LFG-Based Machine Translation Engine for English and Filipino

Allan B. Borra, Erwin Andrew O. Chan, Chris Ian R. Lim, Richard Bryan S. Tan, Marlon Cromwell N. Tong
College of Computer Studies
De La Salle University
2401 Taft Ave.
1004 Manila, Philippines
(+632) 524-0402
{borraa, chris.lim, erwin.chan, richard.tan.a, marlon.tong}@dlsu.edu.ph

ABSTRACT
This paper discusses the LFG-based machine translation engine developed for an English-Filipino bi-directional translator. The whole engine includes the analysis to f-structure, transfer of source to target f-structure, and generation from f-structure. Initial linguistic resources were established to test the engine and to develop the full bidirectional English-Filipino machine translator system. These linguistic resources include the formal grammar rules for the English and Filipino language, mono-lingual dictionaries for both languages and the transfer dictionaries, which include transfer rules (structural level) and transfer dictionary (word level). Testing involved subjecting the system to different sentences and sentence constructions in both languages (English and Filipino). Results show that translation quality is extremely dependent on the available linguistic resources.

KEYWORDS
Rule Based, Lexical Functional Grammar, Machine Translation, National Language Processing, Transfer Rules

1. INTRODUCTION
Early attempts to machine translation to Philippine Languages such as IsaWika! [11] established initial computational linguistic resources for subsequent efforts. The formal grammar established from IsaWika! indeed became a stepping stone, as claimed by the authors of said project, to later efforts to machine translation or language technology, in general, using Philippine Languages (specifically, Tagalog). The grammar and the lexicon of IsaWika! was extended and improved in an effort to migrate to Lexical Functional Grammar for a transfer-based English-Filipino machine translation [3]. Theoretical computational linguistics studies of Tagalog [7] proved helpful in coming up with a relatively wider scope of formal Tagalog rules.

These early empirical attempts paved the way for more language technology researches and development of tools, specifically using the LFG formalism. An LFG-based Natural Language Parser Generator was developed [11] that uses Generalized-LR parsing mechanism to address inherent ambiguity, free word order, and multi-modality of structure and words in Tagalog. This became the basis for the analysis phase of the LFG-based Machine Translation engine. FiSSAn (Filipino Syntax Semantics Analyzer) [1] was an earlier system that also employs GLR parsing mechanism, but the lexical entries are static and does not provide mechanism to add more functional schemata. Both empirical works of [3] and [1] implemented inextensible parsers, limited linguistic coverage in terms of sentence constructions; lexicon attributes, as well as, transfer rules in the case of [3].

The LFG-Based Machine Translation engine attempts to address the inextensibility and limited linguistic coverage of earlier empirical efforts. The parser generator for LFG developed by [11], an extensible transfer module, as well as a robust f-structure surface generator, were developed to complete the LFG-based MT Engine.

2. SYSTEM ARCHITECTURE
This section describes the system architecture of the LFG-based MT Engine. The architecture outlines provisions for addressing extensibility as well as mechanisms for conformity to the Lexical Functional Grammar formalism. The main concept of the system is based on the Lexical Functional Transfer (LFT) framework [8]. The LFT is a transfer framework, as seen on Figure 1, which is designed for the LFG formalism.

Figure 1. Translation Model based on Lexical Functional Transfer Framework
2.1. Analysis Phase

The main task of the analysis phase is to convert the input sentence to its corresponding c-structure and then to its f-structure. The output of the analysis phase is the input sentence’s f-structure in XML format. Sentences accepted by the system are dependent on the grammar. The wider the coverage of the grammar, the more types of sentences can be accepted. For the translation process to proceed, the analysis phase must first accept the sentence. In the meantime, once a sentence does not produce any f-structure, then that sentence is not accepted by the system.

![Diagram](image)

Figure 2. NLP Parser Generator Architecture

Reference source not found.

Figure 2 shows the architectural design of the NLP Parser generator developed by [11]. The NLP Parser is mainly created as an extensible parser with some foundations on LFG; it is not yet complete with all the needs of a translation system. The parser was extended in order to output F-Structures that are more ideal for manipulation and analysis by the other modules to complete the translation process. These changes include simplifying the structures in more meaningful ones, and conforming more to the properties offered by the LFG framework.

Figure 3 outlines the analysis phase of the MT engine. Before the translation is started, the file first passes through the preprocessing component, which acts as a file parser. Since the translation needs only strings for processing, only the texts need to be extracted from the different file formats in order to be inputted into the translation process. The tokenizer accepts a string as input and separates them by words. Each punctuation mark is also treated as a separate token. The output of this module is a vector of tokens. The vector of tokens is then passed to analysis module in order for its attributes to be attached to them. These attributes would contain semantic and functional values for each word. These attributes would then be used as the basis for constructing the corresponding C-structure, which represents the syntactical form of the sentence. After a C-structure is made to represent the given sentence, it is now converted to its F-structure equivalent. The attributes of each token are then narrowed down to represent the values that are currently being used in the particular sentence. Also, phrases and clauses are tagged with attributes to describe its syntactic and functional form.

![Diagram](image)

Figure 3. Analysis Phase Architecture

2.2. Transfer Module

The main task of the transfer phase is to map the f-structure of the input sentence to the f-structure of the target output sentence. Transfer rules are used to do this. For the English-Filipino MT system, these transfer rules were hand-crafted through observing the f-structure outputs of the analysis phase for both languages involved, i.e., English and Filipino sentences, and establishing a qualified transfer rule (structurally and lexically) to establish an equivalent f-structure of the target language. These are then manually inputted into the system’s database. The system outputs an XML file that shows the output f-structure of the transfer phase. If no f-structure was outputted, that means there was an error encountered in the transfer phase.

![Figure 4](image)

Figure 4. Sample F-structure from Analysis Phase

The input to the Transfer phase could be seen in Figure 4. Figure 4 is a sample f-structure output from the analysis module which is the output of the Analysis Phase discussed in the previous section. The equivalent f-structure representation used by the system based on the XML format can be seen in Figure 5. The XML tree (f-structure) will be passed on to transfer module to map the structural and lexical information of the source language to the target language using the LFG’s f-structure.

![Figure 5](image)
As shown in Figure 6, the transfer module requires transfer resources such as the monolingual dictionaries for languages involved and the transfer dictionary which includes the structural and lexical bidirectional mappings from one language to another. The corresponding output f-structure would resemble the same format as that of the input f-structure as shown in Figure 7. Output f-structures are also represented using the XML tree format defined for the system.

Figure 6. Transfer Architecture

Figure 5. Sample F-structure in XML Format

Sample transfer dictionary entries are shown in Figure 8. which contains mappings for both structural and lexical levels. A formal language was defined to facilitate creation or addition of transfer rules for both structural and lexical levels. This would allow addition or improvement of more transfer rules making the system more dynamic. Moreover, an interpreter was also constructed to perform the actual mapping process given the formal transfer rules using formal specification or language.

Figure 7. Output F-structure of Transfer Phase

Figure 8. Sample Transfer Rules

A sample transfer rule based on a formal language is shown in Figure 9. For emphasis, the formal specification or language defined allowed extensibility of the system through addition and improvement of existing rules in the database. Although it should be noted that the symbols used in the formal language should be existing or corresponding to symbols of the syntactic rules of the parser, i.e., there is a correspondence of symbols found in the annotations of the context free grammar of the languages involved and the symbols used in the formal transfer rules.
2.3. Generation Phase

The task of the generation phase, as shown in Figure 10, is to generate the target output sentences from the f-structures outputted by the transfer phase. It uses the Hyper Template Planning Language (HTPL) [10] as a representation for f-structures in order to simplify the generation process.

Figure 10. Generation Phase Architecture

The Hyper Template Planning Language (HTPL) provides a formal template from an f-structure and a target formal grammar, i.e., CFG with annotations, which could be interpreted to establish a surface representation of a given sentence.

```
Phrase: LFG
[SUBJ phrase = ]
  [w(det, the), w(noun, Japanese)], the
  [w verb, went,]

[COMPLEMENT phrase = ]
  [w(noun, market)], to

[OBJOFPRREP phrase = ]
  [w(noun, market)], the
```

Figure 11. Sample Generation Template

Figure 11 is the equivalent HTPL representation used in the MT engine leading to the generation of the surface text: “The Japanese went to the market.” While generating the template, the sequencing of the words was established appropriately after matching the input f-structure with the corresponding target grammar.

3. SYSTEM INPUT AND RESOURCES

For the complete LFG-Based Machine Translation Engine to work, the following linguistic resources should exist:

- Formal Grammar for languages involved;
- Monolingual Dictionaries for each language involved; and
- Transfer Dictionaries which include syntactic and/or structural Transfer rules and Lexical Transfer rules.

Some dependencies of these linguistic resources to other resources will be discussed to establish the criticality of these dependencies in making the system work properly.

3.1. Formal Grammar

The sample formal grammar rules showed in Table 1 served as the main input to the NLP Parser Generator developed by [11]. The actual input grammar is defined in a text file (see Figure 12) indicating the terminal symbols first, the nonterminal symbols, and then the production rules that allows annotations.

```
Terminals:
  <terminal 1>
  <terminal 2>
  <terminal 3>
  ...
  <terminal n>
NonTerminals:
  <non-terminal 1>
  <non-terminal 2>
  <non-terminal 3>
  ...
  <non-terminal n>
Start:
  <start symbol>
Rules:
  <rule 1>
  <rule 2>
  <rule 3>
  ...
  <rule n>
**END**
```

Figure 12. Specification Syntax for Grammar Rules

It is imperative that the symbols used in the annotations of the grammar rules must be the same set of symbols allowed to be used in the Transfer Dictionary entries. In other words, there is a dependency between the symbols used in the annotations of the formal grammar and the symbols to be used for the transfer rules. Mismatches in symbols found in either linguistic resource would lead to erroneous translation process.

<table>
<thead>
<tr>
<th>Left Hand Side</th>
<th>Right Hand Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>NP */ SUBJ = *</td>
</tr>
<tr>
<td>IC</td>
<td>IC = VPh2 */= *</td>
</tr>
<tr>
<td>IC</td>
<td>adverb */ QUESTION = *</td>
</tr>
<tr>
<td>VPh2</td>
<td>ADP */ PREVERBMOD = *</td>
</tr>
</tbody>
</table>
The same format or specification of the grammar will be used subsequently for the generation phase, specifically during the generation of the HTPL. The template is constructed by matching the formal grammar of the target language with that of the f-structure output of the Transfer phase. This is another aspect of the dependency of linguistic resources. In this case, the symbols used in the annotations of the formal grammar of both source and target languages, as well as the symbols used in the transfer rules database must be the same. Otherwise, it will lead to failure of translation processes.

3.2. Monolingual Dictionaries

The monolingual dictionaries for languages involved are stored in a database. These dictionaries will contain information such as part of speech, functional schemata and semantics. The morphological analyzer was not implemented for the system. Thus, the word “boy” and “boys” will be two different entries in the dictionary.

An entry in the two monolingual dictionaries must contain the word and the corresponding part of speech of that word. A word in the dictionary must have only one corresponding part of speech. If a word has more than one part of speech, it must be placed as a separate entry.

Each word in the monolingual dictionaries has four properties: an identification number (id), a semantic category, a lexical entry property, and a functional schemata property. The id property serves as the primary key of each word. The lexical entry property, which is a string-type and is the equivalent notation for the extensible functional schemata of lexical entries of LFG, allows addition of lexical semantics on the fly while keeping the LFG-based MT engine intact.

The semantic category property is mainly used in verbs and nouns. It determines which verbs can be performed by which nouns. Each word may have more than one semantic category. If a word has more than one semantic category, each one must be separated by a comma (,).

Ex. human, animal

The functional schemata property checks for the correctness of sentence structures. Each word may have more than one functional schemata. A functional schemata entry must be enclosed inside a parenthesis “( “)”. The values inside the parenthesis are the labels or features than can be found in the grammar.

Ex. (SUBJ)

If there is more than one label or feature in a functional schemata entry, it must be separated by a comma followed by a space.

Ex. (SUBJ, OBJ)

If an entry has more than one functional schemata entry then each functional schemata entry must be separated by a semi-colon (;).

Ex: (SUBJ);(SUBJ, OBJ)

For example, if the verb “eating” has a functional schemata of (SUBJ); (SUBJ, OBJ), it shows that either only a Subject or both Subject and Object is required for the sentence using the verb “eating” to be semantically correct.

The lexical entry property harnesses the descriptive capabilities of the LFG framework. Each word may have more than one lexical entry. A lexical entry has at least one attribute-value pair. The attribute and corresponding value are separated by an equal sign. The attribute can be found in the left side of the equation and the value on the right side.

Example: num=sing.

If an attribute has other labels attached to it, the labels must be separated by adding spaces between them including the attribute itself.

Example: SUBJ HEAD num=sing.

If there is more than one lexical entry, then the lexical entries must be separated by a semi-colon (“;”).

Example: SUBJ HEAD num=sing;SUBJ HEAD pers=3.

3.3 Transfer Dictionary

The transfer dictionary is divided into two databases: the transfer dictionary database and the transfer rules database. The transfer dictionary database contains the words in the monolingual dictionaries that have their corresponding translation/s.

3.2.1. Transfer Dictionary Database

The transfer dictionary is a Look-Up table for the translation of the source-target words. It matches each word with all its possible translations. Each word has a transferWordID. This field is the ID of the word in its specific monolingual dictionary. The lang field determines which dictionary will the word come from. If the value is 1, the word is found in one language dictionary. If the value is 2, the word is found in the other language dictionary.

Words with multiple translations are repeated in the database but with different values for the translation column.

3.2.2. Transfer Rules Database

The transfer rule database contains all transfer rules used in the system. These transfer rules are responsible for the translation of the input f-structure to the output f-structure. Outlined in Table 2, the different fields provided mechanisms to formalize these rules for improvement and addition of more transfer rules. Formalized rules are facilitated by a formal specification of rules in the form of strings.

Table 2. Transfer Rules Database Schema

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Format</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>LanguageRule</td>
<td>the corresponding rule for the English</td>
<td>Text</td>
<td>String</td>
</tr>
</tbody>
</table>

40
4. TESTING AND RESULTS

This section discusses the English and Filipino linguistic resources that were manually created to test the system’s concept. Aside from the linguistic resources, the testing inputs are also described.

4.1. Linguistic Resources Available

The common trait of linguistic resources necessary to make the LFG-based machine translation engine to work was that these were manually created and added to the system. Previous empirical attempts to machine translation provided invaluable linguistic resources that could readily be improved to address more coverage.

4.1.1 Formal Grammar Rules

The system derives its grammar from the grammars used in previous Machine Translation systems. The English grammar was derived from the grammar of the NLP Parser while the Filipino grammar was derived from the grammar of the FiSSAn. Extensions to the grammar were made in order to address more sentence construction types for both languages. But amid these improvements, the grammar input for both languages are still not exhaustive. For a complete listing of the grammar for English and Filipino used in the system, refer to [5].

4.1.2. Monolingual Dictionaries

Both monolingual dictionaries for English and Tagalog were manually populated. Each monolingual dictionary is composed of at least 1,000 entries. The system was fed with English word entries that were based on the 1,000 most commonly used English words [5]. Then, through these English words, 1,000 Filipino translations were also derived. Three online English-Filipino dictionaries were used as a reference for the translation of all English words. The online dictionaries used were Foreignword Tagalog-English Dictionary, Tagalog-English Dictionary, and Gabby’s Dictionary.

4.1.3. Transfer Dictionaries and Rules Databases

The transfer dictionary database which involves word/lexical-level mappings was populated with more than 2,000 bidirectional pairs. The transfer rules database, on the other hand, which maps the structural level for both languages, was populated manually with 15 hand-crafted rules. The rules were based on observing output f-structures and prescribing a target f-structure. This is done for both languages. The complete listing of these transfer rules using the specification syntax defined for the system can be found in [5].

4.2. Testing Inputs

Given the initial set of linguistic resources for English and Tagalog as discussed in the previous section, the LFG-based machine translation engine was tested using 12 Filipino sentence structures and 10 English sentence structures. Different lexical entries were used for the 12 Filipino and 10 English sentence structures. A corpus of 53 sentences was also used to test the output of the sentence. For the complete details of the sentences used for testing, refer to [5].

4.3. Results

It is observed that after subjecting the test inputs to the system, successful translation can be achieved if three requirements are met: 1. both input and output sentence constructions must be accepted and generated by the grammar; 2. words must already exist in the dictionary; and 3. transfer rules must exist for a given f-structure. This would reaffirm common trait of rule-based machine translators with regards to their very high dependency to the linguistic resources available.

5. CONCLUSION AND RECOMMENDATION

An extensible LFG-based machine translation engine was developed that can improve by adding and improving the linguistic resources it uses. Thus, as the quality of linguistic resources gets better, so will the quality of translation produced by engine. In the same manner, a small set of linguistic resources would also render the engine unable to complete the translation process.

The initial set of linguistic resources for the English-Filipino translator established for the system showed that the LFG-based machine translation engine works properly as long as the linguistic resources covers the input sentences. Continuous update and improvement of the linguistic resources: 1. the formal grammar for languages involved; 2. the monolingual dictionaries with ample and appropriate semantic information; and 3. the transfer dictionaries, would definitely lead to more robust and better quality of translation. Tools can be created so that it would facilitate additions and improvements of these linguistic resources. Equally important is training of users that would adhere to
the LFG formalism and conform to the specification syntax defined for the system’s linguistic resources. The advocacy to a particular formalism would not render the LFG-based machine translation engine useless.

6. BIBLIOGRAPHY


