

Learning Translation Rules from Bilingual English – Filipino Corpus

**Michelle Wendy Tan, Raymond Joseph Ang, Natasja Gail Bautista,
Ya Rong Cai, Bianca Tanlo**

De La Salle University - Manila

2401 Taft Avenue

1004 Manila, Philippines

tanmw@dlsu.edu.ph, 10137661@dlsu.edu.ph, 10598677@dlsu.edu.ph,

10592709@dlsu.edu.ph, 10127003@dlsu.edu.ph

Abstract

Most machine translators are implemented using example based, rule based, and statistical approaches. However, each of these paradigms has its drawbacks. Example based and statistical based approaches are domain specific and requires a large database of examples to produce accurate translation results. Although rule based approach is known to produce high quality translations, a linguist is necessary in deriving the set of rules to be used. To address these problems, we present an approach that uses the rule based approach in translating from English to Filipino text. It incorporates learning of rules based on the analysis of a bilingual corpus in an attempt to eliminate the need for a linguist. The learning algorithm is based on seeded version space learning algorithm as presented by Probst (2002). Implementation of the algorithm has been modified to allow learning of non-lexically aligned languages and to adapt to the complex free word order of the Filipino language.

1. Introduction

The demand for language translation has greatly increased as globalization takes place. Some businesses have turned to machine translators (MT) that usually provide fast and consistent translation results as compared to human translators. However, the quality of such translation services is usually poor as compared to text translated by humans. Several MT paradigms have attempted to improve the quality of translation such as rule based, example based, and statistical-based MT. Rule based MT systems usually have impressive results for a given domain. However, creating the translation rules is tedious and time consuming, requiring a linguist who thoroughly knows the construct of both languages. In addition, since language is constantly changing, these rules should be updated regularly in order to maintain the system. Example based MT systems presents a different approach. It uses a bilingual corpus as the basis for translation. Although this approach is more flexible, the quality of the translation greatly depends on the quality of the examples. A new approach in Statistical-based MT is by using probabilities to determine if a phrase or word occurs in the current context. Using a corpus, statistical-based MT allows flexibility to accommodate different domains. However, it still has its drawbacks. Usually the output is ungrammatical and usually unacceptable to human linguist. It is still important to incorporate abstract syntax rules to improve the quality of translation.

The paper presents an approach that combines these paradigms. Rather than building the rules by hand, the system incorporates a training phase where it learns transfer rules by examples found in a bilingual corpus. As such, since the system learns from examples, there is no need for a human linguist to generate the rules for translation. It also allows the MT system to translate in different domains. Aside from this, the corpus can be updated to accommodate changes in language.

A substantial amount of research has been done on the area of rule learning and extraction. One example is the ALLiS Algorithm by Herve (2002) that uses rule induction to generate new rules.

Patterns generate certain rules and training data is used to test it. Accurate rules are kept whereas the other are deleted. The paper by Probst (2002) presented another learning approach to handle automatic rule learning for low-density languages. The key idea is for the system to read from a training corpus and allow the system to deduce the seed rules. It uses Seeded Version Space Learning to generate seed rules and performs compositionality. Previously learned rules can be used to translate part of the seed rule to remove specificity of the rule and to make it more general.

The system presented is based on the Seeded Version Space by Probst (2002). This is applied to the problem of learning rules for translating English to Filipino. Implementation of the algorithm has been modified to allow learning of non-lexically aligned languages and to adapt to the complex free word order of the Filipino language.

2. Language Resource

The system uses an English context-free grammar (CFG), three lexicons, and bilingual corpora. For the purpose of the research, a subset of the English grammar is used, accepting only imperative and declarative sentences, containing annotations for LFG taken from Borra (1997). An English and a Filipino lexicon is used to learn rules from the corpus. The lexicon contains the word, POS tag (e.g. noun, adj, adverb), type of word (e.g. action, person, number), quantity (e.g. singular, plural, first person), and property (e.g. proper noun, male, edible). For the translation phase, a bilingual lexicon is used which contains the English word, its Filipino equivalent, and POS tag. The three lexicons contains only root words. Inflections in the corpus and input sentences are handled by the system's morphological analyzer. Finally, each bilingual corpus is assumed to be sentence aligned and syntactically and morphologically correct. If the sentences are not recognized by the parser based on the CFG, this is ignored during the training phase.

3. Training Module

The system is an English to Filipino machine translator that incorporates learning of rules from bilingual corpora. The system is divided into two modules. The Training Module employs example based learning of rules from analyzing aligned bilingual corpus. This module stores the learned rules into a repository. Figure 1 illustrates the architecture of the Training Module. In turn, the Translation Module uses these rules to translate input sentences into its corresponding target language equivalent.

Given a bilingual corpus, the Training Module extracts each English-Filipino sentence pairs using the Sentence Tokenizer. Only the English sentence is passed to the the Lexical and Morphological Analyzer where it will be tokenized into word units and where its root word will be determined using the lexicon. Based on the root word, its corresponding POS tags and word information (such as type, quantity, etc.) are attached. These are passed to the Parser where it produces all possible parse trees of the English Sentence.

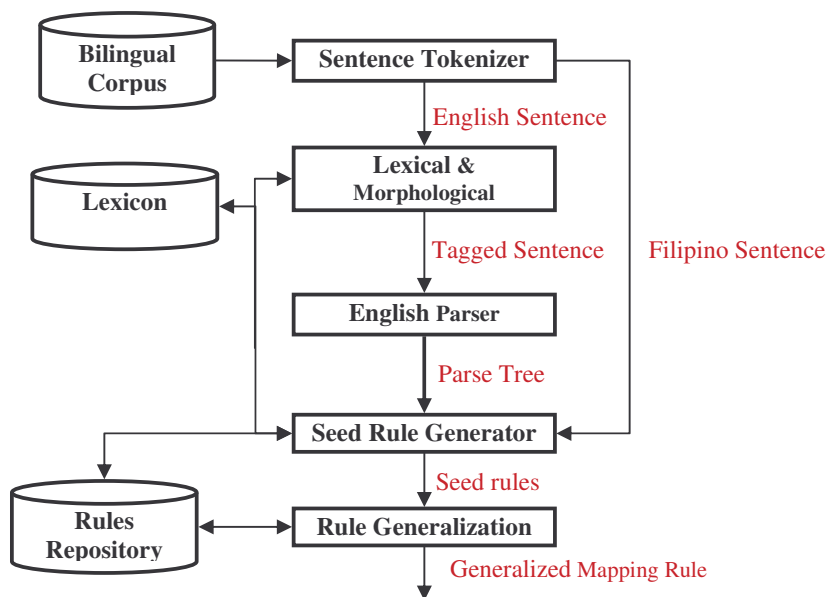


Figure 1: Architecture for Learning Module.

Each sentence pair goes through the process of seed rule generation. First, the sentence pair is automatically aligned word by word based on the lexicon. The alignment algorithm uses the parse tree and the Filipino sentence. The parse tree is traversed in order to reach the leaf nodes containing the English word. The English word is compared to each word in the Filipino sentence. If a match is found, the Filipino word is mapped and tagged with the information attached in the English word. Figure 2 illustrates the alignment.

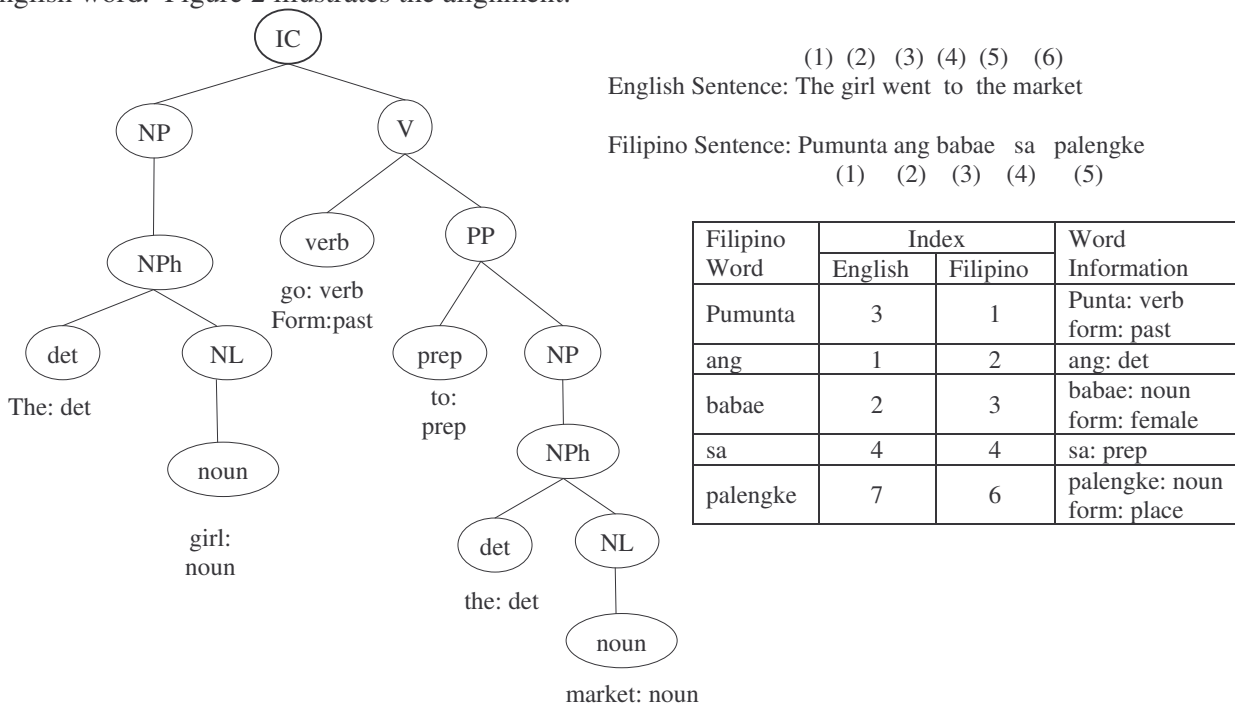


Figure 2: Alignment of English and Filipino Sentence Pair.

The algorithm by Carbonell (2002) required that the pair is lexically aligned. However English and Filipino cannot always be lexically aligned. Figure 3 illustrates this problem. In Filipino, the word "Si" identifies the succeeding noun as a person. For the purpose of learning, a limited list of constants were identified.

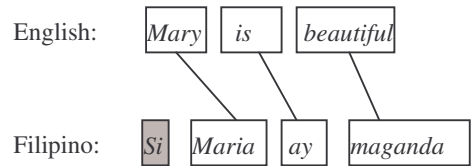


Figure 3: Lexical Alignment of English and Filipino.

Based on the aligned sentence, parse tree, and constraints, the seed rules are generated. Figure 4 shows the format of the seed rule.

Seed Rule Format	Example	Description
Production Rule	NP	The left hand side of the rule
English Rule	det noun	The right hand side of the rule in English Grammar
Filipino Rule	noun	The right hand side of the rule in Filipino Grammar
XCons	((X0 POS) = det);((X1 POS) = noun);((X1 quantity) = singular);,	The information about the terminals in English
YCons	((Y0 POS) = constant); ((Y1 POS)= noun);,	The information about the terminals in Filipino
XYCons	NP	The Filipino Rule of the parent of this learned rule

Figure 4: Seed Rule Format.

To create the rule, the module iterates through the English parse tree using depth first search. For every pass, the module searches for a non-terminal node and treats this as the root node for the next iteration. The new parse tree is traversed until a terminal node is reached. The module retrieves its POS tag and constraints, then generates the English rule. Using the aligned sentence pair, the corresponding aligned Filipino word is retrieved by including the POS tag of the Filipino words, index in the Filipino sentence, together with the different constraints of the English. To illustrate the seed rule generation, in the parse tree presented in Figure 2, the first node IC is the root node. In every recursion, a seed rule is generated. At this time, an IC seed rule is created with no XY Constraint. IC is filled in the Production Rule entry of the seed rule. Traversing this tree results to terminal nodes det, noun, verb, prep, det, and noun. The information about each terminal node is stored in the English Rule and the x constraint is added using the rule with its corresponding index in the English sentence. Figure 5 illustrates the seed rule.

The: det
girl: noun
go: verb
Form: past
to: prep
the: det
market : noun

English Rule: det noun verb prep det noun

X Constraint: ((X0 POS) = det);,
((X1 POS) = noun);((X1 Type) = person);,
((X2 POS) = verb);((X2 Type) = action);,
((X3 POS) = prep);,
((X4 POS) = det);,
((X5 POS) = noun);((X5 Type) = place);,

Figure 5: Partial Seed Rule for IC (English Rule).

After generating the English rule, the Filipino aligned pair is used. Using the Filipino sentence, each Filipino word is retrieved and its POS and information are stored. These are used to generate the Filipino Rule and the Y Constraint. Figure 6 illustrates the Filipino seed rule. The same process is done for the succeeding nodes under IC.

Punta: verb
form: past
ang: det
babae: noun
form: female
sa: prep
palengke: noun
form: place
Punta: verb
form: past

Filipino Rule: verb det noun prep noun

X Constraint: ((Y0 POS) = verb);((Y0 Type) = action);,
((Y1 POS) = det);,
((Y2 POS) = noun);((Y2 Type) = person);,
((Y3 POS) = prep);,
((Y4 POS) = noun);((Y4 Type) = place);,

Figure 6: Partial Seed Rule for IC (Filipino Rule).

To complete the seed rule generation, each rule goes through a process of compositionality where low-level rules are to produce a higher level representation. Given a seed rule, the system iterates through all existing rules. It selects candidate rules and checks if it can correctly represent a chunk in the seed rule. If it applies, then the candidate rule replaces the English and Filipino rule, and merges their constraint. Figure 7 illustrates the comparison of the seed rule (SR) and candidate rule (CR). The CR is able to represent the chunk of the SR. This is determined by comparing the chunks of the SR and CR's English Rule. In addition, the Filipino Rule, and corresponding X and Y constraints matches. As such, the Figure 8 illustrates the resulting seed rule applying compositionality.

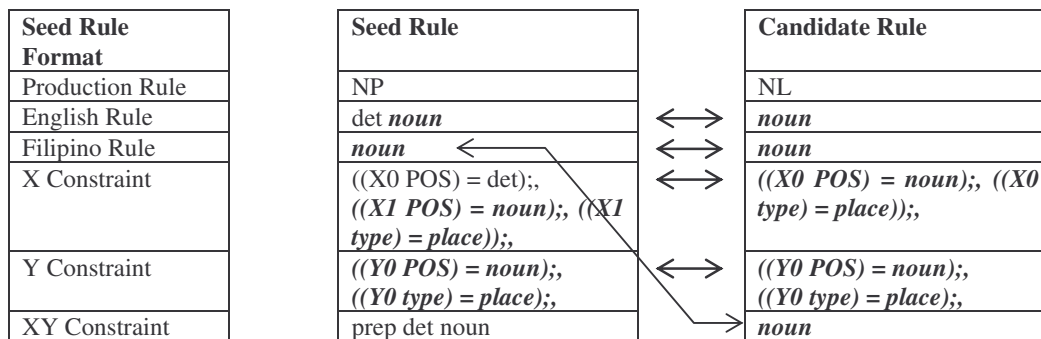


Figure 7: Comparison of Seed Rule and Candidate Rule.

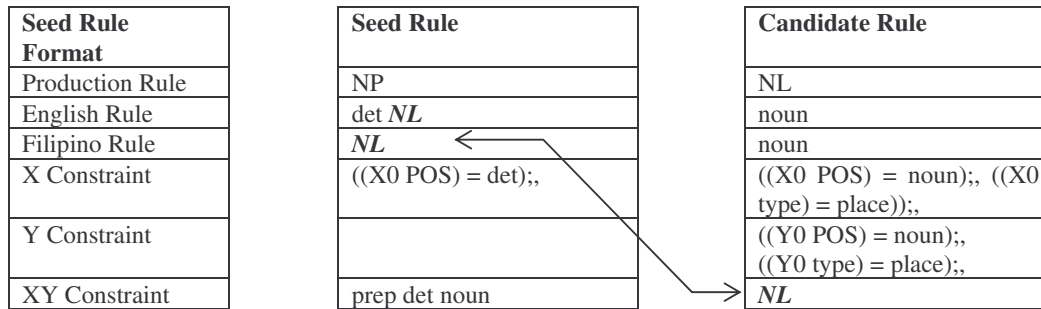


Figure 8: Resulting of Seed Rule using Compositionality.

A rule is determined to be learned by running the rule into the translation engine. If the Filipino translation is the same as the original Filipino sentence, then the rule is partially accepted and saved as a seed rule. If there is a constant word identified, then this is included in the Filipino rule as a constant terminal node.

There are some cases wherein a phrase in English is split into components when translated into Filipino. Take the English sentence “I ate an apple” where the verb phrase is “ate an apple.” When translated into its Filipino counterpart, the sentence output would be “Kumain ako ng mansanas.” The verb phrase, in this case is “kumain” “ng mansanas,” has been separated into two parts by the focus “ako.” In effect, in Filipino, the phrase can no longer be considered a verb phrase. The seed rule generation is likewise able to capture this kind of phenomenon.

Finally, the seed rules are generalized in order to compress seed rules generated from all sentence pairs in the corpus. If there are two rules that can be generalized, this pair passes through the Seeded Version Space Algorithm. If there are more than two rules, the Clique algorithm is used in order take all combinations into consideration. The Seeded Version Space Algorithm based on (Carbonell, 2002) works in three iterative steps. These are: (1) Delete value constraints; (2) Delete agreement constraints; and (3) Merge values into agreement constraints. These steps are repeated for all constraints. Figure 9 illustrates the process of generalization.

Seed Rule 1:

Production Rule	English Rule	Filipino Rule	XCons	YCons	XYCons
NP	pronoun	pronoun	((X0 Type) =identification); ((X0 POS)=pronoun); ((X0 Quantity)=3),;	((Y0 POS)=pronoun),;	VP NP

Seed Rule 2:

Production Rule	English Rule	Filipino Rule	XCons	YCons	XYCons
NP	pronoun	pronoun	((X0 Type) =identification); ((X0 POS)=pronoun); ((X0 Quantity)=3),;	((Y0 Type) =identification); ((Y0 POS)=pronoun); ((Y0 Quantity)=3),;	VP NP

Generalized Rule:

Production Rule	English Rule	Filipino Rule	XCons	YCons	XYCons
NP	pronoun	pronoun	((X0 Type) =identification); ((X0 POS)=pronoun); ((X0 Quantity)=3),;	((Y0 POS)=pronoun),;	VP NP

Figure 9: Seed Rule Generalization

4. Evaluating Rules

When the Seeded Version Space Learning Algorithm successfully produces a generalized rule, it is not readily accepted as a correct rule by the system. The system verifies that the resulting rule would still be able to translate the sentences that the rules prior to generalization can translate. It should be able to translate 80% for the sentences before it is to be accepted as a permanent rule. This is with consideration to the possible errors in the encoding of both the Morphological Analyzer and Lexicon. As these two components are improved, the threshold value should also be adjusted accordingly.

5. Translation Engine

Translation begins by processing the input sentence into its corresponding English parse tree. Using the English parse tree and the rules from the repository, the module attempts to construct the Filipino parse tree. The Filipino parse tree presents the proper sequencing of words to be expected in the final translated output. In the meantime, the leaf nodes of the Filipino parse tree contain English words together with its constraints and morphological tag. The tree is traversed using depth-first search to ensure that the sequencing of words is followed. The Filipino parse tree is passed to a translation function where each English word is translated to its corresponding Filipino word based on its constraints. After this function, the Filipino sentence is produced.

6. Results

The Training Module was tested by entering one sentence pair at a time from a corpus of 500 sentences with over 4,000 words. The rules generated were manually verified by comparing it with the structure of the sentence based on the CFG. From this, the system was able to achieve 100% of correctness in rules generation.

A linguist evaluated the results of the Training Module. From 35 sentence pairs that generated 200 learned rules, 6% of the extracted rules were evaluated as incorrect. This was due to incorrect Filipino translations in the corpus during the training phase. 20% of the sentence pairs were not able to generate rules because these sentences were not accepted by the Parser. Finally, 74% of the sentence pairs were able to generate correct and accurate learned rules. A summary is presented in Table 1.

Table 1: Rate of Correct Rules Produced.

Verdict	Percentage Value
Incorrect	6%
Not Accepted by Parser	20%
Accepted and Correct	74%

The system was also tested for the effects of learning new rules. The output sentences of the translation engine were evaluated by a human English-Filipino translator based on perceived translation quality. The evaluator used the following rating scheme: 1 – Unacceptable, 2 – Poor, 3 – Acceptable, 4 – Good, and 5 – Accurate. The evaluator was given 110 English sentences with the system's translation output. The results of the evaluation are found in Table 2.

Table 2: Subjective Sentence Error Rate without Learning.

Rating	Percentage Value
Accurate	0%
Good	11%
Acceptable	16.5%
Poor	45.9%
Unacceptable	26.6%

After training the module with only 30 sentences, the same English sentence were entered into the system and the system's output were evaluated. A substantial increase in quality is found (refer to Table 3) from learning only a few number of sentences during the training phase.

Table 3: Subjective Sentence Error Rate without Learning.

Rating	Percentage Value
Accurate	11%
Good	42.2%
Acceptable	31.2%
Poor	13.8%
Unacceptable	1.8%

7. Future Work

The paper presented an approach of incorporating learning into an MT system in the hope of improving translation quality. Using Filipino as the target language also allows researchers to further understand and formalize an evolving language. Filipino is currently evolving into a combination of the original Filipino and English language. As such, the research allows flexibility by using a bilingual corpus instead of fixed transfer rules. The researchers are currently improving the system to accommodate a larger set of English grammar and to enhance the learning algorithm and evaluation method. Aside from this, to improve quality, semantic analysis will also be included. The next phase of the development would be the translation from Filipino to English. The challenge in this is that in Filipino, pronouns are gender-less. Anaphora resolution algorithms may be incorporated to address this issue.

8. References

- Borra, A., et. al. 1997. IsaWika: A Machine Translation from English to Filipino, A Prototype, University of the Philippines.
- Brown, R., Lavie A., and Levin, L. 2002. Automatic Rule Learning for Resource-Limited MT, [online] available: <http://www.cgi.cs.cmu.edu/~kathrin/amta02CarbonellEtAl.pdf>
- Carbonell, J., Probst, K., Peterson, E., Monson, C., Lavie, A., Brown, R., and Levin, L. 2002. *Automatic Rule Learning for Resource-Limited MT*. In AMTA 2002, Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas, Tiburon, California, 2002.
- Elkan, C. and Kauchak, D. 2002. Learning Rules to Improve a Machine Translation System, [online] available: <http://www.cs.ucsd.edu/~dkauchak/kauchak03ecml.pdf>

- Herve, D. 2002. Learning Rules and Their Exceptions, [online] available: <http://www.ai.mit.edu/projects/jmlr/papers/volume2/dejean02a.pdf>
- Hutchins, J. 2003. Machine Translation and Computer-Based Translation Tools: What's Available and How It's Used, [online] available: <http://ourworld.compuserve.com/homepages/WJHutchins/Valladolid-2003.pdf>
- Menezes, A. and Richardson S. 2000. A best-fit alignment algorithm for automatic extraction of transfer mappings from bilingual corpora, [online] available: <http://acl.upenn.edu.W/W01/W01-1406.pdf>
- Omidvar, O. 2004. Syntax and Rule Based Decoding for Statistical Machine Translation Systems. [online] available: <http://jazz.nist.gov/atpcf/prjbrieffs/prjbrief.cfm?ProjectNumber=00-00-6945>
- Probst, K. 2002. "Semi-Automatic Learning of Transfer Rules for Machine Translation of Low-Density Languages". In Proceedings of the 7th ESLLI student session at the European Summer School in Logic, Language, and Information (ESLLI-02).
- Probst, K. 2003, Automatic Learning of Syntactic Transfer Rules for Machine Translation, [online] available: <http://www.cgi.sc.cmu.edu/People/kathrin/Research/SummaryOfProposal.pdf>

