension or as a stand-alone program: https://www.languagetool.org/

- tagset for Filipino, patterned after existing studies on Tagalog part-of-speech (POS) tagging [22], [23];
- classification of errors into three: wrong word, missing word, and transposition of words;
- basic rules such as proper ligature usage;
- error representation through regular expressions.

We present in the succeeding texts some of the different style and grammar rules considered for this paper, and the changes we made to both the tagger dictionary and the rule file to represent these errors.

4.1 Autocorrect

Through experience, one of the problems when using word editors is the autocorrect function. Two examples are shown in Table 5. To capture these, the tokens were declared in the tagger dictionary with the correct form as base form. A rule was then added declaring the POS tag of these words as patterns to be matched and the base form as suggestion.

Table 5. Examples of words with autocorrect

Token	Correct
lagging	laging
nagging	naging

4.2 Verb Affix Usage

According to a linguistics study [24], certain affixes cannot be used on all verbs. This reaffirms earlier studies that affixes can contain semantic information [10], [11]. Examples are shown in Table 6. The tokens and the correct forms were added in the rule file using regular expressions in both the pattern to be matched and the suggestion. An example is shown in Table 7. The affix "nag" should be changed to the infix "um".

Table 6. Examples of agent affix usage

Case description	Token	Correct
Verbs take the infix -um- to focus the agent	nagbanggit nagbunot	bumanggit bumunot
Verbs used for non-volitional or inanimate objects	nag-agos nag-andar	umagos umandar
Verbs that can take a volitional agent or a non-volitional agent	nagbagal nagbilis naggulong	bumagal bumilis gumulong
Verbs take the prefix mag- to focus the agent	umabang umahit	nag-abang nag-ahit

Table 7. Example regular expressions to handle verb affix usage

Pattern to be matched	Suggestion
nag(bisita|buhay|bulong|bunot)	regexp_match="nag(.)(.*)" regexp_replace="$1um$2"

4.3 Intervocalic Tapping

Certain words in Filipino undergo sound and letter changes when preceded by a vowel or a consonant [25], [26]. Examples are shown in Table 8. If the preceding word ends with a vowel or with 'w' or 'y', the word "daan" (trans. hundred) should start with the letter 'r'. Using also regular expressions in both the pattern to be matched and the suggestion, previous tokens were taken into consideration. An example is shown in Table 9. The word "raan" or "raang" should be changed to "daan" or "daang".

Table 8. Examples of d/r letter change

Previous token(s)	Token	Correct
Anim na	daan	raan
Tatlong	raan	daan

Table 9. Example regular expressions to handle sound and letter changes

Pattern to be matched	Suggestion
isang|dalawang|tatlong|limang|pitong raang?	daang?

4.4 Plurality

A number of studies [27], [28] have noted plurality on certain verb forms. Examples are shown in Table 10. Plurality was added to verb attributes and verbs with plural forms were added to the tagger dictionary.

Table 10. Examples of plurality usage

Affix	Token	Translation
Mangagsipag- -um-	Mangagsipag-umiyak	Cry
Mangagsi-	Mangagsisama	Accompany

4.5 Spelling Variations

We also want to address spelling variations. One study [19] tracked linguistic change in the past decades and came up with a list of spelling variants. Examples of variant pairs are shown in Table 11. One school of thought would say that the shorter the better, another would say that the longer should be observed. Given this kind of conflict, which is the correct one?

Table 11. Examples of variant pairs

Case description	Example
/w/ vs. /uw/	kwento / kuwento
/y/ vs. /iy/	superstisiyon / superstisyon

For this paper, we propose the use of statistical data to address this conflict. This involves scoring the word pairs based on the frequency count in percent (*PF*) and average frequency of the 3-grams (*AF*). The formula for the weighted score is shown in equation (3). The variant with the higher weighted score (*WS*) is included in the suggestion. As an example, weighted score for the /w/ vs. /uw/ variant pair is shown in Table 12. It can be noticed that the /uw/ has a higher weighted score.

$$WS = PF * .5 + AF * .5 \quad (3)$$

Table 12. Weighted score of variant pairs

Variant 1	Weighted score	Variant 2	Weighted score
sekswalidad	0.39	seksuwalidad	0.61
samakatwid	0.05	samakatuwid	0.95
kwentista	0.19	kuwentista	0.81
itinatwa	0.25	itinatuwa	0.75
katwiran	0.17	katuwiran	0.83
eskwela	0.27	eskuwela	0.73
pwedeng	0.23	puwedeng	0.77
kwarto	0.32	kuwarto	0.68
kwento	0.23	kuwento	0.77
pwersa	0.17	puwersa	0.83
pwesto	0.15	puwesto	0.85
pwede	0.21	puwede	0.79

5. EVALUATION AND RESULTS

To test the system, a 1,000-sentence corpus from the Tagalog Wikipedia was used. This is separate and distinct from the training data used. Because of the nature of Wikipedia articles – being prone to error themselves with some articles written in English[8] – the training data and the results of the system had to be evaluated. We asked an expert to validate a subset of 100 sentences of the system output.

5.1 Data Filtering

Results for data filtering are shown in Table 13. The system properly identified Filipino sentences with an accuracy rate of 83% to 89%. A slight increase in accuracy was noted with increase in the size of the training data. It can also be noted that a 290K-word corpus is sufficient as training data for LID systems.

Table 13. LID test results

Language	Expert Evaluation	Corpus		
		PALITO	Tagalog Wiki	UP DSP
Filipino	100	83	85	89
Other	0	17	15	11

Analyses of the false negatives reveal that they contain proper nouns. Some of the sentences are as follows:

- May mga problema rin sina Junior sa kanyang kasintahan na si Evangeline (Koronel), na humimbang sa kanyang abay sa Santacruzan.
- Ang Asociacion de las Damas Cristianas ay naalingasngas nang matuklasan na si Kuala ay buntis.
- Nagpakita si Junior at tinulungan ang buntis na Kuala sa paraan na maibalik sa barungbarong ni Berto.

This could be addressed by removing proper nouns or automatically through the use of named entity recognition approaches.

5.2 Style and Grammar Checking

Among the 1,000 sentences, the system detected 388 errors. Results for style and grammar checking are shown in Table 14. The system properly flagged 40 out of 49 sentences as erroneous and 50 out of 51 sentences as correct. Overall, it has a 97.56% precision, 81.63% recall, and 90% accuracy.

Table 14. Grammar checking test results

Sentence type	Expert	Correctly flagged	Not flagged
Correct	51	50	1
Erroneous	49	40	9
Total	100	90	10

Analyses of the false positive and false negatives reveal one weakness with rule-based approaches – its dependence on rules; if there are fewer rules, fewer errors would be detected. The false negatives are not are not within the scope of the rule file, that is why they were not detected. Examples are shown in Table 15. The false positive is the word "Botswana", which refers to a country in Southern Africa. The system corrected is as "Botsuwana" because of the rule referring to spelling variations. This can be addressed by declaring proper nouns as exceptions to the rule.

Table 15. Example tokens with errors flagged as correct

Token(s)	Correct Form
Nina Alexander	Nila Alexander
Linalarawan	Inilalarawan

6. CONCLUSION

We have presented in this paper a system that can be used for data preparation. We used Apache Tika to perform LID and LanguageTool to perform grammar checking. We have shown that n-gram based language identification can be used to filter documents not part of the domain and we have also shown that rule-based grammar checking can be used to correct errors. Test results show that the system cannot properly identify the language of text input with a lot of proper nouns and the system was not able to detect errors not declared in the rule file. Future work includes the development of an error corpus and automatically developing the rules through statistics.

7. ACKNOWLEDGMENTS

This project is supported in part by the University Research and Coordination Office of De La Salle University (project no. 09 IR U 2TAY12-2TAY13). The authors would like to thank Dr. Raquel Sison-Buban for being instrumental to the completion of this paper.

[8] Sample Wikipedia article (accessed January 12, 2014): http://tl.wikipedia.org/wiki/Panahon_ng_bagyo_sa_hilagang-kanlurang_Pasipiko_ng_2011

8. REFERENCES

[1] Krishnan, N. Recent trends in Big Data analysis. In *Proceedings of the 3rd International Conference on Recent Trends in Information Technology*, (Chennai, India, 2013), IEEE Xplore.

[2] Bollman, M. POS tagging for historical texts with sparse training data. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, (Sofia, Bulgaria, July 25 – 27, 2013), Association for Computational Linguistics, 11-18.

[3] Dimalen, D.M. and Roxas, R.E. Autocor: A query based automatic acquisition of corpora of closely-related languages. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*, (Seoul, South Korea, November 1 – 3, 2007), Korean Society for Language and Information, 146-154.

[4] Araw, D., Duguran, J.C., Martinez, M., Pangilinan, R.M., and Cordel, M. An approach to discriminate hand dorsal vein patterns for low-cost biometric identification. In *Proceedings of the 12th Philippine Computing Science Congress*, (Quezon City, Philippines, March 1 – 3, 2012), De La Salle University.

[5] Oco, N., Ilao, J., Roxas, R.E., and Syliongka, L.R. Measuring language similarity using trigrams: Limitations of language identification. In *Proceedings of the 3rd International Conference on Recent Trends in Information Technology*, (Chennai, India, July 25 – 27, 2013), IEEE Xplore.

[6] Yeong, Y.-L. and Tan, T.-P. Language identification of Malay-English words using syllable structure information. In *Proceedings of the 2nd International Workshop on Spoken Languages Technologies for Under-resourced Languages*, (Penang, Malaysia, May 3 – 5, 2010), University Sains, Malaysia, 142-145.

[7] Nidhi, V.G. Domain based classification of Punjabi text documents. In *Proceedings of the 24th International Conference on Computational Linguistics*, (Mumbai, India December 8 – 15, 2012), Indian Institute of Technology Bombay, 297-304.

[8] Tsay, J.-J. and Wang, J.-D. Design and evaluation of approaches to automatic Chinese text categorization. *Computational Ling. Chinese Lang. Process.* 5, 2 (Aug. 2000), 43-58.

[9] Oco, N. and Borra, A. Tagalog support for LanguageTool. In *Proceedings of the 8th National Natural Language Processing Research Symposium*, (Manila, Philippines, November 24 – 25, 2011), De La Salle University, 2-9.

[10] Oco, N. and Borra, A. A grammar checker for Tagalog using LanguageTool. In *Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011*, (Chiang Mai, Thailand, November 12 – 13, 2011), Asian Federation of Natural Language Processing, 2-9.

[11] Jasa, M.A., Palisoc, J.M., and Villa, M. 2007. *Panuring Pampanitikan (PanPam): A Sentence Syntax and Semantic Based Grammar Checker for Filipino*. Undergraduate Thesis. De La Salle University.

[12] Dimalen, D. and Dimalen, E. An OpenOffice spelling and grammar checker add-in using an open source external engine as resource manager and parser. In *Proceedings of the 4th National Natural Language Processing Research Symposium*, (Manila, Philippines, June 14 – 16, 2007), De La Salle University, 69-73.

[13] Crysmann, B., Bertomeu, N., Adolphs, P., Flickinger, D., and Klüwer, T. Hybrid processing for grammar and style checking. In *Proceedings of the 22nd International Conference on Computational Linguistics*, (Machester, UK, August 18 – 22, 2008), 153-160.

[14] Cavnar, W. and J. Trenkle. N-gram-based text categorization. *In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, (Las Vegas, NV, April 11 – 13, 1994), 161-175.

[15] Oco, N., Syliongka, L.R., Ilao, J., and Roxas, R.E. Dice's coefficient on trigram profiles as metric for language similarity. In *Proceedings of the 16th Oriental COCOSDA*, (Gurgaon, India, November 25 – 27, 2013), IEEE Xplore.

[16] Dita, S., Roxas, R.E., and Inventado, P. Building Online Corpora of Philippine Languages. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, (Hong Kong, China, December 3 – 5, 2009), City University of Hong Kong Press, 646-53.

[17] Dita, S. and Roxas, R.E. Philippine languages online corpora: Status, issues, and prospects. In *Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011*, (Chiang Mai, Thailand, November 12 – 13, 2011), Asian Federation of Natural Language Processing, 59-62.

[18] Oco, N. and Roxas, R.E. Pattern Matching Refinements to Dictionary-based Code-switching Point Detection. In *Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation*, (Bali, Indonesia, November 7 – 10, 2012), Faculty of Computer Science, 229-36.

[19] Ilao, J, Guevara, R.C., Llenaresas, V., Narvaez, E.A., and Peregrino, J. Bantay-wika: Towards a better understanding of the dynamics of Filipino culture and linguistic change. In *Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011*, (Chiang Mai, Thailand, November 12 – 13, 2011), Asian Federation of Natural Language Processing, 10-17.

[20] Milkowski, M. Developing an open-source, rule-based proofreading tool. *Software: Practice and Experience* 40, 7 (April 2010), 543-66.

[21] Jiang, Y., Wang, T., Lin, T., Wang, F., Cheng, W., Liu, X. Wang, C., Zhang, W. A rule based Chinese spelling and grammar detection system utility. In *Proceedings of 2012 International Conference on System Science and Engineering*, (Dalian, China, June 30 – July 2, 2012), IEEE Computer Society, 437-40.

[22] Rabo, V. and Cheng, C. A template-based part-of-speech tagger for Tagalog. *J. Research Sci. Computing Eng.* 3, 1 (2006).

[23] Miguel, D. and Roxas, R.E. Comparative evaluation of Tagalog part-of-speech (POS) taggers. In *Proceedings of the 4th National Natural Language Processing Research Symposium*, (Manila, Philippines, June 14 – 16, 2007), De La Salle University, 74-77.

[24] Endriga, D. Refining the Agent. In *Proceedings of the 11th Philippine Linguistic Congress*, (Quezon City, Philippines, December 7 – 9, 2011), University of the Philippines.

[25] Zuraw, K. Using the web as a phonological corpus: A case study from tagalog. In *Proceedings of the 2nd International Workshop on Web as Corpus*, (Trento, Italy, April 3, 2006), 59-66.

[26] Sentro ng Wikang Filipino – Diliman. 2008. *Gabay sa Editing sa Wikang Filipino*. University of the Philippines, Quezon City.

[27] Kroeger, P. 1993. *Phrase Structure and Grammatical Relations in Tagalog*. CSLI Publications, Stanford, CA.

[28] Schachter, P. and Otanes, F. 1972. *Tagalog Reference Grammar*. University of California Press, Berkeley, CA.