Template-Based English-Filipino Machine Translation System

Ethel Ong
Kathleen Go   Vince Andrew Nuñez
Manimin Morga  Francis Veto
De La Salle University
2401 Taft Avenue
1004 Manila, Philippines
(632) 524-0402
onge@dlsu.edu.ph
go.kathleen@gmail.com, Min_Morgz@yahoo.com, link7488933@yahoo.com, francisveto@yahoo.com

ABSTRACT
This paper presents a template-based machine translation system that extracts templates from a given bilingual corpus, then uses these templates to perform bi-directional English-Filipino translations. The system extended the similarity template learning algorithm of Cicekli and Guvenir [2] by refining existing templates and deriving templates from previously learned chunks. Chunk alignment and splitting algorithms are integrated into the training process to improve the quality of the extracted templates. Tests results verified that a strict chunk alignment scheme used in the training process, including the filtering of commonly occurring words, generated better templates and chunks. Correct extraction of templates and chunks during the learning process led to reduced word and sentence error rates by as much as 50% during translation. Tests also showed that the translation with the highest score selected from a set of candidate translations is consistently the best choice as validated against automatic evaluation methods.

Keywords
Example-Based Machine Translation, Templates, Machine Learning, Bilingual Corpus, Natural Language Processing

1. INTRODUCTION
TExt Translation is an example-based machine translation system that translates English sentences to Filipino and vice versa, using templates and chunks that have been extracted or learned from a set of bilingual corpora. This approach was used in order to maximize the amount of important information derived from the input corpora, specifically example translation sentence pairs between source and target languages, using the least amount of linguistic resources. Utilizing templates is more flexible as compared to word-for-word translation [4] and to pre-defined rules which would be hard to modify at a later time. Moreover, as the Filipino language is evolving in response to calls for globalization, chunks learned during training can supplement limited entries in the lexicon, and are used in further learning sessions to extract more templates from the corpus.

Kaji [4] defines a translation template as “a bilingual pair of sentences in which corresponding units (words and phrases) are coupled and replaced with variables”. Kinoshita [5] presents a translation template to contain “at least a pair of patterns, namely source and target patterns, each of which consists of constants and variables”, and that a source pattern is used as a comparison reference to the sentence being translated while the target pattern is used to generate the translation of the input. The patterns preserve the ordering of words in the translation, regardless of the variance in the sentence structures of the source and target languages, which, as tests results have shown, increases the quality of the translation.

A simple template looks as follows:

T1: I ate <Cl> ↔ Kumain ako ng <Cl>

“I ate” and ‘Kumain ako ng’ are called the constants of template T1 while <Cl> is a variable representing a chunk, which is a sequence of words used to substitute the variables in a given template. Possible chunks that could be learned from a given bilingual corpus and used for the sample template T1 above are as follows:

<Cl.1>: bananas ↔ mga saging
<Cl.2>: a fruit ↔ isang prutas

2. TEMPLATE EXTRACTION
Figure 1 shows the four main components of TExt Translation, namely analysis, training, translation, and knowledge base. During input analysis, the bilingual corpus is processed and the output is passed to the succeeding module, which could either be the training module or the translation module. The analysis phase consists of three main steps for the training module: sentence segmentation, tokenization, and unit alignment. For the translation module, only sentence segmentation and tokenization are performed. Sentence segmentation divides the input text into sentences. Tokenization then divides the sentences into smaller fragments such as words, punctuation marks and other symbols. Unit alignment aligns corresponding tokens between the source sentence and the target sentence according to the entries in the lexicon. In the training module, aligned sentence units are analyzed to acquire templates and chunks. There are three subprocesses in the training module, which are discussed in the following sections.

2.1. Template Refinement
Template Refinement (TR) compares an aligned sentence pair against existing templates in the knowledge base. An aligned sentence is said to match a given template whenever it contains a token that matches exactly with a corresponding token in the template itself. There must be a corresponding match in both the source and target languages in order for the template to be considered. Through these similarities a candidate refinement is identified. Consider the following aligned sentence pair S1 and template T1:
There are cases when not all constants in a given template do not make a new template but instead refines its remaining new similarities with the knowledge base. If the aligned sentence pair does not have a token matching

2.2. Derivation of Templates from Sentences

If an aligned sentence pair does not have a token matching those of any of the templates in the knowledge base, the sentence pair is then compared with other aligned sentence pairs. This process is termed Derivation of Templates from Sentences (DTS). Consider the following sentences:

S1: I ate meat \(\leftrightarrow\) Kumain ako ng karne.
S2: We happily went to the hospital. \(\leftrightarrow\) Mahinahon kaming pumunta sa pagamutan.
S3: The woman walked happily. \(\leftrightarrow\) Masayang naglakad ang babae.
S4: The couple walked happily. \(\leftrightarrow\) Masayang naglakad ang mag-asawa.

Only the underlined segments match between S2 and T2. TR does not make a new template but instead refines T2 to reflect its remaining new similarities with S2. Template T2 is discarded; a new template, T3, and a new chunk group \(<C3>\) are created:

T3: We \(<C3>\) went to \(<C2>\) \(\leftrightarrow\) kaming pumunta sa \(<C2>\).
\(<C2.5>\) the hospital \(\leftrightarrow\) pagamutan
\(<C3.1>\) happily \(\leftrightarrow\) masayang
\(<C3.2>\) calmly \(\leftrightarrow\) mahinahong

The DTS algorithm takes all similar tokens between the source and target sentences in S3 and S4 (underlined above) and preserves them as constants in the new template T4 while the differing elements are created as chunks in \(<C4>\).

T4: The \(<C4>\) walked happily \(\leftrightarrow\)
Masayang naglakad ang \(<C4>\).
\(<C4.1>\) woman \(\leftrightarrow\) babae
\(<C4.2>\) couple \(\leftrightarrow\) mag-asawa

There are cases when a single long differing chunk in the source sentences corresponds to two short differences in the target sentences. Using S3 once more and a new sentence S5:

S3: The woman walked happily. \(\leftrightarrow\) Masayang naglakad ang babae.
S5: The man watched happily. \(\leftrightarrow\) Masayang nanood ang lalaki.

Upon initial inspection, the source sentences have a single long differing chunk, “woman walked” for S3 and “man watched” for S5, whereas in the target sentences there are two differing simple chunks, namely “naglakad” and “babae” for S3 and “nanood” and “lalaki” for S5. In this case, the single long chunk is split into two shorter chunks to see whether the resulting subchunks would match to either of the corresponding differing chunks in the target, a process called Strict Chunk Alignment with Splitting (SCAS). The new template T5 and chunks \(<C5>\) and \(<C6>\) look as follows:

T5: The \(<C5>\) \(<C6>\) happily. \(\leftrightarrow\)
Masayang \(<C6>\) ang \(<C5>\).
\(<C5.1>\) woman \(\leftrightarrow\) babae
\(<C5.2>\) man \(\leftrightarrow\) lalaki
\(<C6.1>\) walked \(\leftrightarrow\) naglakad
\(<C6.2>\) watched \(\leftrightarrow\) nanood

2.3. Derivation of Templates from Chunks

When templates cannot be derived from aligned sentence pairs using TR and DTS, the system then performs Derivation of Template from Chunks (DTC). While the system is learning templates from aligned sentence pairs, it is able to simultaneously extract a significant number of chunks. As there are no other resources available – other than the lexicon – which could be used in the learning process, the chunks are also reused because they are additional information from actual examples. Consider the new sentence S6 and existing chunks from the knowledge base:

S6: Filipinos are cheerful and hospitable. \(\leftrightarrow\) Masayahin at mapanauhin ang mga Pilipino.
\(<C6.4>\) Filipinos \(\leftrightarrow\) mga Pilipino
\(<C7.9>\) hospitable \(\leftrightarrow\) mapanauhin

DTC simply takes matching chunks from the knowledge base, (chunks \(<C6.4>\) and \(<C7.9>\)) and uses them to substitute parts of the aligned input sentence. In this example, input sentence S6 is transformed to generate the new template T6:

T6: \(<C6.4>\) are cheerful and \(<C7.9>\) \(\leftrightarrow\)
Masayahin at \(<C7.9>\) ang \(<C6.4>\).

3. TRANSLATING USING TEMPLATES

During translation, input sentence tokens are analyzed to collect candidate templates and chunks from the knowledge base. A template or a chunk is considered a candidate if it has at least one word used in the input sentence. Consider input sentence II:
I1: The pretty girl ran.

All chunks and templates having the words “The”, “pretty”, “girl”, and “ran” will be added to the list of candidates. This is necessary because any word in the input sentence can be a variable and it is possible that only one constant (or word) is present in the candidate.

The list of candidates are then reviewed and filtered to exclude inappropriate matches. A candidate is an inappropriate match if it contains words that are not found in the input sentence, if it is longer than the sentence, or if the word order is different from the input sentence. For example, only the chunk “pretty <C28>” will be useful for input sentence I1. Chunks “boy and girl” and “ran away” are inappropriate and will be removed from the candidates list.

The remaining candidates are assigned scores according to the structure of the template or chunk, the presence or absence of chunk variables in templates, and the presence of word matches in templates. The translation output that produces the highest total score is used for the given input. If there are candidates having the same score as the highest scorer, the first candidate with the highest score will be selected.

Given the input sentence I2, candidate template T12, and candidate chunk <C8.1> below:

I2: I ate vegetables.  
T12: <C8> ate <C1>.  
<C8.1>: I

In the first iteration of the scoring algorithm, T12 will receive points for the matching string constant “ate” and for the presence of the two chunk variables <C8> and <C1>. During the second iteration, chunk <C8> will be replaced with more specific chunk <C8.1>, resulting in the partial match “I ate <C1>.” Additional points will be added due to the matching string constant “I”, and because the assigned chunk <C8.1> has the same domain as the target chunk <C8>. However, for chunk <C1>, no match can be found. No points will be given for this chunk since word-for-word translation will be performed.

Note that during the candidate identification process, the matching chunks may have domains that are different from the one specified in the matching template. This allows the use of chunks from various domains as candidates in lieu of the lexicon which uses word-for-word translation. However, in the scoring process, a match that has the same domain as specified will be given a higher score.

During learning, the extracted templates and chunks have bi-directional properties; that is, the same set of templates and chunks will be learned for a given training set, regardless if the source language is English or Filipino. This same property should hold true during translation. If an exact match is formed for a given sentence in English, the same candidate would be formed for the Filipino sentence. However, it is possible that a template or chunk in the source language contains no constants but the one in the target language does. The bi-directional property will not hold in this situation, and the resulting translations in both directions can differ.

4. TEST RESULTS

Various tests were conducted to determine the quality of both the templates and chunks extracted during training and of the output generated during translation.

4.1. Learning Module

To test the two algorithms, namely LCA (Loose Chunk Alignment) and SCAS (Strict Chunk Alignment with Splitting), of the learning module, the 50 Ways corpus (referred to as corpus #4 in this research) provided by the Filipino Department of De La Salle University – Manila was used. This bilingual corpus by Cindy Haynes has numerous sentences with the same structure and similar tokens. TExt’s TR (see Section 2.1) and DTS (see Section 2.2) processes will not be able to derive templates if the structures of the input sentences are totally different and no matching tokens between a sentence pair and another sentence pair or a template pair exist.

The SCAS algorithm used in deriving templates from sentences requires that all tokens are aligned and the number of given chunks in the source language is equal to that in the target. This results in learning more templates that are of good quality, as shown in Table 1, as compared to the LCA approach. Correctness in Table 1 refers to the actual templates and chunks learned as well as the proper alignment of tokens in the source and target template or chunk. The LCA approach has a higher error rate; moreover, the resulting templates and chunks are not bi-directional since the approach did not learn the same number of templates and chunks from the same corpus.

<table>
<thead>
<tr>
<th>Table 1. Test Results for Chunk Alignment Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corpus Details</strong></td>
</tr>
<tr>
<td># of sentence pairs in Bilingual Corpus #4</td>
</tr>
<tr>
<td>English to Filipino</td>
</tr>
<tr>
<td># of template pairs learned</td>
</tr>
<tr>
<td># (%) correct template pairs</td>
</tr>
<tr>
<td># of chunk pairs learned</td>
</tr>
<tr>
<td># (%) correct chunk pairs</td>
</tr>
<tr>
<td># of unused sentence pairs</td>
</tr>
<tr>
<td># (%) correct sentence pairs</td>
</tr>
<tr>
<td>Filipino to English</td>
</tr>
<tr>
<td># of template pairs learned</td>
</tr>
<tr>
<td># of chunk pairs learned</td>
</tr>
<tr>
<td>% correct chunk pairs</td>
</tr>
<tr>
<td># of unused sentence pairs</td>
</tr>
</tbody>
</table>

Further analysis of the extracted templates showed that there are too many frequently occurring words which do not contribute to the quality of the derived templates. These common words, which were identified based on existing studies of English and Filipino corpora, need to be filtered from becoming the remaining constant in any given template because it would match with many aligned sentences, leading to templates with a small coverage during translation.

Tests were conducted to identify how much more useful the generated templates and chunks would be if a filter for common words was implemented. Four corpora were used. Sentences in corpora numbers 1 to 3 were manually created by the proponents, and fall under different domains. The fourth corpus is the same 50 Ways corpus above.

A version of the Training Module, called NCWF (No Common Words Filtering) does not implement such a filter while another implementation, called CWF (Common Words Filtering) does. The results in Table 2 indicate that the CWF version produces more templates but fewer chunks.

NCWF generated fewer templates because TR is performed more often to produce templates with common word remainders. This also resulted in having derived more chunks. CWF generated more templates and fewer chunks which is preferable because templates are able to capture proper
sentence structures, thus preserving word order in the resulting translation. More templates would also mean more candidates for refinement in subsequent training executions.

Table 2. Test Results for Filtering Algorithms

<table>
<thead>
<tr>
<th>Corpus Details</th>
<th>NCWF</th>
<th>CWF</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sentence pairs in Bilingual Corpus #1</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td># of sentence pairs in Bilingual Corpus #2</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td># of sentence pairs in Bilingual Corpus #3</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td># of sentence pairs in Bilingual Corpus #4</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>Total # of sentence pairs</td>
<td>163</td>
<td>163</td>
</tr>
</tbody>
</table>

| # of template pairs learned | 50   | 73  |
| # (%) template pairs learned from bilingual corpus #4 | 31 (52.54%) | 30 (41.10%) |
| Total # of chunk pairs learned | 237  | 210 |
| # (%) chunk pairs learned from bilingual corpus #4 | 47 (19.83%) | 38 (16.57%) |
| Total # of unused sentence pairs | 25   | 25  |
| # (%) unused sentence pairs from bilingual corpus #4 | 18 (55.17%) | 17 (56.62%) |

4.2. Translation Module

Automatic evaluation of the translated outputs was carried out using the online MT evaluation tools available in DCU – Dublin City University. Three bilingual corpora, used as input files, were first manually tagged in the Standard Generalized Markup Language (SGML) format required by the automated evaluator. The metrics used are the word error rate, sentence error rate, and bilingual evaluation understudy for the three system versions LCA, SCAS-NCWF and SCAS-CWF. Corpus 3 reflects translations for exactly the same document learned, corpus 5 reflects translations for sentences derived from corpora 1-4 used for training the system, while corpus 7 reflects translations for sentences derived from the second half of sentences from corpora 1-4 of which the first half was used for training.

Word error rate (WER) computes the percentage of words which are to be deleted or replaced in the translation with the aim of obtaining the reference sentence. Figure 2 shows that the result of translating from the same corpus produces very low WER (corpus 3). WER is still commendable for translating based on structures from previously learned sentences (corpus 5). Furthermore, translation of sentences which have relatively different structures than those learned lead to low quality output (corpus 7).

The percentage of sentences whose translations have not exactly matched the corresponding sentence in the reference sentences is computed by the sentence error rate (SER). SER takes into consideration the incorrectness of entire sentences. It is therefore understandable that the error rates shown in Figure 3 are somewhat significantly higher than those of WER in Figure 2. The same observations and conclusions regarding the performance of the various versions on the three corpora are the can be made for SER.

The bilingual evaluation understudy (BLEU) metric is a modified n-gram precision measure. It employs a weighted geometric average of n-gram matches between test sentences and reference sentences, which is modified to penalize over-generation of correct word forms. A multiplicative brevity penalty is also incorporated which penalizes test sentences found to be shorter than the reference sentences as computed at the corpus level. The resulting score is a numeric metric intended to indicate the closeness of a set of test sentences to their reference sentences considering the length, word order and word choice. The scoring captures both adequacy and accuracy. Adequacy is satisfied using the same words (unigrams) as in the references. Accuracy is reached with the longer n-gram matches.

BLEU measures the accuracy and adequacy of translations and so it evaluates in a somewhat opposite manner to those for WER and SER, that is, the higher the score, the better. One observation is that eventhough the exact corpus used for training is also used for translation, the BLEU scores (Figure 4) are not exactly perfect. This could be attributed to unused sentences for which no templates could be generated and for which the often unreliable word-for-word translation is used. This case holds true even for the results of WER (Figure 2) and SER (Figure 3).
As can be seen from the test results in Figures 2 – 4, the SCAS-CWF version consistently performs better than the LCA or SCAS-NCWF versions. Correct extraction of templates and chunks that are of good quality during the learning process led to reduced word and sentence error rates by as much as 50% during translation.

Tests were also conducted to determine the correctness of the scoring algorithm. Ten input sentences that were correctly translated by the system were identified. For each of these sentences, the top five candidate translations are submitted to the automatic evaluator for comparison against the reference translation. Ten input sentences that were incorrectly translated by the system were also evaluated in the same manner. Results in Table 3 show that the translation with the highest total score is consistently the best choice as validated against automatic evaluation methods.

Table 3. Test Results for Top 5 Candidates for Correctly and Incorrectly Translated Input Sentences

<table>
<thead>
<tr>
<th>Top Candidates</th>
<th>Correctly Translated</th>
<th>Incorrectly Translated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Version</td>
<td>WER (%)</td>
</tr>
<tr>
<td>Top 1</td>
<td>CWF</td>
<td>0.00</td>
</tr>
<tr>
<td>Top 2</td>
<td>CWF</td>
<td>39.89</td>
</tr>
<tr>
<td>Top 3</td>
<td>CWF</td>
<td>53.44</td>
</tr>
<tr>
<td>Top 4</td>
<td>CWF</td>
<td>58.57</td>
</tr>
<tr>
<td>Top 5</td>
<td>CWF</td>
<td>59.57</td>
</tr>
</tbody>
</table>

5. CONCLUSION

Learning translation patterns from examples provides a more flexible MT system that is less dependent on limited linguistic resources. TExT Translation is able to generate quality translation of English or Filipino documents, using templates and chunks that the system acquired during training from bilingual corpora. The Strict Chunk Alignment with Splitting (SCAS) algorithm using Common Words Filtering (CWF) derived more templates and fewer chunks, which aid in producing better translation quality since sentence structures are captured and preserved. Moreover, the availability of more templates allow for refinement in subsequent training sessions. The learning algorithm employed in this research also allows for chunks to be learned directly from the given examples, thus supplementing the available lexicon. These chunks are also utilized in deriving new templates from input sentences where no matching existing template can be found.

However, there are still some issues that need to be resolved to further improve the system. The algorithm for pairing similarities between tokens of two different sentences during learning needs to address disambiguation of matching. Currently, the implementation looks for the first encountered similar token in the other language regardless of the other occurrences throughout the rest of the sentence.

Words are found to have different meanings depending on the context of the sentence that it occurs in and so consequently have different translations in the other language. In the current implementation, the first corresponding translation of a given word, regardless of context, is used even though another translation may be much more appropriate than the first.

Template refinement is a major component in the learning process in this system. It is possible, but not verified, if the refinement of chunks would be able to provide better information coverage.

6. References


