A Grammar Checker for Tagalog using LanguageTool

Nathaniel Oco
Center for Language Technologies
College of Computer Studies
De La Salle University
2401 Taft Avenue
Malate, Manila City
1004 Metro Manila
Philippines
nathanoco@yahoo.com

Allan Borra
Center for Language Technologies
College of Computer Studies
De La Salle University
2401 Taft Avenue
Malate, Manila City
1004 Metro Manila
Philippines
borgz.borra@delasalle.ph

Abstract

This document outlines the use of Language Tool for a Tagalog Grammar Checker. Language Tool is an open-source rule-based engine that offers grammar and style checking functionalities. The details of the various linguistic resource requirements of Language Tool for the Tagalog language are outlined and discussed. These are the tagger dictionary and the rule file that use the notation of Language Tool. The expressive power of Language Tool's notation is analyzed and checked if Tagalog linguistic phenomena are captured or not. The system was tested using a collection of sentences and these are the results: 91% precision rate, 51% recall rate, 83% accuracy rate.

1 Credits

LanguageTool was developed by Naber (2003). It can run as a stand-alone program and as an extension for OpenOffice.Org[1] and LibreOffice[2]. LanguageTool is distributed through LanguageTool's website: http://www.languagetool.org/.

2 Introduction

LanguageTool is an open-source style and grammar checker that follows a manual-based rule-creation approach.

LanguageTool utilizes rules stored in an xml file to analyze and check text input. The text input is separated into sentences, each sentence is separated into words, and each word is assigned a part-of-speech tag based on the declarations in the Tagger Dictionary. The words and their part-of-speech are used to check for patterns that match those declared in the rule file. If there is a pattern match, an error message is shown to the user. Currently, LanguageTool supports Belarusian, Catalan, Danish, Dutch, English, Esperanto, French, Galician, Icelandic, Italian, Lithuanian, Malayalam, Polish, Romanian, Russian, Slovak, Slovenian, Spanish, Swedish, and Ukrainian to a certain degree.

Tagalog is the basis for the Filipino language, the official language of the Philippines. According to a data collected by Cheng et al. (2009), there are 22,000,000 native speakers of Tagalog. This makes it the highest in the country, followed by Cebuano with 20,000,000 native speakers. Tagalog is very rich in morphology, Ramos (1971) stated that Tagalog words are normally composed of root words and affixes. Dimalen and Dimalen (2007) described Tagalog as a language with "high degree of inflection".

Jasa et al. (2007) stated that the number of available Tagalog grammar checkers is limited. Tagalog is a very rich language and LanguageTool is a flexible language. The development of Tagalog support for LanguageTool provides a readily-available Tagalog grammar checker that can be easily updated.

3 Related Works

Ang et al. (2002) developed a semantic analyzer that has the capability to check semantic relationships in a Tagalog sentence. Jasa et al. (2007) and Dimalen and Dimalen (2007) both developed syntax-based Filipino grammar checker extensions for OpenOffice.Org Writer. In syntax-based grammar checkers, error-checking is based on the parser. An input is considered correct if

[1] OpenOffice.Org is available at http://www.openoffice.org/
[2] LibreOffice is available at http://www.libreoffice.org/

Proceedings of the 9th Workshop on Asian Language Resources, pages 2–9,
Chiang Mai, Thailand, November 12 and 13, 2011.

parsing succeeds, erroneous if parsing fails. Naber (2003) explained that syntax-based grammar checkers need a complete grammar to function. Erroneous sentences that are not covered by the grammar can be flagged as error-free input.

4 LanguageTool Resources

Discussed here are the different language resources required by the tool. The notations, formats, and acquisition of resources are outlined and discussed.

4.1 Tagger Dictionary

Language Tool utilizes a dictionary file, called the Tagger Dictionary. The tagger dictionary, which contains word declarations, is utilized in pattern matching to identify and tag words with their part-of-speech.

The tagger dictionary can be a txt file, a dict file, or an FSA-encoded[3] dict file. The tagger dictionary contains three columns, separated by a tag. The first column is the inflected form. The second column is the base form. The third column is the part-of-speech tag. The format for the Tagalog tagger dictionary follows the three-column format. The first column is the inflected form, which could contain ligatures. The second column is similar to the first column, except that ligatures were omitted. This serves as the base form. The third column is the proposed tag, which is composed of the part-of-speech or POS of the word and the corresponding attribute-value pair, separated by a white space character. This serves as the POS tag. Figure 1 shows a sample declaration from the Tagalog tagger dictionary.

```
doktor  doktor  NCOM
ako     ako     PANP ST S
kumakain        kumakain        VACF IN
nasa    nasa    PRLO
mga     mga     DECP
hoy     hoy     INTR
```

Figure 1. Tagalog Tagger Dictionary Example Declarations

Evaluation and test data from different researches on Tagalog POS Tagging (Bonus, 2004; Cheng and Rabo, 2006; Miguel and Roxas, 2007) were used to come up with almost 8,000 word declarations for the Tagalog Tagger Dictionary.

4.2 Tagset for the Tagger Dictionary

A tagset for the Tagalog tagger dictionary is proposed. The tagset is based on the tagset developed by Rabo and Cheng (2006) and the modifications by Miguel and Roxas (2007). The discussions on Tagalog affixation (1971) and case system of Tagalog verbs (1973) by Ramos, verb aspect and verb focus by Cena and Ramos (1990), different Tagalog part-of-speech by Cubar and Cubar (1994), and inventory of verbal affixes by Otanes and Schachter (1972) were taken into account.

Table 1 shows the proposed noun tags. Nouns were classified into proper nouns, common nouns, and abbreviations. Kroeger (1993) explained that the determiners used for proper nouns and common nouns are different to a certain degree.

NOUN: [tag] [semantic class]	
Tag	
NPRO	Proper Noun
NCOM	Common Noun
NABB	Abbreviation

Table 1. Noun Tags

Table 2 shows the proposed pronoun tags. Grammatical person and plurality attribute were added to aid in distinguishing different types of pronouns.

PRONOUN: [tag] [grammatical person] [plurality]	
Tag	
PANP	"ang" Pronouns
PNGP	"ng" Pronouns
PSAP	"sa" Pronouns
PAND	"ang" Demonstratives
PNGD	"ng" Demonstratives
PSAD	"sa" Demonstratives
PFOP	Found Pronouns
PINP	Interrogative Pronouns
PCOP	Comparison Pronouns
PIDP	Indefinite Pronouns
POTH	Other
Grammatical Person	
ST	1st person
ND	2nd person
RD	3rd person
NU	Null

[3] FSA stands for Finite State Automata. Morfologik was used to build the binary automata. Morfologik is available at http://sourceforge.net/projects/morfologik/files/morfologik-stemming/

Plurality	
S	Singular
P	Plural
B	Both

Table 2. Pronoun Tags

Table 3 shows the proposed verb tags. Verb focus and verb aspect were added. The verb focus can indicate the thematic role the subject is taking. This is useful for future works.

VERB: [focus] [aspect]	
Focus	
VACF	Actor Focus
VOBF	Object / Goal Focus
VBEF	Benefactive Focus
VLOF	Locative Focus
VINF	Instrument Focus
VOTF	Other
Aspect	
NE	Neutral
CM	Completed
IN	Incompleted
CN	Contemplated
RC	Recently Completed
OT	Other

Table 3. Verb Tags

Table 4 shows the proposed adjective tags. Plurality was added to handle number agreement. Kroeger (1993) stated that if the plurality of the nominative argument does not match the plurality of the adjective or the predicate, the sentence considered ungrammatical.

ADJECTIVE: [tag] [plurality]	
Tag	
ADMO	Modifier
ADCO	Comparative
ADSU	Superlative
ADNU	Numeral
ADUN	Unaffixated
ADOT	Other
Plurality	
S	Singular
P	Plural
N	Null

Table 4. Adjective Tags

Table 5 shows the proposed adverb tags. An additional attribute was added to distinguish the POS of the word being modified. Ramos (1971) stated that adverbs in Tagalog can modify verbs, adjectives, and other adverbs.

ADVERB: [tag] [modifies]	
Tag	
AVMA	Manner
AVNU	Numeral
AVDE	Definite
AVEO	Comparison, group I
AVET	Comparison, group II
AVCO	Comparative, group I
AVCT	Comparative, group II
AVSO	Superlative, group I
AVST	Superlative, group II
AVSC	Slight comparison
AVAY	Agree (Panang-ayon)
AVGI	Disagree (Pananggi)
AVAG	Possibility (Pang-agam)
AVPA	Frequency (Pamanahon)
AVOT	Other
Modifies	
VE	Verb
AD	Adjective
AV	Adverb
AL	Applicable to All

Table 5. Adverb Tags

Conjunctions, prepositions, determiners, interjections, ligatures, particles, enclitic, punctuation, and auxiliary words are also part of the proposed tagset. These tags however, do not contain additional properties or corresponding attribute-value pairs. Overall, the tagset has a total of 87 tags from 14 POS and lexical categories.

4.3 Rule File

The rule file is an xml file used to check errors in a sentence. If a pattern declared in the rule matches the input sentence, an error is shown to the user.

The rule file, case insensitive by default, is composed of several rule categories which may cover but is not limited to spelling, grammar, style, and punctuation errors. Each rule category is composed of one or more rules or rule groups. Each rule is composed of different elements and attributes. The three basic elements a rule has are pattern, message, and example. The pattern element is where the error to be matched is declared. The message element is where the feedback and suggestion, if applicable, is declared. The example element is where incorrect and correct examples are declared. Figure 2 shows a pseudocode that describes what happens in the event a pattern is matched and Figure 3 shows an example rule in the Tagalog rule file.

```
if(pattern in rule file = pattern in input) {
        mark error;
        show feedback;
        provide suggestions if applicable;
}
```

Figure 2. Pseudocode

```
<rule id="MGA_MGA" name="mga mga
(ang mga)">
        <pattern case_sensitive="no"
        mark_from="0">
                <token>mga</token>
                <token>mga</token>
        </pattern>
        <message>Do you mean
                <suggestion>ang
                \2</suggestion>? "mga" can
                not be followed by another
                "mga".
        </message>
        <short>Word Repetition</short>
        <example correction="ang mga"
        type="incorrect">Maganda
        <marker>mga mga</marker>
        tanawin.</example>
        <example type="correct">Maganda
        <marker>ang mga</marker>
        tanawin.</example>
</rule>
```

Figure 3. Rule File Declaration for “ang ang” word repetition

Pattern matching can utilize tokens, POS tags, and a combination of both to properly capture errors. Regular expressions[4] are also used to simplify or merge several rules. Figure 4 shows different examples of using regular expression. Different methods of pattern-matching explained in LanguageTool’s website are shown in Figure 5. It should be noted that if a particular error is not covered by the tagger dictionary and the rule file, the error will not be detected.

```
ding? = din or ding
ring? = rin or ring
.*[aeiou] = any word that ends in a vowel
.*[bcdfghjklmnpqrstvwxyz] = any word that
ends in a consonant
```

Figure 4. Regular expression usage

```
<token bla="x">think</token>
matches the word “think”

<token regexp="yes">think|say</token>
matches the regular expression think|say, i.e.
the word “think” or “say”

<token postag="VB" />
<token>house</token>
matches a base form verb followed by the
word house.

<token>cause</token> <token regexp="yes"
negate="yes">and|to</token>
matches the word “cause” followed by any
word that is not “and” or “to”

<token postag="SENT_START" /> <to-
ken>foobar</token>
matches the word “foobar” only at the begin-
ning of a sentence
```

Figure 5. Different methods of pattern-matching described in LangaugeTool’s website

The following resources were used as basis in developing rules: Makabagong Balarila ng Pilipino (Ramos, 1971), Writing Filipino Gramamar: Traditions and Trends (Cubar and Cubar, 1994), Modern Tagalog: Grammatical Explanations and Exercises for Non-native Speakers (Cena and Ramos, 1990), Tagalog Reference Grammar (Otanes and Schachter, 1972) and Phrase Structure and Grammatical Relations in Tagalog (Kroeger, 1993).

5 Tagalog Grammar Checking

Errors are classified into three types: wrong word, missing word, and transposition of words. This section discusses the different types of errors and the corresponding method for capturing these errors. Figure 6 shows a pseudocode explaining how an error is classified.

[4] Standard Regular Expression Engine of Java. Described at: http://download.oracle.com/javase/1,5.0/docs/api/java/util/regex/Pattern.html

```
if(POS sequence != unoccurring)
        Wrong Word;
else if(POS sequence = unoccurring)
        if(POS sequence before !=
        unoccurring || POS after != unoccur-
        ring)
                Missing Word;
        else
                Transposition;
```

Figure 6. Pseudocode

5.1 Wrong Words

Wrong words are often caused by using the wrong determiner and affixation rule. Also, morphophonemic change and verb focus are often not taken into consideration. There are cases where relying on part-of-speech alone will not capture certain errors. To address this issue, grammatical person and plurality of pronouns, focus and aspect of verbs, plurality of adjectives, and word modified by adverbs were considered in developing the tagset. Consider the example in Figure 7. Both have the same POS but only one is correct. Kroeger (1993) pointed out that plurality in adjectives is demonstrated by the reduplication of the first syllable. An error caused by the disagreement of the plurality of the adjective and the plurality of the nominative argument can not be handled by considering the part-of-speech only.

Correct:

Magaganda	*kami.*
Adjective	1st person Pronoun
Plural	Plural
Beautiful	we.

We are beautiful.

Incorrect:

Magaganda	*ako.*
Adjective	1st person Pronoun
Plural	Singular
Beaautiful	me.

(For: I am beautiful)

Figure 7. Number Agreement

Consider the sentences in Figure 8. The enclitic "*din*" is used if the last letter of the preceding word is a consonant. Otherwise, "*rin*" is used. Cena and Ramos (1990) explained that sound and letter changes occur in affixation and even in word boundaries. "*din*" and "*rin*" is one of many examples. To address this, a simple token matching is performed. Regular expressions were employed to make rule files shorter.

Correct:
Magnanakaw din siya.
He is also thief.

Incorrect:
Magnanakaw rin siya.
(For: He is also thief.)

Figure 8. Sound and Letter Change

Other errors like proper adverb and ligature usage also fall into this type of error.

5.2 Missing Words

Missing words are often due to missing determiners, particles, markers, and other words composed of several letters. Usually, missing words cause irregular and unoccurring POS sequence. Figure 9 illustrates an example. Unoccurring POS sequence are checked and matched against specific rules. The missing word is added to the sentence as feedback. In the sentences in Figure 9, it is unnatural for a pronoun to be immediately followed by an adjective. Missing words are captured by looking for unoccurring POS sequence often caused by a missing word.

Correct:

Ikaw	*ay*	*maganda.*
Pronoun	Marker	Adjective
You		beautiful

You are beautiful.

Incorrect:

Ikaw	*maganda.*
Pronoun	Adjective
You	beautiful

(For: You are beautiful)

Figure 9. Missing Lexical Marker "*ay*"

5.3 Transposition

The process of detecting errors caused by transposition is similar to missing words. The main difference is tokens and POS tags before and after the unoccurring POS sequence are considered and checked for any irregularities.

6 Performance of Language Tool: Results and Analysis

The system was initially tested using a collection of sentences. The collection is composed of evaluation data used in FiSSAn (Ang et al., 2002), LEFT (Chan et al., 2006), and PanPam (Jasa et al., 2007). Test data used by Dimalen (2003) examples from books (Kroeger, 1993; Ramos, 1971), and additional test data are also part of the collection. A total of 272 sentences from the collection were used. Table 6 shows a summary of figures. 186 out of 190 error-free sentences were marked as error-free, 4 out of 190 error-free sentences were marked as erroneous, 42 out of 82 erroneous sentences were marked as erroneous, and 40 out of 82 erroneous sentences were marked as error-free.

Sentences	Correctly Flagged	Incorrectly Flagged	Total
Error-free	186	4	190
Erroneous	42	40	82
Total	228	44	272

Table 6. Summary of Figures

The test showed that the system has a 91% precision rate, 51% recall rate, and 83% accuracy rate. Figure 10, Figure 11, and Figure 12 show the formulas used for precision, recall, and accuracy, respectively. True Positives refer to erroneous evaluation data properly flagged by the system as erroneous. False Positives refer to error-free evaluation data flagged by the system as erroneous. True Negatives refer to error-free evaluation data properly flagged by the system as error-free. False Negatives refer to erroneous evaluation data flagged by the system as error-free.

$$\frac{TruePositives}{TruePositives + FalsePositives}$$

Figure 10. Precision Formula

$$\frac{TruePositives}{TruePositives + FalseNegatives}$$

Figure 11. Recall Formula

$$\frac{TruePositives + TrueNegatives}{TotalNumberOfEvaluationData}$$

Figure 12. Accuracy Formula

The system flagged 4 error-free sentences as erroneous. This is mainly because of wrong declarations in the tagger dictionary file. Figure 13 shows one of the sentences. In the tagger dictionary, "*mag-aral*" was declared as a noun and "*maingay*" was declared both as an adverb and as an adjective. In the Tagalog language, if a common noun is preceded by an adjective, there should be a ligature between them. Figure 14 demonstrates proper Tagalog ligature usage.

Umalis	*ang*	*mabait*
Verb	Det	Adjective
Leave	the	good
ngunit	*maingay*	*mag-aral.*
Conjunct	Adverb	Verb
but	noisy	study

Figure 13. Flagged as erroneous

Root word ends with a vowel, add "-ng"

Matalino	+	*bata*
Adjective		Common Noun
Intelligent		Child

=*Matalinong bata*
Intelligent Child

Root word ends with the letter "n", add "-g"

Matulin	+	*bata*
Adjective		Common Noun
Fast		Child

=*Matuling bata*
Fast Child

Root word ends with a consonant, add "na"

Matapang	+	*bata*
Adjective		Common Noun
Brave		Child

=*Matapang na bata*
Brave Child

Figure 14. Ligature usage

The presence of ellipsis in one of the sentences is another reason why error-free sentences were flagged as erroneous. Ellipsis was not declared in the rule file. This resulted in two sentences being recognized as one.

The system flagged 40 out of 42 erroneous sentences as error-free. A close analysis on there errors reveal that majority of the sentences contains free-word order errors, transposition of more than 2 words, extra words. Some sentences contain errors that focus on semantic checking. Figure 15 shows 9 of these sentences. These are the type of errors that are not handled by the system and are not declared in the rule file. Future research works can focus on these areas.

Humihinga ang bangkay.
The corpse is breathing.

Nagluto ang sanggol.
The baby cooked.

Naglakad ang ahas.
The snake walked.

Kumain ang plato.
The plate ate.

Nabasag ang basong mabilis.
The fast glass shattered.

Kumain ang plato sa baso.
The plate ate at the glass.

Kumain ang aso ng plato.
The dog ate the plate.

Tumakbo ang sapatos.
The shoe ran.

Nagluto ang pusa ng pagkain.
The cat cooked food.

Figure 15. Flagged as error-free

Among the 42 erroneous sentences it correctly flagged as erroneous, the system provided the correct feedback for 41 sentences. The sentence with incorrect feedback is shown in Figure 16. The sentence, used to test free-word order, contains transposition of several words. The system detected it as a missing last word error because the determiner "*ang*" can not be the last word of a sentence.

Pinalo tatay ng makulit batang ang.

Correct Form:
Pinalo ng tatay ang batang makulit.
The father spanked the naughty child.

Figure 16. Sentence with incorrect feedback

For comparative evaluation, the same collection was tested on PanPam (Jasa et al., 2007) and these are the results: 23% precision rate, 46% recall rate, and 38% accuracy rate. Table 7 shows a summary of figures.

Sentences	Correctly Flagged	Incorrectly Flagged	Total
Error-free	68	122	190
Erroneous	38	44	82
Total	106	166	272

Table 7. PanPam Results

The comparative evaluation shows that the system scored 68% higher than PanPam in terms of precision, 5% higher in terms of recall, and 37% higher in terms of accuracy.

Overall, these findings reaffirm earlier analysis by Konchady (2009) that rule-based grammar checkers that follow a manual-based rule-creation approach tend to produce low recall rate but precision rate is above average. This is because the total number of rules isn't sufficient to cover a variety of errors. Also, because of pattern-matching, majority of the errors detected are indeed errors. It is also important to note, especially in the case of LanguageTool, that the patterns being captured are erroneous sentences and not error-free sentences. This makes rule-based grammar checkers dependent on the rules declared for error checking coverage.

LanguageTool can support the Tagalog language to a certain degree. Although developing a tagger dictionary and a rule file is a tedious task, it is necessary to create a tagger dictionary, a tagset, and rules that can handle the different Tagalog linguistic Penomena.

Acknowledgements

The authors acknowledge developers and maintainers of LanguageTool especially Daniel Naber, Dominique Pellé, and Marcin Milkowski for being instrumental to the completion of this study and to the completion of the Tagalog support for LanguageTool. The August 07, 2011 snapshot of LanguageTool would not include Tagalog support if not for their assistance. The authors also acknowledge the opinions and thoughts shared through email by Vlamir Rabo, Manu Konchady, and Mark Johnson.

References

Charibeth K. Cheng, Nathalie Rose T. Lim, and Rachel Edita O. Roxas. 2009. Philippine Language

Resources: Trends and Directions. *Proceedings of the 7th Workshop on Asian Langauge Resource (ALR7),* Singapore.

Charibeth K. Cheng and Vlamir S. Rabo. 2006. TPOST: A Template-based Part-of-Speech Tagger for Tagalog. *Journal of Research in Science, Computing, and Engineering,* Volume 3, Number 1.

Dalos D. Miguel and Rachel Edita O. Roxas. 2007. Comparative Analysis of Tagalog Part of Speech (POS) Taggers. *Proceedings of the 4th National Natural Language Processing Research Symposium (NNLPRS),* CSB Hotel, Manila. ISSN 1908-3092.

Daniel Naber. 2003. *A Rule-Based Style and Grammar Checker.* Diploma Thesis. Bielefeld University, Bielefeld.

Davis Muhajereen D. Dimalen and Editha D. Dimalen. 2007. An OpenOffice Spelling and Grammar Checker Add-in Using an Open Source External Engine as Resource Manager and Parser. *Proceedings of the 4th National Natural Language Processing Research Symposium (NNLPRS),* CSB Hotel, Manila.

Don Erick J. Bonus. 2004. A Stemming Algorithm for Tagalog Words. *Proceedings of the 4th Philippine Computing Science Congress (PSCS 2004),* University of the Philippines – Los Baños, Laguna.

Editha D. Dimalen. 2003. *A Parsing Algorithm for Constituent Structures of Tagalog*. Master's Thesis. De La Salle University, Manila.

Ernesto H. Cubar and Nelly I. Cubar. 1994. *Writing Filipino Grammar: Traditions and Trends*. New Day Publishers, Quezon City.

Erwin Andrew O. Chan, Chris Ian R. Lim, Richard Bryan S. Tan, and Marlon Cromwell N. Tong. 2006. *LEFT: Lexical Functional Grammar Based English-Filipino Translator*. Undergraduate Thesis. De La Salle University, Manila.

Fe T. Otanes and Paul Schachter. 1972. *Tagalog Reference Grammar*. University of California Press, Berkeley, CA.

LanguageTool. http://www.languagetool.org/

Manu Konchady. 2009. *Detecting Grammatical Errors in Text using a Ngram-based Ruleset*. Retrieved from: http://emustru.sourceforge.net/detecting_grammatical_errors.pdf

Michael A. Jasa, Justin O. Palisoc, and Martee M. Villa. 2007. *Panuring Pampanitikan (PanPam): A Sentence Syntax and Semantic Based Grammar Checker for Filipino*. Undergraduate Thesis. De La Salle University, Manila.

Morgan O. Ang, Sonny G. Cagalingan, Paulo Justin U. Tan, and Reagan C. Tan. 2002. *FiSSAn: Filipino Sentence Syntax and Semantic Analyzer*. Undergraduate Thesis. De La Salle University, Manila.

Paul Kroeger. 1993. *Phrase Structure and Grammartical Relations in Tagalog*. CSLI Publications, Stanford, CA.

Resty M. Cena and Teresita V. Ramos. 1990. *Modern Tagalog: Grammatical Explanations and Exercises for Non-native Speakers*. University of Hawaii Press, Honolulu, HI.

Teresita V. Ramos. 1971. *Makabagong Balarila ng Pilipino*. Rex Book Store, Manila.

Teresita V. Ramos. 1973. *The Case System of Tagalog Verbs*. Doctoral Dissertation. University of Hawaii. Honolulu, HI.