Lexicon for an English-Filipino Machine Translation System

Nathalie Rose T. Lim

Jason Oliver Lat
Spencer Troy Ng
Kenneth Sze
Gene Derrick Yu

De La Salle University
2401 Taft Avenue
1004 Manila, Philippines
(632) 524-0402

limn@dlsu.edu.ph, {ricochet_008, speng_14, rushbaal, doughboy_derrick}@yahoo.com

Abstract

Lexicons are lists of words that form the language. Dictionaries and thesauri are types of lexicons. In natural language processing, applications for machine translation, question answering, information extraction, and information retrieval, among others, require an electronic lexicon. However, these applications do not just require a listing of the words of the language. Depending on the application, attributes, such as meanings, synonyms, or translations, are necessary. There are several lexicons available for different languages. However, there is currently none for Filipino that can be readily used for natural language processing. This paper presents the database design for the lexicon used by an English-Filipino machine translation system. Approaches on how the lexicon was built and improved are also discussed.

1. Introduction

Recent researches and developments on Natural Language Processing (NLP) for Machine Translation (MT) Systems have been successful in many different languages. These languages include English, Spanish, Arabic, Nihongo, Mandarin and many more; however, there has been little research with regards to the Filipino language. One of the main reason behind this is the lack of resources that would facilitate the advance in NLP research. Resources that are normally needed are electronic corpora which will serve, at the least, as a basis of comparison of result of the manual translation as opposed to machine translation. These could also be used as input for machine learning systems. Corpora may be classified as parallel or non-parallel. Parallel corpora refer to direct translations of documents. On the other hand, non-parallel corpora could either be comparable or non-comparable. Comparable corpora are not direct translations, but they are documents that talk about the same topic. Although it is easier to compare parallel documents or use them in NLP applications, this type of resource is scarce.

Another resource which is essential in NLP is the lexicon, especially for applications that require natural language understanding and generation. Word listings are not enough. Each entry should have associated attributes to lend semantics for better and reliable results in NL applications.

This paper discusses two approaches adapted by the proponents in designing the lexicon for the English-Filipino MT system. Section 2 covers the design of the database and the justification for such a design. Section 3 shows the semi-automatic approach for filling up the lexicon. Section 4 discusses the main features of the editor for manual encoding of entries. In the last section, recommendations for improvements for the lexicon are discussed.

2. Lexicon Database Design

For a MT system, the minimum requirements in the lexicon would be the term and its translation. However, since there are several senses of a word, other features are also included. Since the lexicon was built independently of the MT system, it is difficult to predict all the possible features that should be included and considered for the design of the database. This left the proponents with the task of making the lexicon database flexible in accepting attributes that may be needed later on. Since there may be several attributes that need to be modeled for each term, the database is split into different tables. The lexicon database for the MT system consists of the following tables:

1. English to Filipino table has the following fields:
    - id – unique identification number
    - English term – may be in the base form or inflected form (since entries can be added from automatic lexicon extraction)
    - Filipino translation – translation (may be more than 1 word) of the English term
    - Part of speech – the general part of speech tag (eg. Noun, verb)
    - Co-occurring words – list of words that are associated with the English term
2. English term feature table has the following fields:

- id – unique identification number (foreign key)
- attribute type – may be any attribute string that the machine translation engines need to provide information for more accurate translation (example: gender, synsetId)
    - the synsetId is based on the hypernym of the English term in WordNet.
    - this is where the proponents envision the other elements (that a lexicon should have) be stored (depending on the need of the MT system) ex. Subcategorization frame
- value – may be any string to represent the value of the attribute (example: male)

3. Filipino to English table

   In reference to number 1, English to Filipino, this table serve as its counter part for Filipino to English extraction.
   The table's fields are follows:
   - id – unique identification number
   - Filipino term – may be in the base form or inflected form (since entries can be added from automatic lexicon extraction)
   - English translation – translation (may be more than 1 word) of the Filipino term
   - Part of speech – the general part of speech tag (eg. Noun, verb)
   - Co-occurring words – list of words that are associated with the Filipino term
4. Filipino term feature table

   In reference to number 2, this serves as its Filipino counter part for word features.
   The table's fields are follows:
   - id
   - attribute type
   - value

   (Refer to number 2 for definition of fields).

The English-Filipino lexicon is separated from the Filipino-English lexicon so as to capture the words in each language that do not have direct translations to the other language. Also, many to many correspondences of terms may be addressed. An example would be the Filipino term *laba* which means to wash clothes, the Filipino term *hilamos* which means to wash face, and the general Filipino term *hugas* which means to wash (but this cannot be used with clothes (damit) or face (mukha)).

In addition, separating the two lexicon may provide a faster access time later on (when doing Filipino to English translation) if there is only one table to access.

An initial lexicon taken from the IsaWika![1], with an entry of 22,940 English to Filipino words and 19,980 Filipino to English words was ported (via a script and manual editing) into the lexicon database. To populate the database with more entries, both a semi-automatic process of lexicon extraction system and manual encoding was employed. These are discussed in the succeeding sections.

3. Automatic Lexicon Extraction

The Automatic English and Filipino Lexicon Builder (AEFLex) system is a lexicon extraction system designed for the English and Filipino language. It is automatic in the sense that the system generates candidate translations of words not found in the lexicon based on input parallel or comparable corpora [6]. This process may be considered semi-automatic because the expert user evaluates the candidate translations before it can be added into the lexicon database. The main basis of this study is that co-occurring words in a language would most likely co-occur in other languages [4].

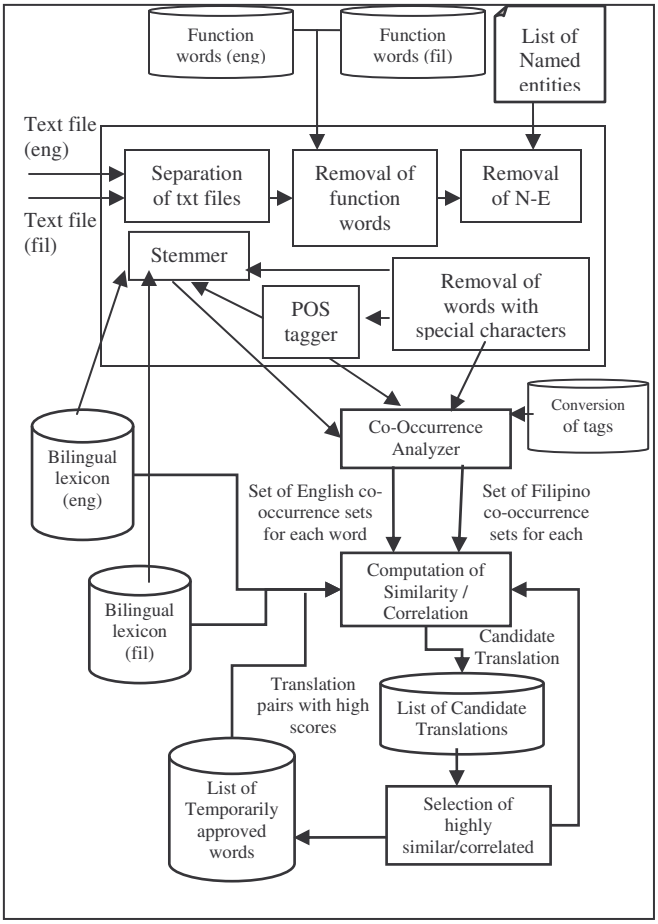

Figure 1. The Architectural Design of English-Filipino Lexicon Extractor

The system starts by accepting the text files of the English and Filipino corpora. The corpora are then stripped off of any function words, named-entities and words with special characters it may have. It then undergoes a series of preprocessing before applying the

lexicon extraction algorithm. After the preprocessing, the words then go through the co-occurrence analyzer. This component would determine co-occurrence sets by examining the context of each word in the corpora. By examining the bilingual lexicon and co-occurrence sets of each term, the correlation or similarity scores of candidate translation pairs will be computed. Finally, when the scores are already completed, the resulting list is manually checked to see which words obtained the correct translation.

Figure 1 shows the architecture of the lexicon extractor system. The three main components are Preprocessor, Co-occurrence Analyzer, Computation of Similarity/Correlation.

3.1. Preprocessor

The preprocessor removes insignificant words in the input corpora. It filters words such as function words, named entities and words with special characters. They are removed because they do not contribute to the similarity measurement scores and can only decrease the accuracy.

The system accepts the filtered list of English and Filipino corpora. The corpora may be partially tagged or untagged. If the corpus is untagged it may be tagged using the POS tagging component included in the system. The tagger used for the system simply uses the initial lexicon database as a lookup table. It repeatedly searches the lexicon for the words in the corpora, starting from the first word to the last, and returns a result each time. If the result returns exactly one match, then the part of speech tag attached to the word (in the lexicon) is assigned to the word. If, however, the word does not exist or that it has more than one part of speech tag, "nopos" is assigned.

After tagging each word in the corpora, these undergo stemming. The process of stemming starts by getting each word in the corpora, and repeatedly removes prefixes and/or suffixes the word might have; resulting in having a corpus of mostly root words. The English stemmer used in the system is Porter's stemming algorithm[9]. The Filipino stemmer used in the system was made by following the rules of Filipino word structures found in the English-Tagalog Vocabulary[8].

3.2. Co-occurrence Analyzer

After the preprocessing, the corpora will be handed to the co-occurrence analyzer. The co-occurrence analyzer determines the frequency of each word co-occurring with another word. The basis of the collocates (co-occurring word) is its window size. The system uses a default window size of 2 (meaning 2 words that come before it and 2 words that come after it). These are then passed on to the next process.

3.3. Computation of Similarity/Correlation

This computes for the score of each word and its candidate translation. The system uses the formula for correlation of Kaji[5].

After processing input corpora, the system eliminates insignificant terms based on a factor/value given by the user. Insignificant co-occurring terms are also eliminated based on a significance factor asked from the user. An insignificant term is identified based on the number of occurrence of the specific term. Refer to figure 2.

Example terms in corpus:

Word	Cooccurring words	Frequency of each co-occurring words	Total number of co-occurring words
food	eat	4	12
	water	5	
	sustainable	3	
boy	playing	2	7
	eating	2	
	playground	3	
ball	play	1	4
	roll	1	
	big	2	

if (total number of co-occurring words (refer to table 5-1) > number of unique terms in corpus / significance factor)
word is significant
else word is insignificant

Figure 2. Formula for co-occurrence factor.

Following the elimination of insignificant terms, the system then processes the co-occurring words for both corpora. All English words in the co-occurrence set will be translated to its Filipino translation by consulting the bilingual lexicon. These co-occurring words are then compared to the target translations by basing it on the frequencies of the co-occurrence sets for both English and Filipino word would be compared and the similarity scores would be computed. The formula used for computing similarities between these terms is from Kaji's lexicon extraction algorithm [5]:

$$R(sw, tw) = \frac{|C(sw) \cap C(tw)|}{|C(sw)| + |C(tw)| - |C(sw) \cap C(tw)|}$$

where sw is the source word,
tw is the target word,
C(X) is the co-occurring set of X,
R(X, Y) is the similarity between X and Y, and
| X | is the occurrence of X.

Aside from the method proposed by Kaji, et. al., the similarity formula used by Fung, et. al. was also implemented and tested. The scoring methods and evaluation are different, but accuracy is also limited due to the small lexicon.

From the different similarity measures of Fung, S1 and S3 are specifically tested. According to Fung, S1 is

often used in comparing a short query with a document text. S2, which is multiplied to S1 to compute for S3, is used in comparing two document texts. S3, on the other hand, falls somewhere in between the two. Since the focus of the testing is on the significant words, S1 was selected over S2, and S3 was also selected to test the combination of S1 and S2.

In a two-way extraction, a formula to get the combined score of extracting words from English to Filipino and Filipino to English is used for each translation candidate. Knowing that there are more unknown and less known words for the Filipino corpus than in the English, the scores from extracting Filipino to English will be lesser than the score from extracting English to Filipino. Refer to figure 3 for the formula.

Using Kaji's and Fung's algorithms, the system can also generate a combined score. The score will be the score of Kaji and the score of Fung being combined with the formula in Figure 3. This combination may result in strengthening scores or making scores lower.

score(hi, lo)
= (hi – lo) * lo + (100% – (hi – lo)) * hi

where: hi = higher score,
lo = lower score

Figure 3. Formula for combined scores or Two-way extraction

3.4. Selection of highly similar/correlated

The score computed from the previous process will range from 0.0 being the lowest to 1.0 being the highest. The user will input the threshold value that will serve as a passing mark for the extraction result. Moreover, if the score of a word is below the threshold value, that word will not be displayed in the extracted result.

3.5. Result of Automatic Lexicon Extraction

The system was tested on different corpora. At best, the extractor is only 57% accurate. The main problem involves lack of lexicon entries (several unknown words would mean that translations for co-occurring words cannot be found and thus cannot be used in the computation for candidate translation). Another problem is the absence of a morphological analyzer (the base form of a word may be found in the lexicon, but its inflected forms are not and without the morphological analyzer, these are considered as unknown words).

In addition, the system cannot handle multiword terms because the system separates the input corpora into single word terms. If the input text is already tagged, then the system can support or recognize multiword terms if the words were enclosed in curly braces.

Furthermore, the Filipino language is dynamic. There are several English words that have been assimilated to be used as a normal Filipino word. Some examples of these are "bus", "truck" or "colgate". Assimilated terms are used with Filipino or Tagalog prefixes, suffixes and infixes. Having these words in the corpora might provide additional difficulty in extracting terms.

4. Manual Encoding through Lexicon Editor

The candidate translations generated in the automatic extraction are stored temporarily in a database. This database is then loaded into an interface to allow the expert user (preferably a linguist) to identify which ones are to be included in the bilingual lexicon. However, since it is too ambitious to depend on automatic extraction of candidate translations to populate the lexicon database, additional tools were also implemented. These tools were then combined with the interface to form the Lexicon Editor.

The lexicon editor allows the user to add (refer to Figure 4), edit or delete entries from the lexicon. The editor will enable the user to add attributes to a word in the lexicon. Some examples of the attributes that the user can add are (but not limited to) gender, phonology, sample usage, and synsetID. The user can also note the variants in the spelling of a word.

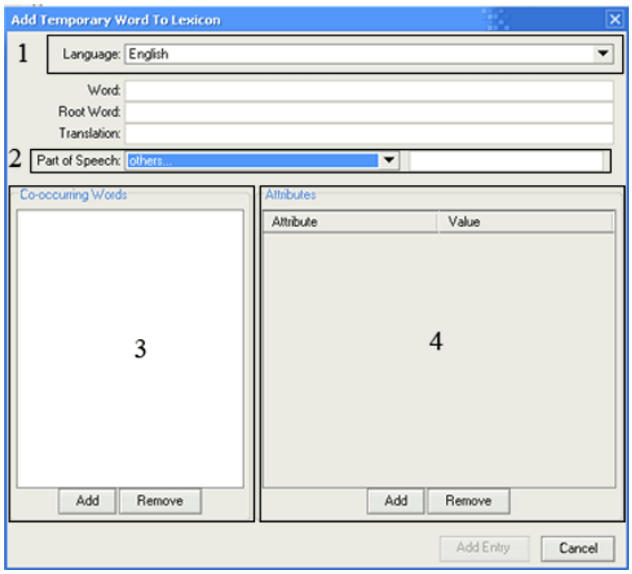

Figure 4. Screen shot of the Add Entry*

(*numbered elements just show the different sections of the screen)

Entries stored in the lexicon database may be used to generate a customized dictionary. Formats may be changed and attributes to be included in the dictionary (from those that have been given as input) may be chosen.

5. Conclusions and Recommendations

The automatic lexicon extractor system can extract unknown words from corpora; however, there are still

several factors that decrease accuracy or produce incorrect results. Below are some possible solutions that can be implemented to improve the results and the accuracy of the system.

First, having efficient morphological analyzers can greatly increase the accuracy of the system. In the course of the testing, by using a stemmer (not a full blown morphological analyzer), the lexicon extractor was able to generate 8% more candidate translations and the accuracy of the candidate translations increased by 10%. Thus, by determining the root words accurately, there will be fewer words and less candidate translations to compare with. At the same time, it will also increase the count of the words linking the unknown word and the candidate translation. These two events can increase the scores for each candidate translation, as well as provide fewer candidates for each word.

The system could also improve by having an increased list of function words and a better named-entity recognition component. By successfully removing insignificant words from the system, only the relevant words will remain and the number of candidate translations and words linking the unknown word and the candidate translation will lessen. The system may then be able to achieve better results.

The approach is statistical. It uses the translations of co-occurring words to determine the candidate translation of a word. If there are several unknown words in a corpus, then computations will result to values below the threshold. This can improve by either having an initial lexicon with more words or by using corpora with a lot of words found in the lexicon.

With initial bilingual terms included either automatically or manually to the lexicon database, the attributes would still have to be encoded. The synsetIDs that come from WordNet is manually matched and encoded into the database based on the translation (or meaning) the term has. The process would be simpler if this can be done semi-automatically, thereby also creating a Filipino WordNet.

However, WordNet and other similar language resources have little mapping to syntax, no predicate argument structures, and no selectional restrictions [7]. In Berkley, a project called FrameNet was developed to document the range of semantic and syntactic combinatory valences of each word in each of its senses by a semi-automated annotation of sample sentences from corpora. Currently, the FrameNet lexical database contains more than 10,000 lexical units (word-meaning pair), more than 6,000 of which are fully annotated, in nearly 800 hierarchically-related semantic frames. A semantic frame is a conceptual structure that depicts a type of situation, object, or event along with its participants and props (arguments). Each sense of a polysemous (multiple meaning) word belongs to a different semantic frame [10]. These information would also be beneficial in a lexicon for machine translation, so a Filipino FrameNet may also be developed to depict events (which are embodied by the verbs of a language).

It is better still if a database would be able to model both the hierarchical relations for objects (i.e., nouns) as in the case of WordNets and the semantic frame as in the case of FrameNets. Once such a notation is established, it may facilitate the extension to other Philippine languages. And when this resource becomes available, customizing existing NLP applications for these languages will not be too far behind.

6. References

[1] Borra, A. et al. (1997). IsaWika. Philippines: University of the Philippines.

[2] Coelho, P. (1988) The Alchemist. United States.

[3] Cornell University. Ithaca, New York, USA. [online]. Available: ftp://ftp.cs.cornell.edu/pub/smart/english.stop

[4] Fung, P. (1998). A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora. University of Science and Technology, Hong Kong.

[5] Kaji, H. et al. (1996). Extracting Word Correspondences from Bilingual Corpora Based on Word Co-occurrence Information. Tokyo, Japan.

[6] Lim, N., Lat, J., Ng, S., Sze, K., Yu, G. (2006). Extraction Of English-Filipino Lexicon From Corpora. 8th Science and Technology Congress, De La Salle University-Manila, March 8, 2006, Manila, Philippines. ISSN: 1908-1154

[7] Palmer, Martha. (2000). Standardizing Multilingual Lexicons. Workshop on Web-based Language Documentation and Description. Available online: http://www.ldc.upenn.edu/exploration/expl2000/papers/palmer/palment.ppt. (Last accessed: January 24, 2007)

[8] Panginiban, J. (1946). English-Tagalog Vocabulary.

[9] Porter's Stemming Algorithm. [online]. Available: http://www.tartarus.org/~martin/PorterStemmer/

[10] Ruppenhofer, Josef, Ellsworth, Michael, Petruck, Miriam, Johnson, Christopher, Scheffczyk, Jan. (2006). FrameNet II: Extended Theory and Practice. Available online: http://framenet.icsi.berkeley.edu/book/book.pdf. (Last accessed: January 26, 2007)

[11] Sadat, F. et al. (2003). Learning Bilingual Translations from Comparable Corpora to Cross-Language Information Retrieval: Hybrid Statistics-based and Linguistics-based Approach. Nara, Japan: Nara Institute of Science and Technology.

[12] Zaide, G. (1999) Jose Rizal: Life, Works and Writings of a Genius, Writer, Scientist and National Hero. Quezon City: All-Nations Publishing Co.

Machine translation, sometimes referred to by the abbreviation MT (not to be confused with computer-aided translation, machine-aided human translation or interactive translation), is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one language to another. On a basic level, MT performs mechanical substitution of words in one language for words in another, but that alone rarely produces a good translation because recognition of whole Online Filipino to English Translation Software - Official Filipino Site for Translating Filipino (Tagalog) to English for FREE. Typing 'Gustung-gusto kong makipag-usap sa filipino' will translate it into 'I love speaking in Filipino'.Â Many websites provide services to translate english for a few dollars. While it is a good idea to pay for translating lots of text (such as books, articles) and for professional service, there is no point paying for commonly used sentences, greeting messages, and other informal use. For these purposes, this tool can be used.Â Their system use machine-language technologies to bring together some of the cutting edge technologies such as artificial intelligence (deep learning), big data, web APIs, cloud computing etc to perform higher quality translations. Translate from English to Filipino. Be it words, phrases, texts or even your website pages - Translate.com will offer the best.Â Type your text & get English to Filipino translation instantly. Communicate smoothly and use a free online translator to instantly translate words, phrases, or documents between 90+ language pairs. Welcome. Please log in to proceed and have access to unlimited machine translation, access to professional translation service along with other benefits. Donâ€™t have an account at Translate.com yet? Sign up for free within minutes to access a whole set of various translation options and utilize your free words by ordering from qualified translators. You have reached the character limit for the last 2