A Morphological Analyzer for Filipino Verbs*

Robert R. Roxas[a] and Gersam T. Mula[a]

[a]University of the Philippines Visayas – Cebu College
Cebu City, Philippines
rrroxas@upv.edu.ph and gersammula@gmail.com

Abstract. This paper presents a morphological analyzer that accepts Filipino verbs conjugated in different forms as inputs and analyzes them to produce the affixes used, the infinitive forms, and the tenses of the original input verbs. A prototype system was implemented and was fed with a file containing 1,050 Filipino verbs conjugated in various tenses using different types of affixes. The preliminary result showed that the accuracy rate was high in three expected outputs, i.e., tenses, infinitive forms, and affixes used.

Keywords: morphological analyzer, context-driven machine translation system, Filipino verbs

1.Introduction

Filipino is a language whose verb conjugations are very complex because it uses different affix (prefix, infix, and suffix) combinations and even duplication of syllables or words. When looking up a word in Filipino dictionary, one needs to know the infinitive form of the verb to be able to find it. This is a difficult task for someone who is not familiar with the way Filipino verbs are conjugated into any of the three tenses: Pangnagdaan (Past), Pangkasalukuyan (Present), and Panghinaharap (Future). These three are commonly known today as Aspektong Perpektibo, Aspektong Imperpektibo, and Aspektong Kontemplatibo, respectively (Dizon, 2006).

Because of the complexity of the language, automated morphological analyzers will be difficult to construct. So researches should be focused on how to capture all possible forms so that a machine translation system can produce accurate translation. One approach is to use a lexicon that stores the context-words that help determine the appropriate equivalent word(s) in the target language. The lexicon must use the infinitive forms of the verbs to facilitate the look up of verbs and use headwords for non-verbs. So with this kind of lexicon in mind, a morphological analyzer that returns the infinitive form of a verb, its tense, and its affix(es) used is necessary.

2.Review of Related Works

When someone attempts to create any machine translation system for natural languages, it is not possible to do away with morphological analyzers. One area of focus for morphological analyzers is the analysis of verbs. Verbs are usually conjugated according to tense, number, voice, and mode while nouns and adjectives are usually declined according to person, number, case, and degree. Some languages have simple forms of verb inflection or conjugation while other languages have complex forms of inflection. Japanese was originally thought to have simple verb inflection, thus it was not the central subject on Natural Language Processing

(NLP). But in recent past, it had become an important subject (Hisamitsu and Nitta, 1994). For other languages, verbs are also given much attention in research. These languages include Chinese (Kim, et. al, 2002), Japanese (Nakamura, 2007), and Korean (Hong, et. al., 2004; Jun, 2007) to name a few.

For languages that have complex ways of adding affixes and duplication of syllables or words, an excellent morphological analyzer is highly necessary. Filipino or Tagalog is one the most complex languages in the world. So several researches have been conducted in the area of morphological analysis for the Filipino language. One is the TagSA (Tagalog Stemmer Algorithm), which tackles on the extraction of the stem of any Tagalog word (Bonus, 2003). Another morphological analyzer, called TagMA (Tagalog Morphological Analyzer), is devoted to extracting the root word both for concatenative and non-concatenative formation (Fortes, 2002). A system called T2CMT (Tagalog-to-Cebuano Machine Translation) was developed (Fat, 2004) that used TagMA and TagSA for its morphological analyzer. Even before the TagMA and TagSA, there was already a morphological analyzer that was created and used in a prototype system that supported English-Filipino and Filipino-English machine translation system (Roxas, 1998).

TagMA (Fortes, 2002) produces three morphological structures (morpheme, CV, and syllabication) to represent an input verb. To do this, it needs to scan the entire word in several stages. It starts by assigning the symbols "C" for consonants or "V" for vowels to the morpheme structure character by character to get the CV structure. Then it scans the CV structure character by character to assign some codes (0-2) to a "C" or "V" to get the syllabification structure. Then the input representation is fed to the GEN function in order to produce a candidate set. It then scans the input string syllable by syllable until the last syllable is encountered. Then the result is subjected to the EVAL function, where the output of the GEN function is checked against the two lexicons (root and affixes) and subjected to some constraints to be able to get the right root of the verb.

Although TagMA was able to morphologically analyze 96% of the sample verbs accurately (Fortes, 2002), the process of analyzing an input verb is quite long and tedious. It only outputs the root, tense, and its affix for a particular input verb. It does not produce the infinitive or dictionary form of the original input verb. The infinitive form is also useful because that is what one usually uses to lookup a word in a dictionary. In fact, T2CMT (Fat, 2004) that used TagMA wrongly translated the Tagalog word "namatay" (to die) to "pinaagi" (by means of or through) in Cebuano, when in fact, the Cebuano translation should be "namatay" also. If a morphological analyzer produces the infinitive form, the translation would be correct. Our proposed system uses the morphological analyzer developed in (Roxas, 1989), which is being extended to cover more possible verb conjugations. If you pass "namatay" to our morphological analyzer, it will give the infinitive form "mamatay" and tense is past or perpektibo. If you consult a dictionary, you will find that its English translation is "to die," which is the intended meaning. Therefore, we believe that there is still a need to develop a morphological analyzer that uses the infinitive forms when checking a word in a dictionary.

3.The Proposed Morphological Analyzer

The morphological analyzer presented in this paper is just one of the components of our Context-driven Filipino-English Machine Translation System. The analyzer will be used during the translation process. This morphological analyzer examines any Filipino verb in various possible inflections and produces the affix(es), the infinitive form, and the tense of the input verb. It should be pointed out that we have no data as to how many infinitive forms are there in Tagalog. We don't know of any study trying to count the infinitive forms of a certain language. Natural languages are evolving. So it is difficult to categorically say that a particular language has that number of infinitive forms. The lexicon will have to be updated from time to time as the language evolves.

In this research, we don't bother so much on generating the root word because the entries in our lexicon will contain the infinitive or dictionary forms of the verbs and will be accessed using that form. We do, however, recognize the value of getting the root of a word, as being done in TagSA and TagMA, but extracting the root of the word is more useful in Information Retrieval System (IRS) than in a machine translation system. We also understand that the root of a word is used in some systems to get the right semantics of compound words in order to come up with a better translation. This is not necessary in our proposed Context-driven Filipino-English Machine Translation System, of which the morphological analyzer being presented here is just a subsystem. We choose the right English equivalent of a Filipino word by checking the neighboring words in a sentence and even in a paragraph. These neighboring words serve as context of the word being translated. So our lexicon must be a different one than the usual dictionary.

The affixes are important because they help in determining the structure of the sentence, the meaning of the entire sentence, etc. The subject of the sentence as well as the object will be easily recognized once the affix has been determined already. For example, if the affix used in the main verb is "UM," the subject of the sentence starts with the word "ang," "si" for personal subject, or uses the nominative case of pronoun. The object starts with the word "ng," "ni" for personal object, or uses the objective case of pronoun. This will be very helpful in the translation itself. As for the tense of the verb that will be output by the analyzer, it is also very useful for the translation process.

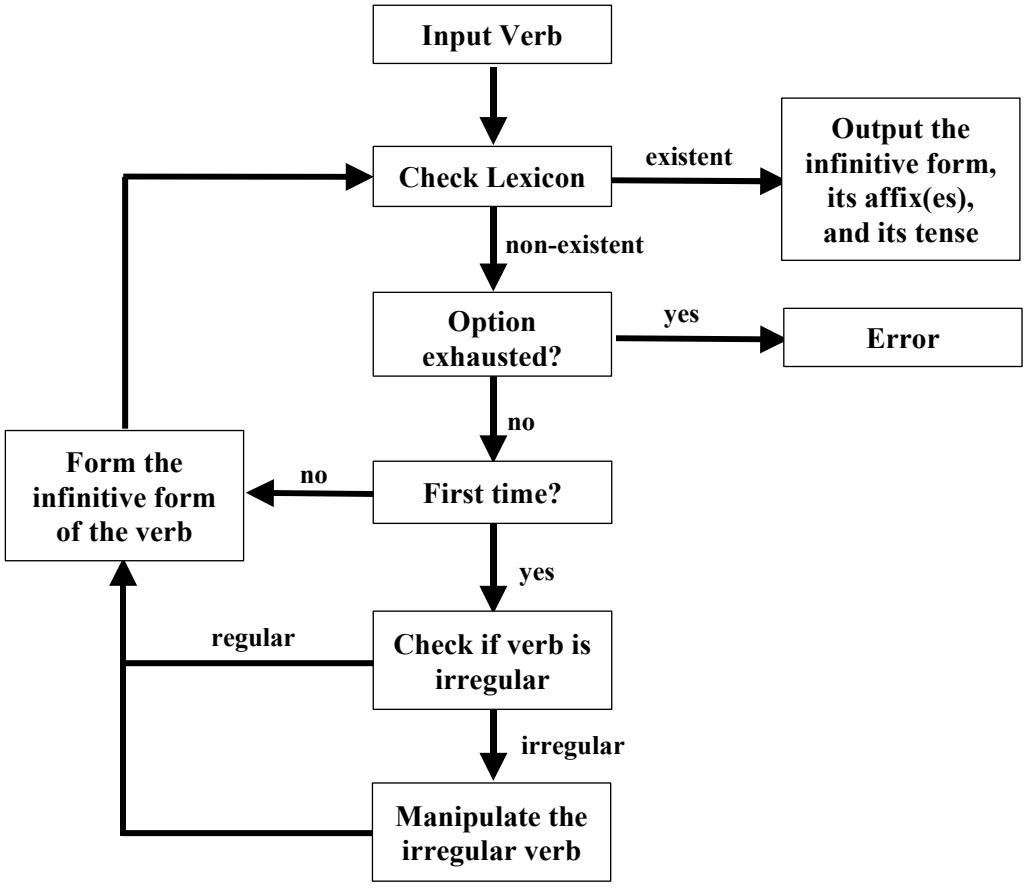

Figure 1: The flow of the morphological analyzer's activities.

4.Extracting Affixes, Infinitive Forms, and Tenses

In getting the infinitive form, the tense, and affix(es) of a Filipino verb, it first consults the lexicon if the word is there. If not, it checks whether or not all possible options were exhausted, in which case, it reports an error. If there are still options, it checks whether or not it is the first time to process the verb. If it is, it has to check whether or not the verb is irregular. If it is, it performs some manipulation and prepares for the formation of the possible infinitive form of the verb. If the verb is regular, the analyzer immediately tries to form the possible infinitive form. Figure 1 shows the flow of actions to take as the input verb is being analyzed.

```
Tense = past        /* default tense, will be changed depending on
                       the input */
If (starting with a double vowel)
  Tense = future
  Replace 1st vowel with “um”      /* iinom -> uminom (inf.) */
Else if (duplicated 1st two chars)
  Tense = future
  Replace 2nd and 3rd chars with “um”/* susulat -> sumulat (inf.)*/
Else if (1st char = 'd' & 3rd char = 'r') and (2nd char = 4th char)
  Tense = future
  Replace 2nd and 3rd chars with “um” /* darating -> dumating (inf.) */
Else if (duplicated 1st 3 letters) and (1st two chars = “ng”)
  Replace 3rd to 5th chars with “um”   /* ngingiti -> ngumiti (inf.) */
Else if (1st two chars) = “um”
  If (3rd char <> 4th char)   /* do nothing, uminom = uminom (inf.) */
  If (3rd char = 4th char)
    Tense = present
    Remove the 3rd char            /* umiinom -> uminom (inf.) */
Else if (2nd & 3rd char) = “um”
  If (1st & 4th chars) <> (5th and 6th chars)
    /* do nothing, kumain = kumain (inf.) */
  If (1st & 4th chars) = (5th and 6th chars)
    Tense = present
    Remove 5th & 6th chars        /* kumakain -> kumain (inf.) */
  If (1st char = “d” & 5th char = “r”) and (4th char = 6th char)
    Tense = present
    Remove 5th & 6th chars     /* dumarating -> dumating (inf.) */
Else if (3rd & 4th chars = “um”)
  If (1st two chars & 5th char) <> (6th to 8th chars)
    /* do nothing, ngumiti = ngumiti (inf.)*/
  If (1st 2 chars & 5th char) = (6th to 8th chars)
    Tense = present
    Remove 6th to 8th chars     /* ngumingiti -> ngumiti (inf.) */
```

Figure 2: The pseudo-code for analyzing “UM–” verbs.

The analyzer performs a certain number of passes until it finds the correct infinitive form of the input verb. If all options have been tried but still the generated infinitive is not found in the lexicon, it reports an error. For each pass, it tries to generate the infinitive form of the input verb. If the generated infinitive form does not exist in the lexicon, it is possible that the option used was not the correct one. So it tries another option. Eventually, it will generate the correct infinitive form. Then it reports the infinitive form, the tense, and the affix(es) used. Filipino verbs use one or more affixes. So it is possible that more than 1 affix is reported.

```
Tense = past        /* default tense, will be changed depending on
                       the input */
If (1st two chars = "in")
   If  (last syllable contains "o")
     Change "o" to "u"
   If (3rd char <> 4th char)
     Move "in" to the end             /* inalis -> alisin (inf.) */
   If (3rd char = 4th char)
     Tense = present
     Remove 1st 3 chars and suffix "in" /* inaalis -> alisin (inf.) */
Else if (2nd & 3rd chars = "in")
   If  (last syllable contains "o")
     Change "o" to "u"
   If (1st & 4th chars) <> (5th  & 6th chars)
     Remove 2nd & 3rd chars & suffix "in"  /* sinulat ->
                                              sulatin (inf.) */
   If (1st & 4th chars) = (5th and 6th chars)
     Tense = present
     Remove 1st 4 chars & suffix "in" /* sinusulat -> sulatin (inf.) */
   If(1st char = 5th char) & (4th char <> 6th char) &
     (4th char = 7th char)
     Tense = present
     Remove 1st 4 chars & suffix "in"  /* tinatrabaho ->
                                          trabahuhin (inf.) */
Else if (last syllable ends with "in")
   Tense = future
   If (1st 2 chars) = (3rd & 4th char)
     Remove 1st two chars             /* susulatin -> sulatin (inf.) */
   If (starting with a double vowel)
     Remove 1st char                 /* aalisin -> alisin (inf.) */
   If(1st char = 3rd char) & (2nd char <> 4th char) &
     (2nd char = 5th char)
     Remove 1st two chars       /* tatrabahuhin -> trabahuhin (inf.) */
```

Figure 3: The pseudo-code for analyzing "-IN" verbs.

It should be pointed out that matching a possible infinitive form, derived from the input verb, with one in the lexicon, is no better than generating the infinitive form from the derived root word and the affix used in the input verb. This is true if the correct root word is used. But if the system has to guess for the correct root word, it is possible that an incorrect root is used, which will make the translation fail. This is the case of the word "namatay" as reported in T2CMT (Fat, 2004).

For irregular verbs, they need to be modified a bit so that when they are passed to the section that generates the infinitive forms, it will generate the correct infinitive forms. For instance, in

getting the infinitive form of an irregular verb "**binibili,**" the analyzer consults Table 1 first. Since the verb "**binibili**" exists in the table, it changes the last "**i**" to "**h**." The verb "**binibili**" becomes "**binibilh**." This word is used to get the infinitive form "**bilhin.**" Figure 1 shows the flow of activities as the analyzer executes.

Table 1: Example of some irregular verbs and their corresponding actions to take.

Irregular Verb	Action
binili, binibili	**Change the last "i" to "h"**
dinala, dinadala	**Change the last "a" to "h"**
kinain, kinakain	**Remove the last "i"**
sinunod, sinusunod	**Remove the "o"**
dinakip, dinadakip	**Remove the last "i"**
nilunod, nilulunod	**Change "od" to "ur"**
tinrabaho, tinatrabaho	**Change "o" to "uh"**
binasa, binabasa	**Add "h"**
binago, binabago	**Change "u" and add "h"**
ginawa, ginagawa	**Remove the last "a"**

There are several verb forms or categories in Filipino: UM-, MAG-, -IN, MA-, MAGKA-, PAKI-, I-, -AN, MAGPA-, and PA-IN (Aspillera, 1981). Each category has a corresponding algorithm, which serves as an option mentioned above. When using the analyzer, the verb is passed to any of the algorithms for determining the category that the verb belongs and until the right infinitive form is found. This is done because some words seemingly belong to a certain category but actually belong to different category. If the generated infinitive form is not in the lexicon, most likely it belongs to different category. When all options have been tried and the generated infinitive form does not exist, then it must be a misspelled word or an invalid Filipino verb. Figure 2 shows the pseudo-code for analyzing "UM-" verbs (active) and Figure 3 shows the pseudo-code for analyzing "-IN" verbs (passive).

5.Results and Discussion

The prototype morphological analyzer was fed with 1,050 Filipino verbs conjugated in three different tenses taken from (Aspillera, 1981) plus some verbs used in everyday life. The verbs belong to the UM-, -IN, -AN, I-, MA-, and MAG- categories only. Table 2 shows the preliminary results. In determining the tense of the input verbs, it returned 98.76% accurately, which is very promising. The 1.14% error for determining the tense was primary caused by the irregular verb forms. The undetermined tense, which is 0.095%, was caused by the wrong input word.

Table 2: The result of the initial testing.

Expected Output	*Correct*	*Error*	*Undetermined*
Tense	1,037 (98.76%)	12 (1.14%)	1 (0.095%)
Infinitive form	1,007 (95.90%)	42 (4.00%)	1 (0.095%)
Affix	1,032 (98.28%)	17 (1.62%)	1 (0.095%)

In determining the infinitive forms of the verbs, it got a 95.90% accuracy rate, which is also good as initial results. The error rate was 4.00% and almost all of them resulted from irregular verb forms. Out of the 42 errors, 14 are caused by failing to change the "o" to "u" of the

penultimate syllable and 4 are caused by failing to insert "h" on the last syllable for vowel-ending roots without a glottal stop. The undetermined infinitive form, which is 0.095%, is purely caused by the wrong input word. As for determining the affixes of the verbs, the accuracy rate is only 98.28%. The error rate of 1.62% was caused partly by irregular verb formation but a large part of it was caused by the presence of "in" or "ni" that are always found any "I-" verbs in their past and present tenses. The undetermined affix was also caused by the wrong input word.

Many of the errors will be greatly reduced by refining the algorithm. It should be noted that the testing was just a single-pass because the lexicon is not yet fully functional. We can expect a higher accuracy rate than what is presented here, if the multi-pass approach will be used because some errors above will be solved by the multi-pass approach.

6.Conclusion and Future Works

A morphological analyzer that analyzes the Filipino verbs and produces their affixes, infinitive forms, and tenses has been presented. The prototype was tested with 1,050 verbs (regular and irregular) conjugated in three different tenses and the initial results showed high accuracy rate for determining the tense, the infinitive form, and the affix used. The higher accuracy rate is expected to even improve once the lexicon is in place and the multi-pass approach is already implemented.

Future work includes the implementation of the lexicon that employs an innovative way of getting the appropriate meaning to support context-driven machine translation system. We will also add other verb forms that were not yet included in the current prototype and perform further tests when all different forms have been taken into consideration. We will also increase the data set in the next testing.

References

Aspillera, P.S. 1981. *Basic Tagalog.* 8th revised ed. Manila: M&L Licudine Enterprises.

Bonus, D.E.J. 2003. *A Stemming Algorithm for Tagalog Words.* Master's Thesis, De La Salle University.

Dizon, C.B. 2006. *Buklod 2 – Batayang Aklat sa Filipino.* Quezon City: Book Craft Publishing Co., Inc.

Fat, J.G. 2004. *T2CMT: Tagalog-to-Cebuano Machine Translation.* Master's Thesis, De La Salle University.

Fortes, F.C.L. 2002. *A Constraint-based Morphological Analyzer for Concatenative and Non-concatenative Morphology of Tagalog Verbs.* Master's Thesis, De La Salle University.

Hisamitsu, T. and Y. Nitta. 1994. An Efficient Treatment of Japanese Verb Inflection for Morphological Analysis. *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 194-200.

Hong, M., Y.-K. Kim, S.-K. Park, and Y.-J. Lee. 2004. Semi-Automatic Construction of Korean-Chinese Verb Patterns Based on Translation Equivalency. *Proceedings of the Workshop on Multilingual Linguistic Resource*, pp. 87-92.

Jun, J.S. 2007. Co-Event Conflation for Compound Verbs in Korean. *Proceedings of the 21st Pacific Asia Conference on Languages, Information, and Computation*, pp. 202-209.

Kim, D., Z. Cui, J. Li, and J.-H Lee. 2002. A Knowledge Based Approach to Identification of Serial Verb Construction in Chinese-to-Korean Machine Translation System. *Proceedings of the First SIGHAN Workshop on Chinese Language Processing*.

Nakamura, H. 2007. Two Types of Complex Predicate Formation: Japanese Passive and Potential Verbs. *Proceedings of the Pacific Asia Conference on Languages, Information, and Computation, pp. 340-348.*

Roxas, R.R. 1998. *A Prototype Machine Translator of Simple English or Filipino Sentences Using Interlingua.* Master's Thesis, University of the Philippines at Los Baños.