ANALOGICAL LEARNER FOR NATURAL LANGUAGE PROCESSING BASED ON STRUCTURED STRING-TREE CORRESPONDENCE (SSTC) AND CASE-BASED REASONING

LIM HUAN NGEE

UNIVERSITI SAINS MALAYSIA
2009

ANALOGICAL LEARNER FOR NATURAL LANGUAGE PROCESSING BASED ON STRUCTURED STRING-TREE CORRESPONDENCE (SSTC) AND CASE-BASED REASONING

by

LIM HUAN NGEE

Thesis submitted in fulfillment of the requirements for the Degree of Master of Science

MAY 2009

ACKNOWLEDGEMENTS

I would like to take this opportunity to show my greatest gratitude to my supervisor Dr. Tang Enya Kong for all the help and advice which he has given throughout the whole period of my research. He has taken great effort to explain everything clearly to me and always tried his best to help me whenever I faced any problem during my research. Without his help, I would not be able to finish my research in the stipulated time.

I am also thankful to Dr. Chan Huah Yong for helping me with the important forms which are needed to be submitted to IPS for approval and processed. He has also given me a lot of advice on my thesis which I really appreciate.

Throughout my research period, there are many other people who have given me help and guidance which I greatly appreciate. I would to Mr. Ye Hong Hoe and Ms. Lim Chai Kim, research officers of UTMK for explaining to me the overall workflow of our EBMT engine patiently and provided me with the material which I needed for my research. I would like to thank Ms. Lim Muk Moi, another research other of UTMK who has provided me with all the BKB files for my testing purpose. With their help, I was able to complete my research smoothly.

I would like to thank my family for giving me their moral support. Lastly, I would like to thank my husband for his support and helped me get through the difficult times. He had also offered me a lot of ideas which are useful for my research.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BKB	Bilingual Knowledge Bank
EBMT	Example-Based Machine Translation
POS	Part-Of-Speech
SSTC	Structured String -Tree Correspondence
S-SSTC	Synchronous Structure String-Tree Correspondence
FDG	Functional Dependency Grammar
KIMD	Kamus Inggeris-Melayu Dewan

PEMBELAJARAN ANALOGI UNTUK PEMPROSESAN BAHASA TABII BERDASARKAN "STRUCTURED STRING-TREE CORRESPONDENCE" (SSTC) DAN KAEDAH BERASASKAN CONTOH

ABSTRAK

Mesin terjemahan melalui contoh menggunakan contoh penterjemahan yang seiras yang didapati daripada bank pengetahuan dua bahasa (BKB). Contoh-contoh (pasangan sumber dan sasaran) di dalam BKB dianotasikan dalam struktur yang fleksibel yang dikenali sebagai 'Structured string-tree correspondence' segerak (S-SSTC).

Pendekatan melalui pengindeksan telah dilaksanakan dalam EBMT Bahasa Malaysia-Bahasa Inggeris untuk memberikan liputan yang baik bagi teks masuk dan meningkatkan ketepatan struktur penterjemahan. Pasangan contoh sumber dan sasaran di dalam BKB diindeks dalam peringkat perkataan dan struktur. Indeks struktur diklasifikasikan mengikut jenis dan struktur.

Kaedah analogi di perkenalkan kepada sistem EBMT untuk meningkatkan ketepatan terjemhan. Dengan kaedah analogi, kita dapat mengenalpasti contoh-contoh BKB yang lebih bersesuaian dengan ayat masuk yang diberikan. Daripada contoh-contoh itu, kita dapat menerbitkan sebanyak mungkin templat dengan menggunakan perkadaran analogi. Templat-templat ini mempunyai struktur yang berkaitrapat dengan ayat masuk berbanding dengan indeks struktur yang dipulangkan oleh kaedah semasa kerana indeks struktur dipilih berdasarkan beberapa kriteria yang ditetapkan oleh penyelidik.

Selepas penerbitan templat-templat, kami membinakan perwakilan pokok bagi setiap templat dengan menggunakan kaedah pertimbangan melalui contoh. Tujuan pembinaan perwakilan pokok adalah untuk mengesahkan templat-templat yang diterbitkan. Setiap templat mesti berselaras dengan perwakilan pokoknya.

Kita telah membuatkan satu perbandingan antara kaedah analogi dengan kaedah pengindeksan struktur dari segi ketepatan terjemahan dan keputusan penilaian telah menunjukkan bahawa kaedah kita telah mencapai keputusan yang lebih baik berbandingkan dengan kaedah pengindeksan struktur.

ANALOGICAL LEARNER FOR NATURAL LANGUAGE PROCESSING BASED ON STRUCTURED STRING-TREE CORRESPONDENCE (SSTC) AND CASE-BASED REASONING

ABSTRACT

Example-Based Machine Translation (EBMT) is using the similar translation examples which are retrieved from the Bilingual Knowledge Bank (BKB) to translate an input sentence. The examples (source and target pairs) in the BKB are annotated based on a flexible annotation schema known as Synchronous Structured String-Tree Correspondence (S-SSTC).

Indexing approach has been implemented into our current English-Malay EBMT to ensure fast retrieval of appropriate examples in the BKB for EBMT to produce well-formed translations. The source and target example pairs in the BKB are indexed in word and structure level. The structural indexes are classified according to different types and structures of examples.

Analogy method is introduced to the EBMT system to increase the accuracy of translation. Using analogy method, we can identify more appropriate BKB examples for a given input sentence. From the examples, we derive as many templates as possible using analogy proportion. These templates are more structurally related to the input sentence compared to the structural indexes return by the current approach because the structural indexes are picked based on certain criteria fixed by the researcher.

After the derivation of the templates, we construct its tree representations using case-based reasoning method. The purpose of constructing tree

representations is to validate the templates which we have derived. Each template must correspond to its tree representation.

We have made a comparison between analogy method and structural indexing approach in term of accuracy of translations and the evaluation results shown that our new approach achieves better results than existing approach.

CHAPTER 1
INTRODUCTION

Analogy method has successfully applied to many Natural Language Processing (NLP)[1] tasks like morphological analysis, part of the speech tagging and many more. Therefore, we will look into how analogy method can be applied to our current Example-Based Machine Translation (EBMT) systems. In this chapter, an overview of EBMT is given, followed by the objectives of applying analogy method to our EBMT system and also the outline of the following chapters.

1.1 A General Overview of Example-Based Machine Translation

Example-Based Machine Translation (EBMT) was first proposed by Nagao Makoto in 1984. The idea of EBMT is to translate a sentence into another target language sentence based on similar translation examples stored in a database.

According to Sumita and Iida, EBMT retrieves similar examples like pair of source and target phrases, sentences or texts from a database of examples and adapt the examples to translate a new input sentence.

A sentence can decomposed into a certain fragmental substrings and they are translated to its target language substrings. The target language substrings are then composed into a complete target language sentence of the source sentence.

The translation examples are collected from parallel corpus which contains sentence pairs like English $\leftrightarrow$ Japanese or English $\leftrightarrow$ Malay or any other language pairs must be aligned before they are used for translation.

[1] Natural Language Processing is computational linguistics.

1.2 Research Objectives

Our English-Malay Example-Base Machine Translation is currently using indexing approach to produce more accurate translation. The examples in the database are indexed in word level and structural level. The structure of the examples in the Bilingual Knowledge Bank (BKB) is indexed and classified into different types like fully lexicalized, partially generalized and fully generalized which will be discussed in the following chapter.

Though indexing approach has increased the accuracy and well-formedness of the translation but it is not as accurate as it should be. It is because the chosen indexes from the database might not be necessary a suitable translation pattern for the input sentence.It is because the criteria to choose the best structural indexes are based on the lowest deepness of a tree followed by longest chunk next most lexicalized and lastly highest frequency. Therefore, it might cause unexpected translation results at times.

The purpose of applying analogy method is to improve the accuracy of the translation in the EBMT. Using analogy method, we try to extract more relevant examples from the bilingual knowledge bank (BKB) based on the given input sentence. From the examples that we extract, we try to derive as many templates as possible for the given input sentence. These derived templates are structurally similar to the given input sentence.

The new derived examples will also be in Part-Of-Speech (POS) tag like “PRON V N ING”. For easier understanding, a new derived example is known as analogy template and this term will be use throughout the explanation. Then, the tree representation of each derived analogy template is constructed using case-

based reasoning method. The constructetd tree representation is encoded with SNODE and STREE which corresponds to the analogy template.

There might be more than 1 analogy templates derived for each input sentence which might cover different segment of the input sentence. An analogy template might cover the whole input sentence structure or part of the input sentence structure. Therefore, it can provide wider coverage for a given input sentence which directly increases the well-formedness of translation.

In this research, we will only be applying analogy method in source portions. We make use of the existing source SSTC examples and structural indexes in the BKB which are related to the input sentence.

1.3 Thesis Outline

There are altogether seven chapters in this thesis. Chapter one is divided into two sections. The first section is an overview of EBMT followed by the objective of applying analogy method to current EBMT system.

In the next chapter, we discuss in details analogy method from its underlying concepts to the techniques use in analogy method. We will also touch on some other concepts like SSTC, S-SSTC, Structural Indexing Approach and Case-Based Reasoning.

Chapter three provides the main part of this thesis. We will discuss on the methodology of this research. In chapter four, we will look into the implementation of of the methodology.

In chapter five, we will simulate an example of implementation for the overall process with translation. For chapter six, we will look into the experiment and results which we obtain from the implementation of the analogy method to EBMT system. We will compare the results which we get using analogy method with the previous work using structural indexing approach.

Lastly, in chapter seven, we have a thorough discussion on some of the future work which still can be done to improve it.

CHAPTER 2
BACKGROUND

Word-base indexing was introduced by Al-Adhaileh (2002) but due to some weaknesses in his work, structural indexing method was introduced by Ye (2006) to overcome the weaknesses which is currently used in the Example-Base Machine Translation (EBMT). Though it does improve the translation in certain level but the accuracy of the translation is not optimal.

Therefore, a study is done on the analogy method to be applied in the current EBMT but at the same time make use of the template created using structural indexing method which was introduced by Ye (2006).

Firstly, we will give a brief overview of structured string-tree correspondence (SSTC) which is used in the current Bilingual Knowledge Bank (BKB) and also some background on word indexing and structural indexing which is used in our current EBMT. Lastly, we carried out a literature survey on analogy method and some of the works using analogy method.

2.1 Structured String-Tree Correspondence (SSTC)

Structured string-tree correspondence (SSTC) was first introduced by Boitet and Zaharin (1988) to overcome the problem of non-projective like featurisation, lexicalization and crossed dependencies between language string with its representation. SSTC separates language string from its representation tree which can be seen in figure 2.1.

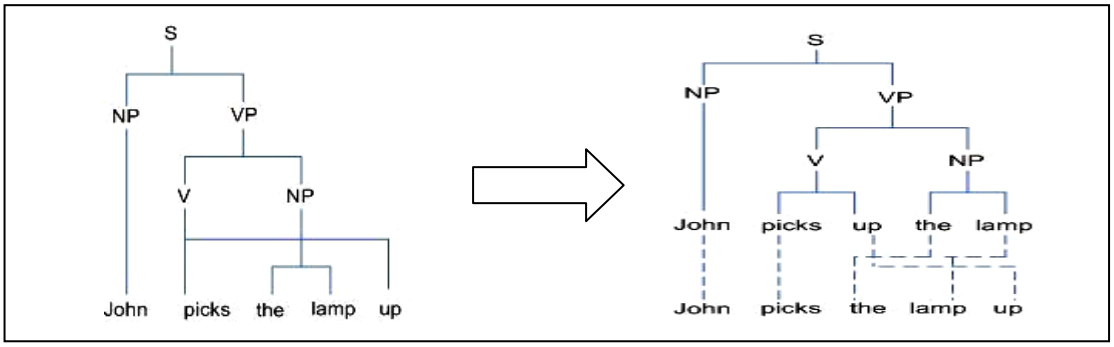

Figure 2.1: One of the examples of cross-dependency and how the string "**John picks the lamp up**" separates from its non-contiguous phrase structure tree modified from (Boilet & Zaharin, 1988)

SSTC is a flexible annotation schema which describes a sentence, a representation tree and the correspondence between substrings in the sentence and subtrees in the representation tree. SSTC correspondence consists of two interrelated correspondence where one is between nodes and substrings and the other one is between subtrees and substrings. The correspondence in SSTC is denoted in a pair of intervals X/Y. It is tied to each node in the representation tree. X which is also known as $X(\text{SNODE})$ denotes the interval containing the substring that corresponds to the node, where Y which is also known as $Y(\text{STREE})$ denotes the interval containing the substrings that corresponds to the substree having the node as root. (Tang and Al-Adhaileh, 2002)

Here is an example of a sentence which is annotated in SSTC format. Each word in the sentence "John kicks the ball", is assigned with an interval starting from (0-1) for "John", (1-2) for "kicks", (2-3) for "the", (3-4) for "lamp" and (4-5) for the punctuation ".". Each tree node of the tree representation for this sentence is encoded with SNODE and STREE. For example, the node "John" with SNODE interval (0-1) corresponds to the word "John" in the sentence which has the same interval (0-1). Therefore, it is written in (0-1/0-1). Figure 2.2 illustrates the sentence "John kicks the ball." annotated in SSTC format.

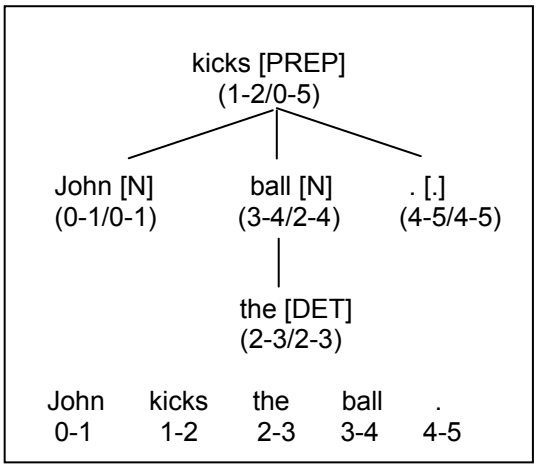

Figure 2.2: Sentence *"John kicks the ball"* annotated in SSTC format

2.2 Synchronous Structured String-Tree Correspondence (S-SSTC)

Al-Aldhaileh and Tang (2002) have proposed a flexible annotation schema which is known as synchronous structured string-tree correspondence (S-SSTC) which can handle some non-standard correspondence cases in translation. S-SSTC contains of a pair of SSTCs with an additional synchronization between them.

It relates expressions of a natural language to its associated translation in another language which we call the two languages source and target languages. The synchronous correspondence is denoted in terms of SNODE pairs and STREE pairs.

Figure 2.3 shows an S-SSTC example for the English sentence "He goes to library" with its target sentence "Dia pergi ke perpustakaan". The arrow in the figure indicates correspondence between source and target SSTCs.

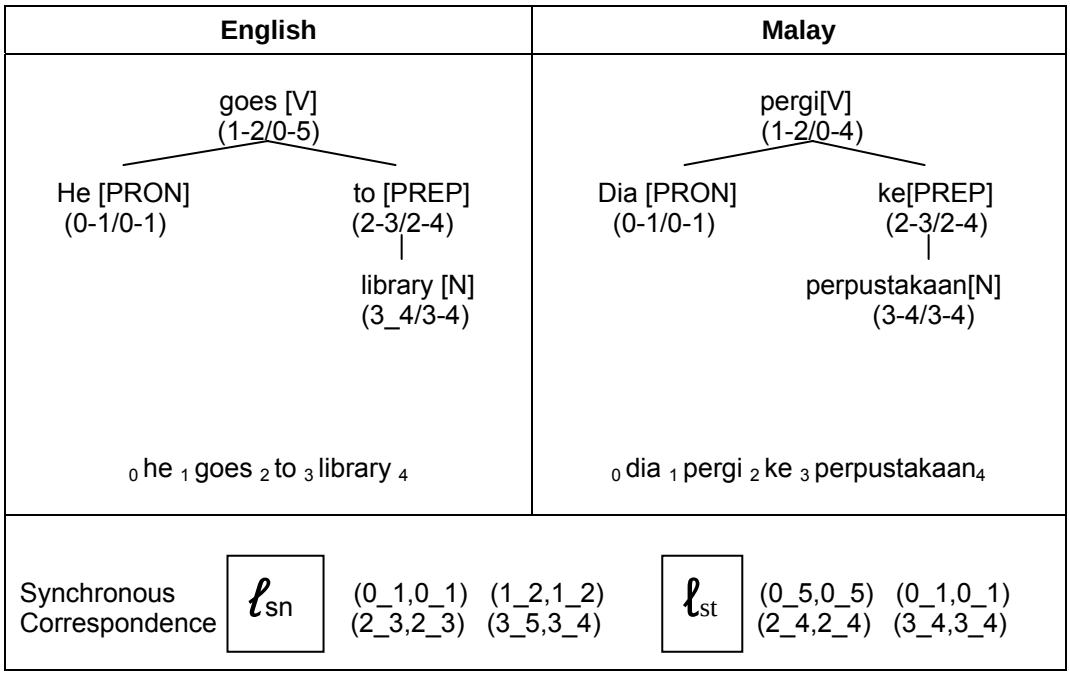

Figure 2.3: An example of S-SSTC for the English sentence "He goes to library".

2.3 Indexing in Bilingual Knowledge Bank (BKB)

Before we go into analogy method, we will look into two types of indexing in our current BKB which are word indexing and structural indexing.

2.3.1 Word Indexing

Word indexing was introduced by Al-Adhaileh (2002) to the current English-Malay Example-Based Machine Translation (EBMT) where it not only handles one-to-one mapping but also one-to-many mapping e.g. "across" → "di seberang" and many-to-one mapping e.g. "pick up" → "mengutip". It is because some words cannot be separated as individual word else the meaning of words will be loss. This can be seen in figure 2.4.

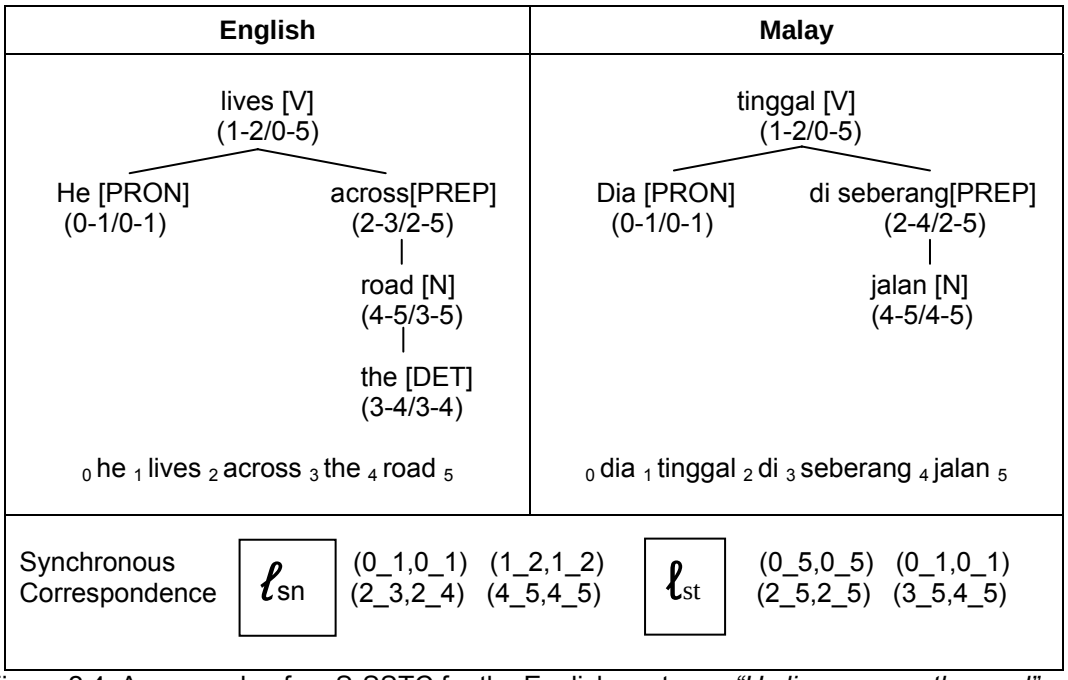

Figure 2.4: An example of an S-SSTC for the English sentence *"He lives across the road"* and its translation in Malay *"Dia tinggal di seberang jalan"*

Though word indexing is flexible but it fails to select words from structurally similar examples. Therefore, Ye (2006) proposed structural indexing to solve this problem and also to improve the well-formness of the translation.

2.3.2 Structural Indexing

Ye (2006) have classified the structural indexing according to different level of generalization which are fully lexicalized, partially generalized and fully generalized which also known as transfer rule. Figure 2.5 illustrates the different level of generalization modified from Ye (2006).

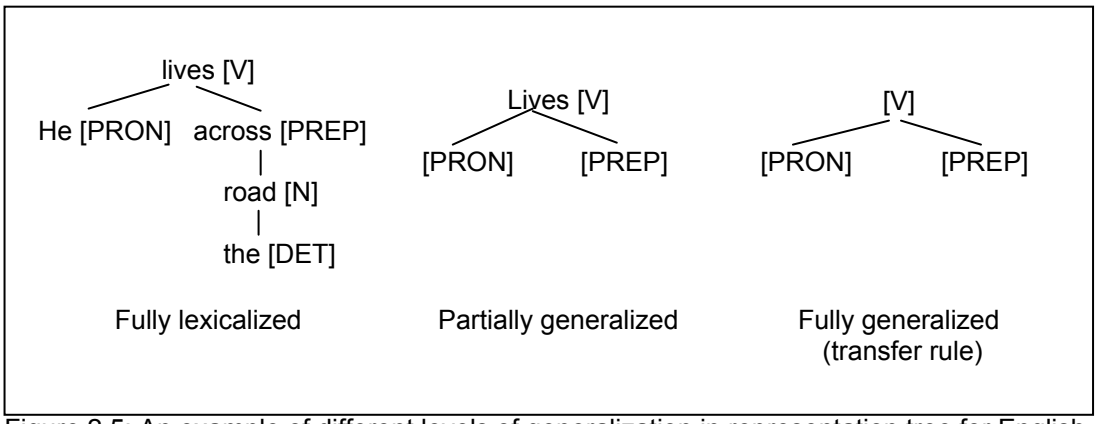

Figure 2.5: An example of different levels of generalization in representation tree for English sentence "He lives across the road" in BKB

Fully lexicalized consists of indexing for source phrases or sentences which can be considered as phrasal index. According to Ye (2006), fully lexicalized sub-examples can be built from subtree correspondences recorded in S-SSTCs. Note that, a subtree which consists of single node cannot be considered as fully lexicalized sub-example because it has been considered under word indexing.

Ye (2006) has divided partially generalized into two type of templates which he named it template type 1 and template type 2. Template type 1 has only one level deep representation of tree where it is divided into different structure; root, intermediate and terminal. Template type 1 terminal nodes contain only POSs. Figure 2.6 shows the different structure of template type 1 extracted from Ye (2006).

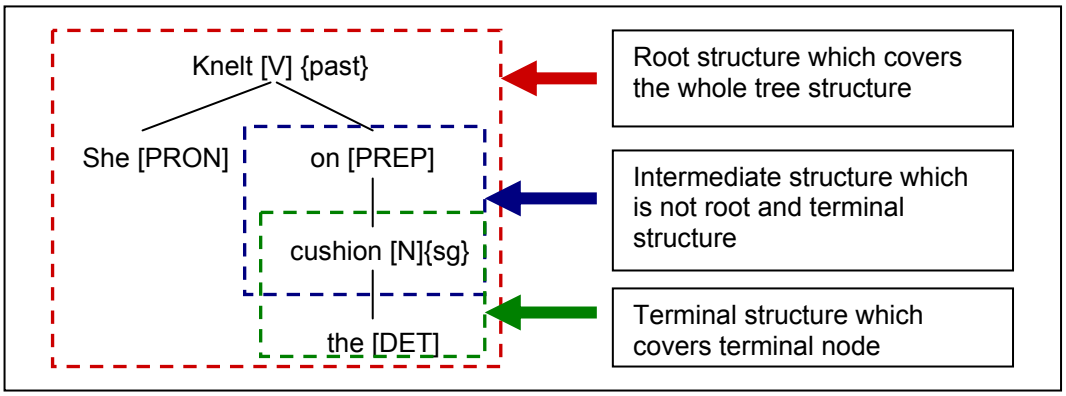

Figure 2.6: Shows an example of different type of structures from the sentence *"She knelt on the cushion"* for template type 1 extracted from Ye (2006)

As for template type II, it extends the context until content word like noun and its tree representation is usually two levels deep. It is organized into two different structure; root and intermediate. Template type II helps in choosing the right preposition in translation process because one preposition in English language can be translated into more than possibilities in Malay language as stated by Ye (2006). Figure 2.7 shows the two different structures in template type II adapted from Ye's thesis.

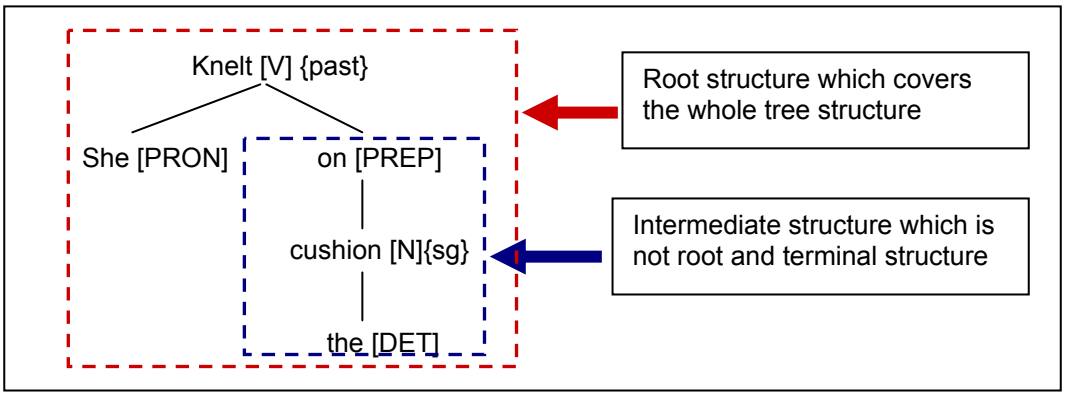

Figure 2.7: Two different structures of template type II from the sentence *"she knelt on the cushion"*

Fully generalized or transfer rule is actually a rule index. Every node in its tree representation only contains POS. The procedure of extracting the rules is similar to template type I. It also contains three different structures; root, intermediate and terminal. See figure 2.8 of some fully generalized examples which also extracted from Ye (2006).

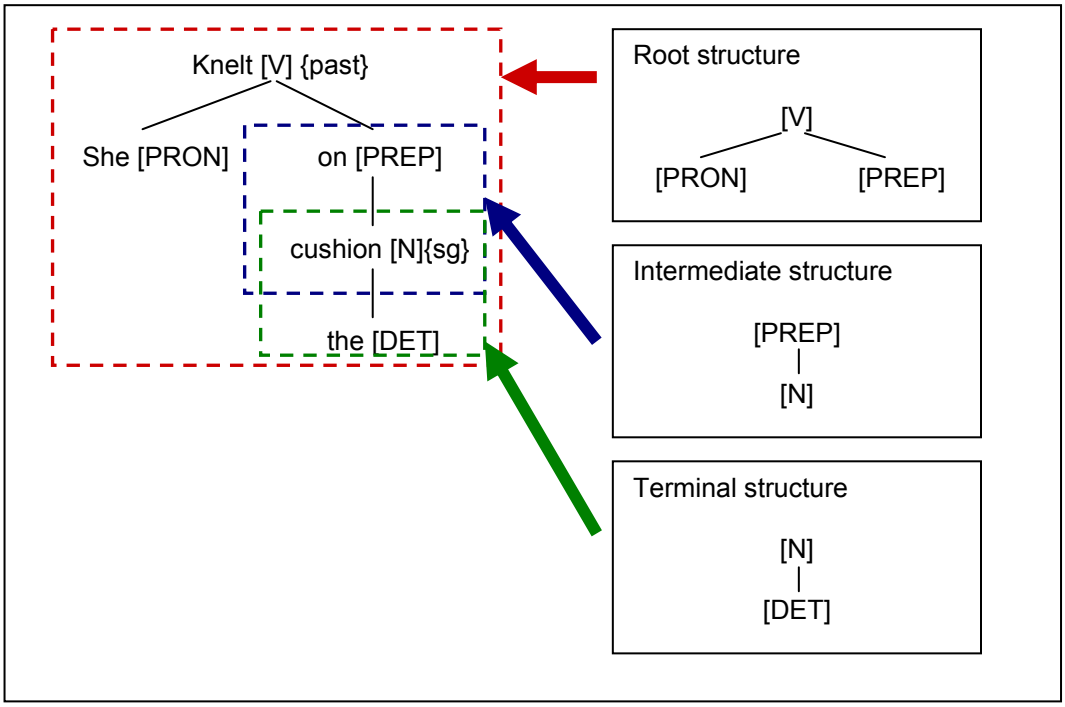

Figure 2.8: Different structures extracted from the sentence “She knelt on the cushion” for transfer rules.

2.4 General Understanding of Analogy Method

In this section, we discuss how the analogy method comes about and a detail explanation of analogy will be given followed by the example of works on analogy.

2.4.1 Foundations of Analogy

Analogy has been studied and discussed by philosophers like Aristotle and Plato and has been applied to many fields like science, law, mathematics and linguistics.

Analogy is a process of transferring information from a particular subject to another particular subject particular by deduction, induction, and abduction, In short, Analogy relates the relationship between two ordered pairs.

Analogy proportion or analogy equation involves 4 elements where the forth element is coined from other three elements. It is expressed as followed: “A is to B as C is to D” since as far in the past as Euclid, Aristotle and isdenoted in this format: A : B :: C : D where D is the forth element which is derived from A, B and C.

Pirrelli and Yvon, 1989 stated that analogy is not an inherent relationship between any two terms but a recurrent proportionality between two series of terms. It involves known objects which are used to infer the missing features. Hence, it can be defined in term of “is to” and “as” relationships and identified in a formal analogical proportion[2].

According to Stroppa and Yvon, 2005, an analogical proportion is a relation involving four elements which are labeled as A, B, C and D in a set of object, X. Proportional analogies have the property of the exchange of the means. Therefore, it allows us to take the four elements (A, B, C and D) apart and form several smaller fragments.

Analogy is actually based on two steps inference processes which are computation of a structural mapping between a new and a memorized situation and transfer of knowledge from the known to the unknown situation (Stroppa and Yvon, 2005). Analogical learning is applicable for parsing and/or example-based machine translation task. It matches and transfers based on a perception which emerges from the analysis of the problem. Analogical learning investigates all possible combinations matching from best case to worse case situation.

[2] These proportions correspond to the Aristotelian [Aristotle] notion of Analogy.

Lepage (1998) has stated that as the examples are arranged in analogical proportion format A : B :: C : D where D is the derived results. Therefore, D is form by going through sentences B and C one element at a time and inspecting the relations of each element to the structure of sentence A. In another word, it looks for the portions which are uncommon between sentence A and sentence B and uncommon portions between sentence A and sentence C and combines the uncommon portions found in the right order.

In short for analogy method, a given situation is understood by comparison with another similar situation. Therefore, analogy method can be used to guide reasoning, to generate conjectures about an unfamiliar domain, or to generalize several experiences into an abstract schema.

Analogy concept can also be represented in finite sets as written by Yvon, Stroppa, Delhay and Miclet (2004). A, B, C and D are known as four sets in X. The analogical relationship A : B :: C : D can be defined as $A \cup D = B \cup C$ which is also equivalent to $D = ((B \cup C) \setminus A) \cup (B \cap C)$. It can be presented in Venn picture like figure 2.9.

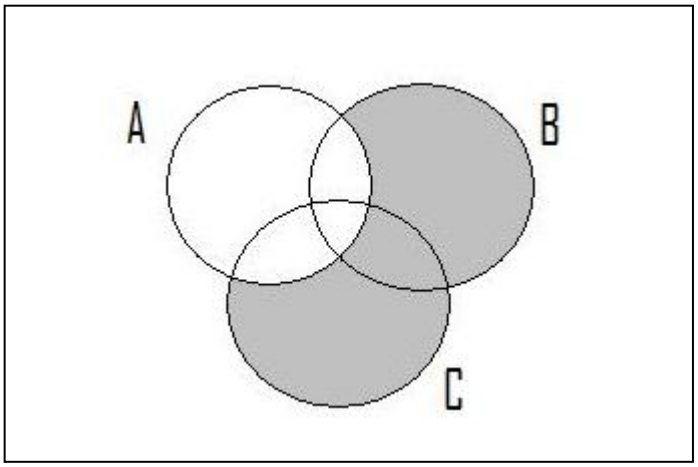

Figure 2.9: Venn picture to represent the analogical relationship A : B :: C : D.

In the next section, we will discuss the theoretical framework for analogy method which is based on Genter's structural mapping theory of analogy.

2.4.2 Structural Mapping Theory of Analogy Method

According to Genter's (1983) structural mapping theory of analogy, it asserts that an analogy is the application of a relational structure that normally applies in one knowledge domain (the base domain) to another, different, knowledge domain (the target domain); unlike less-structural psychological theories, it also sees analogy and similarity as connected processes.

It finds a list of similarities examples where it consists of pair wise matches between the *base* and the *target*, and returns a set of directed mappings between them. The selected lists of examples are calculated based on the edit distance calculation method which will be discussed in the next sub-section.

Overall, structural mapping decomposes the analogical processing into threes stages; the first stage consists of retrieving the set of examples which is analogous or similar to the given current situation or known as input situation. Secondly, the construction of mapping which consists of correspondence between the base and target based on the set of retrieved examples. These are the candidate inferences sanctioned by the analogy. Lastly, estimate the "quality" of the match which involves three kinds of criterion; structural similarity, the validity of the match and lastly whether the analogy is useful to the reasoner's current purposes.

In the following subsection, we will discuss on the edit distance calculation algorithm which is used to calculate each and every examples' distance in order to select the list of best examples which are "qualified" to be used to proceed to the next analogy process.

a. Edit Distance Calculation Algorithm

Edit distance algorithm or known as Levensthein distance was introduced by Vladimir, 1966 for measuring the amount of difference between two sequences. It computes the minimum number of required editing actions in order to transform one sequence into another through an inverse backtracking procedure. The final similarity score is computed based on the algorithm.

Sequences are analyzed and encoded to two dimensional vectors based on the characters. Then the sequence vectors are compared on an equal – not equal basis through the edit distance algorithm. Once all the comparisons are done, the last computed number is the value of the edit distance. The smaller the edit distance, the better it is because as the distance is small, it means that it takes minimal effort to transform a situation to another situation and also has the more similarities.

For instance, the following word examples in figure 2.10 adapted from Yves Lepage, the distance between “like” and “unlike” and distance between “like” and “known” are dist(like, unlike)=2 and dist(like, known)=5. It shows that “like” have more similarities with “unlike” compared to with “known” and it is easier to transform “like” to “unlike”.

	U	N	L	I	K	E			K	N	O	W	N
L	1	2	2	3	4	5		L	1	2	3	4	5
I	2	2	3	2	3	4		I	2	2	3	4	5
K	3	3	3	3	2	3		K	2	3	3	4	5
E	4	4	4	4	3	2		E	3	3	4	4	5

Figure 2.10: The matrices give the distance between “like” and “unlike” and between “like” and “known” with the value of circled in red.

2.5 Survey of Works in Analogy Method

As we know that, analogy is denoted in A: B:: C: D and D is the value we want to search for in order to form the relationship between A, B, C and D. The method will thus to look for those parts which are not common to A and B on one hand and not common to A and C on the other and put them together in the right order to form the value of D. This method has been applied in words, sentences and even on trees by most of the researchers.

In the next section, we will look into some of the works which have applied analogy method to them.

2.5.1 Analogy on Words

According to Lepage (1998), analogy in linguistic works is defined as an operation by which given two forms of a given word, and only one form of a second word, the missing form is coined.

One of the examples in analogy on words is shown in (Pirrelli and Yvon, 1999) where the past tense of "stink" could be guessed by knowing the past tense of the verb "drink" is to "drank" which is abbreviated as in this format: "drink : drank :: stink : stank".

The word "stank" is formed by going through the word "drink" and "drank" one element at a time and inspecting the relations of each element to the structure of "stink". The uncommon word part found between the words "drink" and "drank" are extracted like "ank" found from the word "drank". Next, "drink" is compared with the word "stink". The uncommon portions "st" extracted from the word "stink" are combined with the word "ank" which was found previously in the right order ("st" + "ank").

Drink : Drank = Stink : x $\Rightarrow$ x = Stank

Using A, B, C and D as general terms to represent the example “drink”, “drank”, “stink” and “stank”, an analogical proportion exists between A, B, C and D if and only if A and B on the one hand and C and D in the other hand are perceived as similar and if there exists an isomorphism[3] between the operators generating A and B and the operators generating C and D.

2.5.2 Analogy on Sentences

Analogy principle has been applied in sentences for translation purpose (Lepage, 2005). As proportional analogies have the property of the exchange of the means, therefore, this allows the languages to be taken apart. The translation relation is established through the verification of the analogy relations independently in each language and the translation correspondence between each corresponding term in the analogies.

The process of analogy on sentences basically is the same as analogy on word. A sentence is a string of words which consists of non empty left and right context. Therefore, a sentence’s left and right context is taken into consideration during analogy process. Like word, each word in the sentence is taken apart for comparison during analogy. The uncommon portions found are extracted and combined to form a new sentence like figure 2.11.

[3] Isomorphism applies when two complex structures can be mapped onto each other, in such a way that to each part of one structure there is a corresponding part in the other structure, where "corresponding" means that the two parts play similar roles in their respective structures.

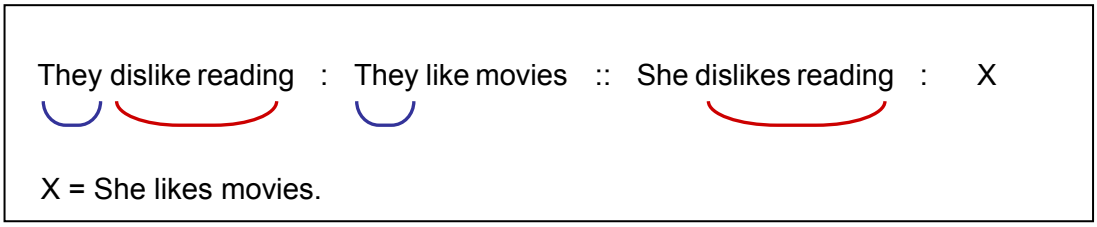

Figure 2.11: An example of analogy on sentence. The common portions are removed and uncommon portions found are combined to form sentence X.

When the sentences are put in analogy format, they are analogical proportion where the two pair of sentences bears the same relationship as the two sentences of the other pair.

2.5.3 Analogy on Trees

Trees are very common structures to represent syntactic structures or terms in a logical representation of sentence. The definition of proportions between trees is quite similar to the one used for words which involves the associative binary operation between trees and the notion of alternating subtrees.

According to Stroppa and Yvon, 2005, to express the definition of analogical proportion between trees, the notion of substitution is introduced. A single substitution is a pair of variable and tree. The application of the substitution to a tree consists in replacing each leaf by the tree.

An example of such tree proportion is illustrated in figure 2.12 with syntactic parse trees which is adapted from Stroppa and Yvon (2005).

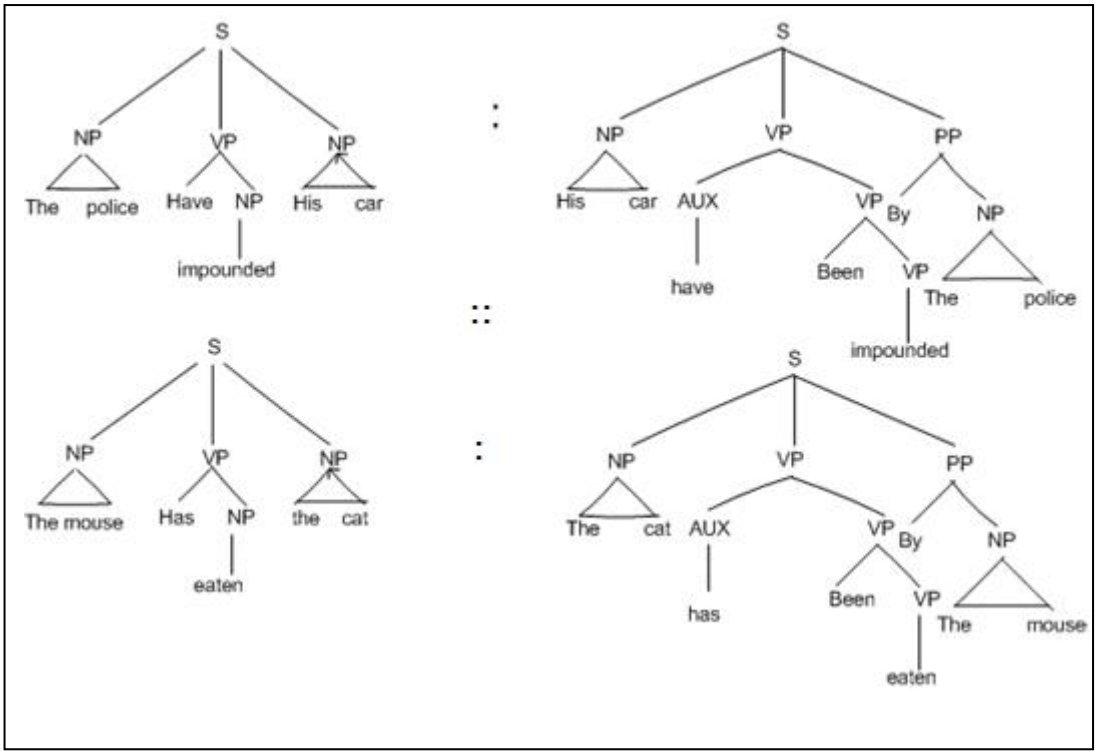

Figure 2.12: An example of analogical proportion for tree adapted from Stroppa and Yvon, (2005).

2.6 Case-Based Reasoning Concept

Case-Based Reasoning (CBR) is actually a problem-solving methodology where a new case is solved by referring to previous case which is most similar to it. The previous case is used as the model for the new case where ithe previous case's solution is adapted to form the new case's solution.

According to Aamodt and Plaza (1994), CBR consists of 4 processes. Firstly, it is the retrieval of most similar cases followed by adaptation of information or knowledge from the retrieved cases. Next, revise the proposed solution and lastly, store the new case with its solution for future problem solving.

Figure 2.13 illustrates the CBR cycle modified from Somers (2001).

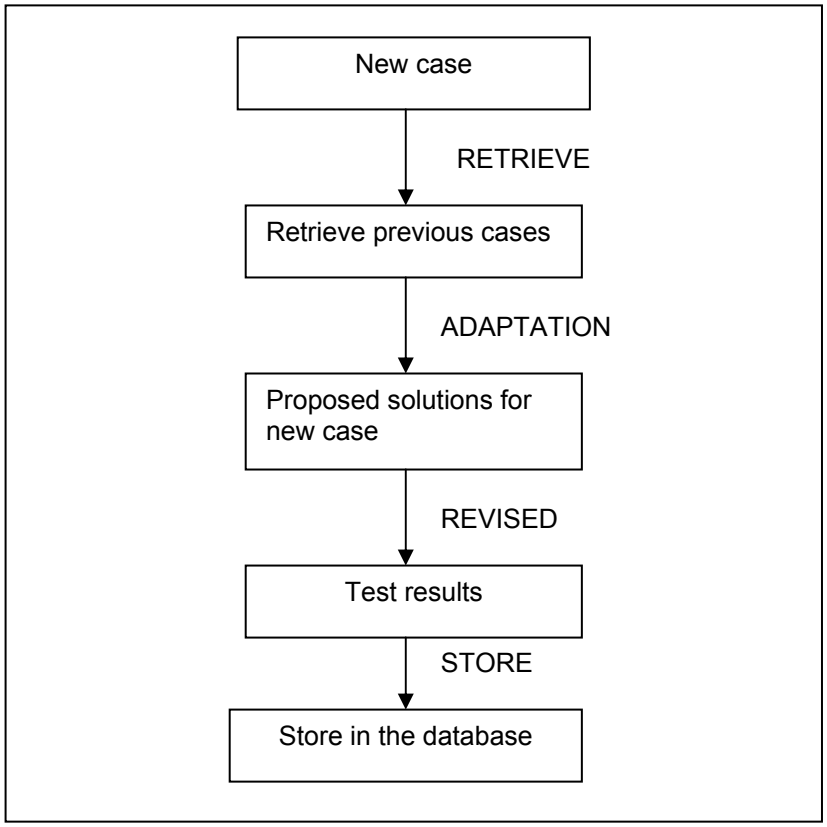

Figure 2.13: Case-Based Reasoning cycles modified from Somers (2001)

2.7 Summary

Overall, we have understood the concepts of analogy method and also case-based reasoning which are useful for our work.

Firstly, we have to make use of the edit distance algorithm for the retrieval of examples which have the closely related structure with the input sentence. Next we need to derive a new templates based on the retrieved examples from the BKB.

After the derivation of the new templates, we construct its tree representation using case-based reasoning methodology.

In the next chapter, we will discuss on how analogy method can help to improve our current EBMT by extending our previous researcher Ye's work (2006).

CHAPTER 3
METHODOLOGY

In this chapter, we will discuss on the methodology of this research before going into details of the implementation of the methodology into the EBMT system.

3.1 Research Methodology

The concept of analogy on SSTC is the same as analogy on words/sentences. The only different is that SSTC is using part-of-speech (POS) sequence. Here, we will briefly simulate an example of analogy on SSTC before going into the details of the process of applying analogy method to our EBMT system.

Firstly, a given input sentence "She likes to eat apple and orange." is parsed into individual lexical units using FDG[4] parser. The words which are parsed into individual word are not taken into consideration of its tenses. We only make use of its root word; for example, "likes" we only use its root word "like" Each of the root word in the input sentence acts as a key to retrieve its own set of best templates from indexed BKB. But in this example, we only use the word “like” to simulate the whole process of analogy on SSTC.

Firstly, we retrieve all templates which consist of the word “like” from indexed BKB. As only 3 best templates are needed for each analogy process, we select the best template based on edit distance algorithm.

The 3 best templates which we found from indexed BKB for the word “like” are:

[4] Functional Dependency Grammar parser

(T.i) N like N N

(T.ii) PRON like AU_INF V

(T.iii) N like V

From (T.i), (T.ii) and (T.iii) templates, we retrieve its SSTC source example from the BKB. These SSTC source examples are the full contents of the best templates.

(S.i) Jerry likes sweets and snacks

(S.ii) She likes to sleep in the afternoon

(S.iii) Jenny likes to read

Then the SSTC source examples (S.i), (S.ii), (S.iii) are converted to its POS sequence:

(P.i) N V N CC N

(P.ii) PRON V AU_INF V PREP DET N

(P.iii) N V AU_INF V

The given input sentence is also converted to POS sequence.

(I) She likes to eat apple and orange

(P) PRON V AU_INF V N CC N

Each SSTC pos sequence are matched against input sentence POS sequence to retrieve longest matching chunk of POS sequence generated from SSTC source example. The longest matching chunks which we get from the POS sequence of SSTC examples (P.i), (P.ii) and (P.iii) are:

(C.i) V N CC N

(C.ii) PRON V AU_INF V

(C.iii) V AU_INF V

The 3 longest matching chunk which we get from SSTC pos sequence are used for analogy process. We call these chunks as sub-examples for easier understanding. From these sub-examples, we derive the forth POS sequence using analogy method. Like analogy method, the POS sequences are arranged in analogy format. But, instead of directly using the analogy rule to arrange the POS sequence, we permute the POS sequence to different combination to take care of all the possibilities. Using this method we might be able to derive more than 1 POS sequence from the different combination. If any of the combination which does not fulfill the analogy rules during analogy process, the combination is counted as invalid.

The combination which fulfills the analogy rules successfully derived the forth POS sequence which has clear relationship with the given the 3 POS sequence. One of the successful derived POS sequence using analogy is:

V AU_INF V : V N CC N :: PRON V AU_INF V : X

Where X = PRON V N CC N

For easier understanding, these derived POS sequence from SSTC is defined as analogy template. In order to ensure that the analogy template is a valid template which can be used for translation, a tree representation is constructed for the analogy template.