INDEXING OF BILINGUAL KNOWLEDGE BANK BASED ON THE SYNCHRONOUS SSTC STRUCTURE

by

YE HONG HOE

Thesis submitted in fulfilment of the requirements
for the degree of
Master of Science

July 2006

APPENDICES 69

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BKB	Bilingual Knowledge Bank
EBMT	Example-Based Machine Translation
ID	Identity
KIMD	Kamus Inggeris-Melayu Dewan
POS	Part of Speech
SSTC	Structured String-Tree Correspondence
S-SSTC	Synchronous Structured String-Tree Correspondence
inf	infinitive form
past	past tense
pl	plural
pre	present tense
sg	singular

MENGINDEKS BANK PENGETAHUAN DWIBAHASA BERDASARKAN STRUKTUR SSTC SEGERAK

ABSTRAK

Idea asas bagi mesin terjemahan berdasarkan contoh adalah untuk menterjemahkan ayat dengan menggunakan contoh-contoh terjemahan yang serupa. Bagi menambah ingatan mesin terjemahan ini, seseorang hanya perlu menambah contoh-contoh terjemahan yang baru ke dalam pangkalan data. Walau bagaimanapun, apabila pangkalan data contoh menjadi semakin besar, semakin sukar untuk mendapat kembali contoh-contoh yang sesuai sebagai rujukan bagi penterjemahan.

Dalam satu kerja sebelum ini, contoh terjemahan dianotasi dengan satu struktur fleksibel yang dipanggil 'Structured String-Tree Correspondence' (SSTC) segerak, dan disimpan dalam satu pangkalan data yang dipanggil Bank Pengetahuan Dwibahasa. Walau bagaimanapun, indeksnya yang berdasarkan perkataan bukan satu cara mendapat kembali contoh-contoh terjemahan yang efisien.

Melanjutkan kerja tersebut, kami mengeksploitasikan persamaan (yang termasuk pemetaan antara bahagian sumber dan sasaran daripada contoh-contoh terjemahan) dalam struktur SSTC segerak itu untuk memperbaiki operasi mendapat kembali contoh-contoh terjemahan. Tambahan pula, kami membuat generalisasi atas contoh-contoh terjemahan untuk meningkatkan liputan teks input tanpa pertambahan dalam pangkalan data contoh.

Berdasarkan persamaan dan generalisasi tersebut, dua kriteria, iaitu perkataan dan struktur, telah digunakan untuk mengindeks contoh-contoh terjemahan. Mengindeks dengan menggunakan perkataan memberikan kami liputan teks input yang baik manakala mengindeks dengan menggunakan struktur boleh digunakan untuk menghasilkan terjemahan dalam bentuk yang baik. Kami mengklasifikasikan

indeks struktur mengikut jenis dan struktur contoh yang berlainan. Indeks struktur termasuk indeks frasa dan indeks umum (yang termasuk pula indeks pencontoh dan indeks peraturan).

Selain mengindeks, kami menambah beberapa maklumat linguistik (iaitu leksikon dwibahasa dan pendasaran ('stemming')) ke dalam sistem terjemahan Inggeris-Melayu kami untuk meningkatkan lagi liputan teks input.

Dalam penterjemahan, diberi satu ayat input, kami pertama sekali menjalankan pemadanan leksikal dengan menggunakan indeks perkataan dan indeks frasa. Kemudian, kami melakukan pemadanan struktur dengan menggunakan indeks umum untuk mencari contoh-contoh terjemahan yang rapat dengan ayat input dari segi struktur.

Keberkesanan pendekatan kami telah dinilai berbanding dengan kerja sebelum ini. Sistem kami mengatasi kerja tersebut dari segi struktur tatabahasa dan ketepatan hasil penterjemahan.

INDEXING OF BILINGUAL KNOWLEDGE BANK BASED ON THE SYNCHRONOUS SSTC STRUCTURE

ABSTRACT

The basic idea of Example-Based Machine Translation (EBMT) is to translate a sentence by using similar translation examples. To increase memory of EBMT, one simply needs to add new translation examples into a database. However, when the example database becomes larger, it becomes more difficult to retrieve proper examples as references for a translation.

In a previous work, translation examples were annotated with a flexible structure called synchronous Structured String-Tree Correspondence (SSTC), and were stored in a database called Bilingual Knowledge Bank. However, its word-based indexing was not an efficient way of retrieving translation examples.

Extending on the previous work, we exploited correspondences (that include mappings between source and target parts of translation examples) in the synchronous SSTC structure to improve retrieval of translation examples. In addition, we generalized translation examples to increase coverage of input text without increasing the example database.

Based on the correspondences and generalization, two criteria, viz. word and structure, were used to index the translation examples. Indexing using words gave us a good coverage of input text while indexing using structures may be used to produce well-formed translations. We classified structural indexes according to different types and structures of examples. Structural indexes include a phrasal index and generalized indexes (which in turn include template indexes and a rule index).

Besides indexing, we added some linguistic information (i.e. bilingual lexicon and root forms) into our English-Malay EBMT system in order to further increase coverage of input text.

In our translation process, given an input sentence, we first carried out lexical matching using a word index and a phrasal index. Then, we performed structural matching using generalized indexes to find translation examples that are structurally close to the input sentence.

The effectiveness of our approach was evaluated against the previous work. Our system outperforms the previous work in terms of well-formedness and accuracy of translation outputs.

CHAPTER 1
INTRODUCTION

Indexing is used to facilitate retrieval of records or information. We will look into how indexing can be applied in Example-Based Machine Translation (EBMT)[1]. As an introduction, we will see what EBMT is, why indexing is needed in EBMT, our problem statement, and outline of the following chapters.

1.1 Example-Based Machine Translation (EBMT)

The basic idea of Example-Based Machine Translation (EBMT) is to translate a sentence by using similar translation examples.

EBMT is analogous to human translation behaviour. According to Nagao (1984), man does not translate a sentence by applying deep linguistic analysis, but rather, by:

(i) decomposing a sentence into phrases

(ii) translating the phrases into target language phrases using similar examples as references, and

(iii) combining the target phrases to give a translated sentence

Translation examples can be collected from a parallel corpus[2] and then stored in a database. But before a parallel corpus can be useful for EBMT, the corpus must be aligned[3]. The corpus can be aligned at sentence or subsentence (e.g. phrase and word) level.

[1] EBMT was first proposed by Makoto Nagao in 1981. However, the paper presented by Nagao was not published until 3 years later (Somers, 1999:116).

[2] "[A] text together with its translation" is called parallel corpus (Somers, 1999:150).

[3] A parallel corpus is aligned when "the two texts have been analysed into corresponding segments" (Somers, 1999:150).

The EBMT approach can perform well-structured translations as long as there are similar examples. For example, given the English sentence (1), if there is a similar example, this approach will produce the well-structured Malay sentence (2) which has different structure from the structure of the English sentence.

(1) he speaks English

(2) dia bercakap dalam bahasa Inggeris
'he' 'speak' 'in' 'English'

In addition, the EBMT approach is suitable for non-literal (e.g. idiomatic expression) translation. For example, given English idiom “take it easy”, if there is a similar example, this approach will not translate the idiom literally into the Malay phrase “mengambil ia mudah”, but will produce the correct translation “bersenang-senang”.

1.2 Motivation

To increase memory of EBMT, one simply needs to add more translation examples into a database. Gradually, the example database becomes huge. Then, it becomes more difficult to retrieve proper examples as references for a translation. So, we need indexing to facilitate retrieval of suitable examples.

How should translation examples be indexed? If the examples are indexed at word level, then this word index will give us a good coverage of text to be translated. For example, word index that contains the words “take”, “it”, and “easy” can cover the English idiom “take it easy”. However, EBMT using word index generally cannot handle idiomatic expressions, and may not produce well-structured translations. For example, the English idiom may be translated word by word as “mengambil ia mudah” which is neither correct nor well-formed.

If the examples are indexed at the phrasal (or structural) level, then EBMT using a phrasal index can handle some idiomatic expressions, and may produce well-structured translations. For example, a phrasal index that contains the phrase "take it easy" can cover the English idiom "take it easy" and may be used to produce the correct Malay translation "bersenang-senang". However, a phrasal index can only be used for phrasal exact match and this exact match limits its usage in translation. For example, a phrasal index that contains the sentence "he speaks English" cannot be used to translate the English sentence "he speaks French".

If the examples can be indexed using some generalized text segments, then this generalized index can increase coverage of input text without increasing the example database. For example, the English sentence "he speaks English" can be generalized as "he speaks <language>". Then, a generalized index that contains the generalized text can be used to match the English sentence "he speaks French", and can be used to produce the well-formed Malay translation "dia bercakap dalam bahasa Perancis".

Hence, a combination of word index, phrasal index and generalized index can meet translation needs of coverage and well-formedness.

1.3 Problem Statement

Our aim is to build an English-Malay EBMT system that can solve the following problems:

i. As a large parallel corpus is hard to come by, how can we increase the coverage of input text using the same parallel corpus?

ii. Given an input sentence, how can we find a set of similar translation examples? The examples should have some words contained in the input sentence or have a structure similar to that of the input sentence.

In this thesis, we propose to extend a previous work (i.e. Al-Adhaileh's (2002) work) in our school to solve the above problems:

i. We use generalized indexes to increase coverage of input text without increasing the size of translation examples.

ii. We use word and phrasal indexes to find translation examples that contain some words in an input sentence. Then, we use structural indexes to find translation examples that are structurally close to the input sentence.

1.4 Outline of Thesis

This thesis is organized into five chapters. This present chapter provides an overview of EBMT, brings out our motivation, and states our problems and solutions. In chapter two, we discuss some underlying concepts before discussing some techniques in indexing. In chapter three, we describe how we construct a word index and structural indexes (which include a phrasal index and generalized indexes). We also describe how we use our different indexes to match similar examples. In chapter four, we describe our experiments and report the results obtained using our methods in an English-Malay EBMT system. In chapter five, we conclude with a discussion of our approach and some suggestions for future work.

CHAPTER 2
BACKGROUND

Our study is an extension on Al-Adhaileh's (2002) work. In Section 2.1, we describe some underlying concepts in his work which include Bilingual Knowledge Bank and synchronous SSTC.

Although Al-Adhaileh (*ibid.*) did propose a flexible annotation schema, there exist some weaknesses in his word-based indexing. So, we have carried out a literature survey on indexing methods used in existing Example-Based Machine Translation (EBMT) approaches, and the methods will be presented in Section 2.2.

2.1 Some Underlying Concepts

Before going into details of Structured String-Tree Correspondence (the basic of synchronous SSTC), we describe what *Bilingual Knowledge Bank (BKB)* is.

Consider BKB as a database (or collection) of translation examples where examples are normally annotated with syntactic tree structures and translation units have been established between source and target parts of the examples (Sadler and Vendelmans, 1990; Al-Adhaileh and Tang, 2001).

2.1.1 Structured String-Tree Correspondence (SSTC)

Correspondences between a language string and its representation tree are not always straightforward (or projective) (see Figure 2.1). For this reason, Boitet and Zaharin (1988) argued for the need to separate the language string from its representation tree, and thus proposed *Structured String-Tree Correspondence (SSTC)*.

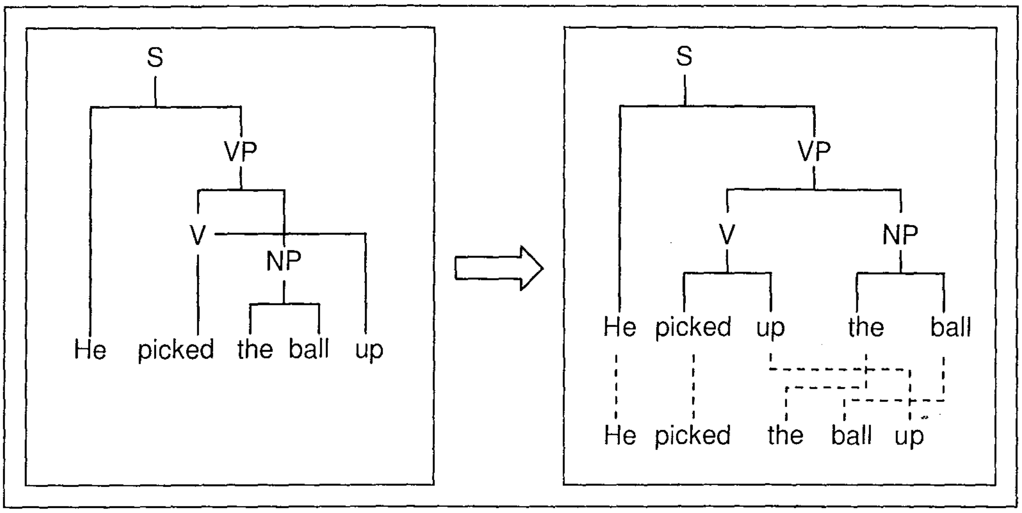


Figure 2.1: Separation of the string "**He picked the ball up**" from its non-contiguous phrase structure tree (adapted from Boitet & Zaharin, 1988).

A SSTC contains two sets of interrelated correspondences. One set of correspondences is between nodes and (possibly non-contiguous) substrings. The other set of correspondences is between (possibly incomplete) subtrees and (possibly non-contiguous) substrings. These two sets of correspondences can be encoded on a representation tree by attaching to each node *N* two sequences of intervals, called *SNODE(N)* and *STREE(N)* respectively (cf. Al-Adhaileh and Tang, 2001; Al-Adhaileh et al., 2002). Figure 2.2 shows an SSTC for the sentence "He picked the ball up". Each word in the sentence is attached with an interval: He(0_1), picked(1_2), the(2_3), ball(3_4) and up(4_5). Each node in the tree is attached with SNODE and STREE intervals. For examples, SNODE for node "picked" is (1_2), STREE for node "picked" is also (1_2), SNODE for node "V" is ∅ (empty), and STREE for node "V" is (1_2+4_5) which corresponds to the words "picked" and "up".

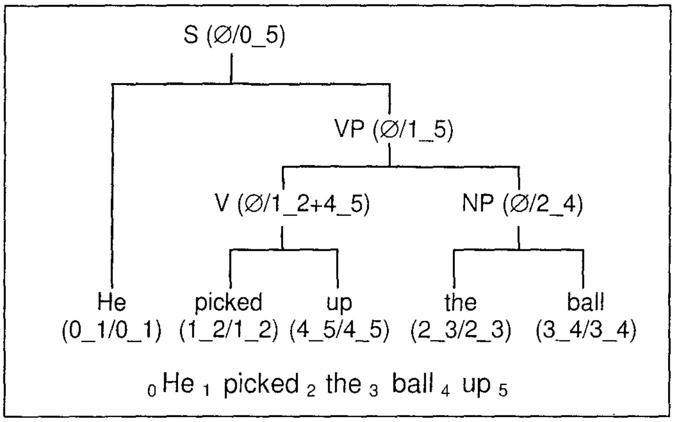


Figure 2.2: SSTC for the sentence "**He picked the ball up**".

2.1.2 Synchronous SSTC (S-SSTC) and Related Work

Before going into details of *Synchronous SSTC (S-SSTC)*, we will first describe the idea of synchronization. Synchronization has been introduced on grammar formalisms for describing structural correspondences between two languages that are closely related but have different structures. For example, the formalism of synchronous Tree Adjoining Grammar (S-TAG) was used to associate the syntactic structure of a natural language with its semantic representation (Shieber & Schabes, 1990) or its translation in another language (Abeillé et al., 1990; Egedi et al., 1994).

Instead of investigating synchronous grammars, Al-Adhaileh et al. (2002) proposed a flexible annotation schema (i.e. S-SSTC) that makes use of synchronous property and flexibility of SSTC to describe translation examples. Moreover, this schema can handle some non-standard correspondence cases in translation of natural languages, such as many-to-one mapping, elimination of dominance and inversion of dominance (illustrated in Figure 2.3).

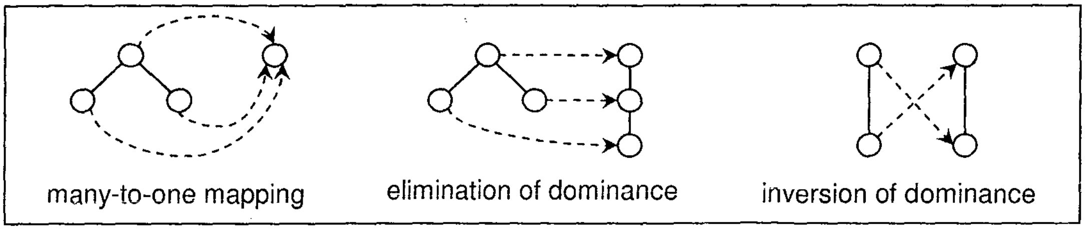


Figure 2.3: Examples of non-standard correspondence cases

An S-SSTC consists of a SSTC of one language, a SSTC of another language, and together with correspondences between the two SSTCs. Figure 2.4 shows an S-SSTC for the English sentence “He picked the ball up” and its Malay translation “Dia mengutip bola itu”. In the figure, a solid arrow indicates a correspondence between a string and its representation tree, whereas a dotted arrow indicates a correspondence between source and target SSTCs. The correspondences are categorized into two: lexical correspondences (ℓ_{sn}) and subtree correspondences (ℓ_{st}). A lexical correspondence is denoted by an SNODE pair. For example, the correspondence between “picked up” and “mengutip” is indicated by the SNODE pair (1_2+4_5, 1_2). On the other hand, a subtree correspondence is denoted by an STREE pair. For example, the correspondence between “the ball” and “bola itu” is indicated by the STREE pair (2_4, 2_4).

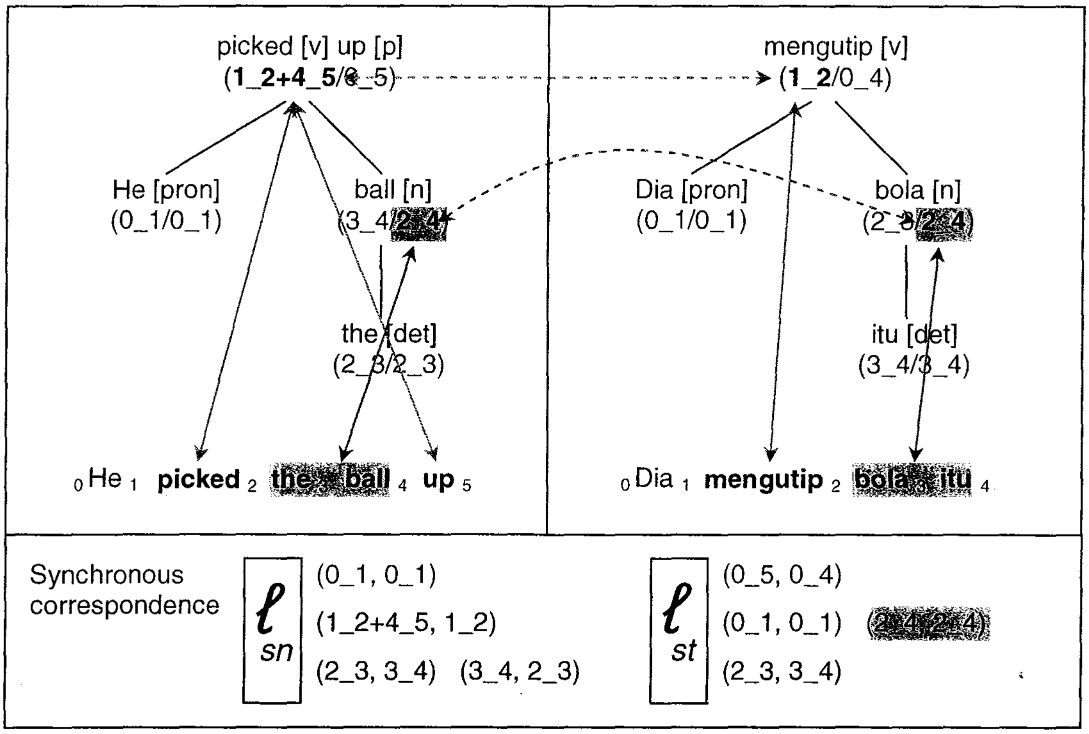


Figure 2.4: An S-SSTC for the English sentence “**He picked the ball up**” and its Malay translation “**Dia mengutip bola itu**” (adapted from Al-Adhaileh et al., 2002).

2.2 Indexes in EBMT

In EBMT, an index can be considered as a list of source language text segments (e.g. words, phrases, etc), and the list facilitates retrieval of target language equivalents. Furthermore, the list may be ordered and each item of the list may contain reference(s) to the original translation example(s).

We have classified indexes into word index, structural index, and generalized index. Each of them will be discussed in the following sections.

2.2.1 Word Indexes

A word index contains a list of source language words, and each word may have reference(s) to the original translation example(s). We will discuss the use of word indexes in some EBMT systems.

The *Pangloss Example-Based Machine Translation* (PanEBMT) (Brown, 1996) contains a bilingual corpus aligned at sentence level. The corpus is indexed at word level. Based on the word index, PanEBMT tries to produce translations of consecutive words in the input text. However, the system does not perform well if the correspondence between source word(s) and target word(s) is not one-to-one. Furthermore, the system is lacking of complete input coverage. The system cannot produce a translation for a word unless the word co-occurs with another word (in both input and corpus) and is properly aligned.

Brown (2004) has adapted the *Burrows-Wheeler Transform* (BWT) which was originally created for data compression to index words in a bilingual corpus. The modified BWT decreases the size of corpus that needs to be read into memory, and ultimately speeds up the process of matching input text with translation examples.

Furthermore, frequent phrasal translations may be precompiled to speed up the translation process. Brown (*ibid.*) uses n-gram (where n > 1) matching which is not suitable for translations involving many-to-one and one-to-many correspondences.

Al-Adhaileh (2002) (cf. Al-Adhaileh & Tang, 1999) proposed an English→Malay EBMT system based on a database (BKB) of synchronous SSTCs (see Section 2.1.2). The synchronous SSTC structure is flexible and can handle many-to-one mappings, e.g. "pick up" → "mengutip". The BKB (see Section 2.1) is indexed at word level. The system does not use n-gram matching but selects example(s) with most words in the source sentence. However, the system may fail to select words from structurally similar example(s). For example, given the input sentence (3), the system may produce the erroneous output sentence (5) instead of the correct output (6) from a set of examples as in (4). Note that words selected for translation are underlined.

(3) he knelt on the floor

(4) *he cut his name on the rock → dia menggoreskan namanya pada batu itu*
'he' 'cut' 'his name' 'on' 'rock' 'the'

she mops the floor → dia mengelap lantai
'she' 'mop' 'floor'

she knelt on the cushion → dia berlutut di atas kusyen
'she' 'kneel' 'on' 'cushion'

(5) *dia berlutut pada lantai itu

(6) dia berlutut di atas lantai itu

Al-Adhaileh (2002) does not consider whether selected source word(s) can be aligned to target word(s). For example, if the source word "name" is selected from the first example in (4), the word may not be translated because 'his name' is aligned to 'namanya' and hence the word 'name' alone is not aligned to anything.

2.2.2 Structural Indexes

A structural index contains a list of source language tree structures or phrase structures. The tree structures may be representations of phrases. We will discuss the use of structural indexes in some EBMT systems.

In Sato's (1995) system, each translation example is represented as a pair of dependency trees with explicit links between subtrees. Suitable examples are retrieved based on a list of *translatable*[4] source trees. Translatable subtrees from different examples are combined to match against the dependency tree of an input sentence. Hence, multiple translation candidates may be generated. The candidates are ranked based on the size of translatable subtrees and similarity between two contexts of the subtrees: in source tree and in translation examples.

In Watanabe's (1995) system, each translation rule (or example) has three components: a matching graph (i.e. source graph), a construction graph (i.e. target graph), and a set of mappings between them. A source node may be connected to two target nodes: one governs the source node, and the other is governed by the source node. These mappings are called *upward mapping* and *downward mapping.* Suitable rules are retrieved based on a list of matching graphs. Given an input graph, a set of rules is selected such that their matching graphs cover the input graph. Apart from graph distance, similarity between graphs may be measured based on syntactic and semantic feature distances.

To store translation examples in an organized way, and to identify (possibly non-contiguous) "multi-word verbs" (MWV) (e.g. *pick up*, *ran into*), Kee (2004) proposed a variant of Al-Adhaileh and Tang's (1999) approach. Translation examples

[4] A translatable source tree has a correspondence link to target tree.

are decomposed into smaller segments which are stored in different databases according to syntactic categories or parts of speech (POSs) of the segments. Bilingual segments are then indexed based on syntactic category or POS, identity of original example, source segment POS pattern, and source segment size. Furthermore, Kee (2004) collected a set of MWVs and uses them to identify MWV in input sentence.

2.2.3 Generalized Indexes

A generalized index contains a list of source language generalized text fragments or tree structures. A generalized index can also be called a template index. A template is obtained by replacing some words (in a text fragment) or some nodes (in a tree structure) with variables.

We have classified generalized indexes into string and structural generalized indexes. We will present these indexes in the following subsections.

2.2.3(a) String Indexes

A string generalized index contains a list of arbitrary generalized text fragments of the source language. We will discuss the use of this kind of index in some EBMT systems.

Researchers from the Department of Computer Engineering and Information Sciences at Bilkent University, Ankara, Turkey, have published their generalization techniques extensively (Cicekli & Güvenir, 1996, 2001; Güvenir & Cicekli, 1998; Cicekli, 2000). The authors proposed to learn translation templates by observing similarities and differences between two translation examples. Similar parts in source sentences should correspond to similar parts in target sentences. Likewise, different parts in source sentences should correspond to different parts in target sentences. A string template is generated by replacing the differences with variables. For example, translation examples that are represented in lexical form (i.e. stems and morphemes) in (7a) and (7b) can be generalized as template (8). Then, source parts of translation templates are used as an index to find the most specific template for a given input sentence. A similar idea can be found in Echizen-ya et al. (1996, 2000).

(7) a. I give+p the ticket to Mary ↔ Mary+'e billet+yH ver+DH+m
b. I give+p the pen to Mary ↔ Mary+'e kalem+yH ver+DH+m
(8) [I give+p the X_s to Mary] ↔ [Mary+'e X_t+yH ver+DH+m] if [X_s ↔ X_t]

McTait and Trujillo (1999) proposed to extract translation patterns based on the co-occurrence of possibly non-contiguous strings in two or more translation examples. For example, the sentence pairs in (9a) and (9b) can be generalized as template (10). Then, the source parts of translation patterns are used as an index to find the most specific pattern that covers the input sentence to the largest extent. McTait (2001a, b) extends this approach by using external linguistic knowledge sources (e.g. morphological analysis and POS tagging) to solve some ambiguities.

(9) a. The Commission gave the plan up ↔ La Comisión abandonó el plan
b. Our Government gave all laws up ↔ Nuestro Govierno abandonó todas las leyes
(10) X_s gave Y_s up ↔ X_t abandonó Y_t

Brown (1999) proposed a generalization technique that uses *equivalence class*. An equivalence class contains words that are interchangeable, such as numbers, weekdays, country names, etc. Translation examples are generalized by replacing some of their words (or phrases) with respective class names. Generalized source parts of translation examples (or source templates) are used as an index to find partial exact matches for a given input. For example, sentence (11) can be generalized as template (12) which can be used to match sentence (13). Brown (2000) proposed to extract the classes automatically by using clustering techniques. Members of a class may differ syntactically and semantically but follow the same sentence pattern. Furthermore, Brown (2001) combines the transfer rule induction technique from Cicekli and Güvenir (2001) and the word clustering technique from Brown (2000), and reported a reduction in needed training text by a factor of twelve.

(11) John Miller flew to Frankfurt on December 3rd.

(12) <person-m> flew to <city> on <date>.

(13) Dr. Howard Johnson flew to Ithaca on 7 April 1997.

Veale and Way (1997) proposed a template-based EBMT system called *Gaijin*. In the Gaijin system, a template is generated by segmenting a translation example into phrases using "marker sets" (i.e. closed word classes, e.g. preposition, determiner and quantifier). Each phrase or segment may begin with a marker word. Each template contains a sequence of segments which in turn contain marker types and also references to original texts. The Example database is indexed by sequences of markers. This structural index is used to find the most specific template for a given input. Then, segments in the selected template may be adapted to fit the input sentence.

2.2.3(b) Structural Indexes

A structural generalized index contains a list of generalized tree structures of the source language. We will discuss the use of this index in some EBMT systems.

Matsumoto and Kitamura (1995) proposed to extract templates from translation examples by using semantic classes. First, source and target sentences are parsed into dependency structures. Then, structural matching between source and target dependency structures produces matched (sub)graphs. A particular source word is identified for generalization. All matching graphs that contain the word as head word are generalized based on semantic classes in a thesaurus. The example database is then indexed based on head words, followed by source structures with semantic conditions. The authors stated that translation quality depends on the thesaurus, and that their method cannot deal with idiomatic expression and complex sentences.

Menezes and Richardson (2001) proposed a method which automatically acquires translation patterns or "transfer mappings" from sentence-aligned corpora. A transfer mapping contains a pair of aligned trees in "logical form" (see Figure 2.5). Moreover, transfer mappings are extended to include context. Proper context can be used to resolve conflict in translation. The transfer mappings are stored in a repository called "MindNet" (Richardson et al., 1998). Source parts of the mappings (treelets) can be considered as the index of the MindNet. Richardson et al. (2001a, b) proposed to use the MindNet in a hybrid MT system to achieve "commercial-quality" translation.

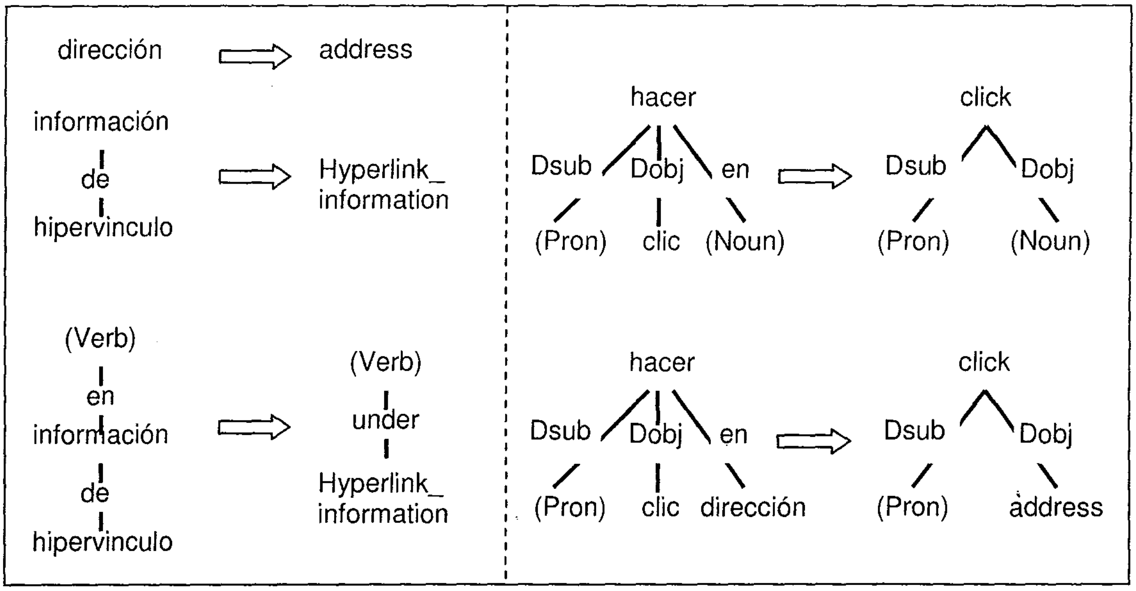


Figure 2.5: Examples of Spanish-English transfer mappings (Menezes & Richardson, 2001).

Collins and Cunningham (1996) proposed to retrieve examples based on case-based reasoning (cf. Somers, 2001). They argued that not only similarity between input sentence and example source sentence is important but also *adaptability* from example source sentence to example target sentence is important. Examples which have high adaptability can be used to produce grammatically correct translations. Retrieval has two phases: string retrieval and syntactic retrieval. String retrieval involves exact word matching whereas syntactic retrieval involves syntax-based matching with different levels of generalization. For syntactic retrieval, generalized source syntactic structures are used as an index.

2.3 Summary

In this chapter, we have just looked at some underlying concepts of our work which include BKB and synchronous SSTC. We have classified indexes in EBMT into word index, structural index, and generalized index. We further divided generalized index into two types, namely string and structural generalized indexes.

Indexing at word level alone is insufficient (cf. Al-Adhaileh, 2002). In the next chapter, we will see how we can improve indexing of BKB, and hence retrieval of examples by integrating a structural index and even a structural generalized index.

CHAPTER 3
METHODOLOGY

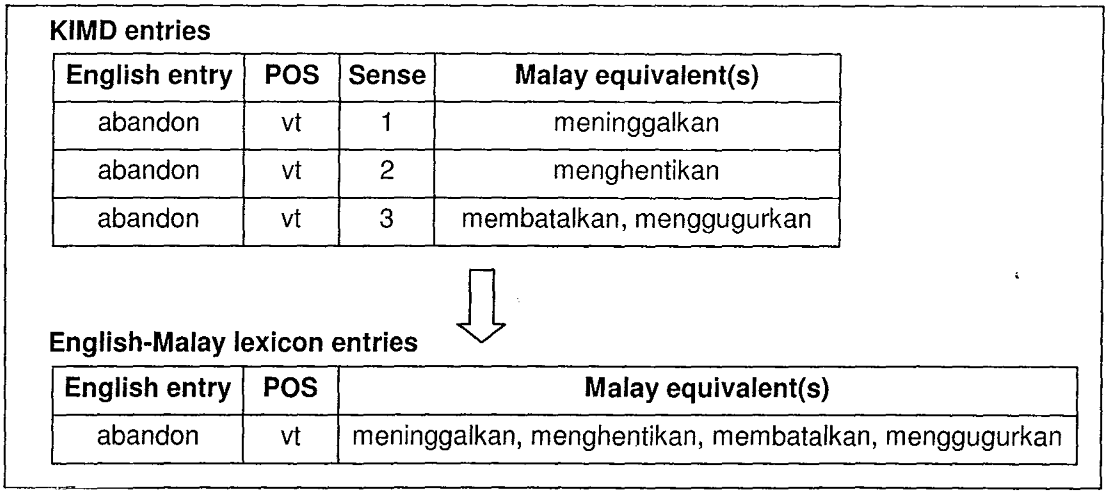
In this chapter, we will describe our methodology in indexing translation examples that reside in our English-Malay EBMT system.

In Section 3.1, we describe some preparation which needs to be done prior to indexing. In Section 3.2, we present in detail our indexing techniques before going to describe how we use the indexes for translation in Section 3.3. In Section 3.4, we present the overall translation process.

3.1 Prior to Indexing

3.1.1 Construction of a Bilingual Lexicon

To improve translation coverage in our system, we construct an English-Malay bilingual lexicon automatically by extracting entries from prolog files (prepared by a previous project) of an English-Malay dictionary, viz. *Kamus Inggeris-Melayu Dewan* (KIMD) (Dewan Bahasa dan Pustaka, 1991). From each KIMD entry, we extract the English entry, the part of speech (POS), and the Malay equivalent(s). Then, we construct the lexicon by grouping English entries by POS. We extracted a total of 44,265 lexicon entries (see Figure 3.1).



KIMD entries

English entry	POS	Sense	Malay equivalent(s)
abandon	vt	1	meninggalkan
abandon	vt	2	menghentikan
abandon	vt	3	membatalkan, menggugurkan

English-Malay lexicon entries

English entry	POS	Malay equivalent(s)
abandon	vt	meninggalkan, menghentikan, membatalkan, menggugurkan

Figure 3.1: Extraction of an English-Malay bilingual lexicon from the KIMD

3.1.2 Lemmatization

In Al-Adhaileh's (2002) work, indexing was based on word surface forms. This manner of indexing restricts his system to exact word match. To increase coverage of input text, we added information on lemma for noun, verb, and also auxiliary verb[5]. In addition, tense (e.g. present (pre), past) and number (singular (sg) or plural (pl)) are added for illustration purpose. Figure 3.2 shows an example of our enriched English SSTC structure.

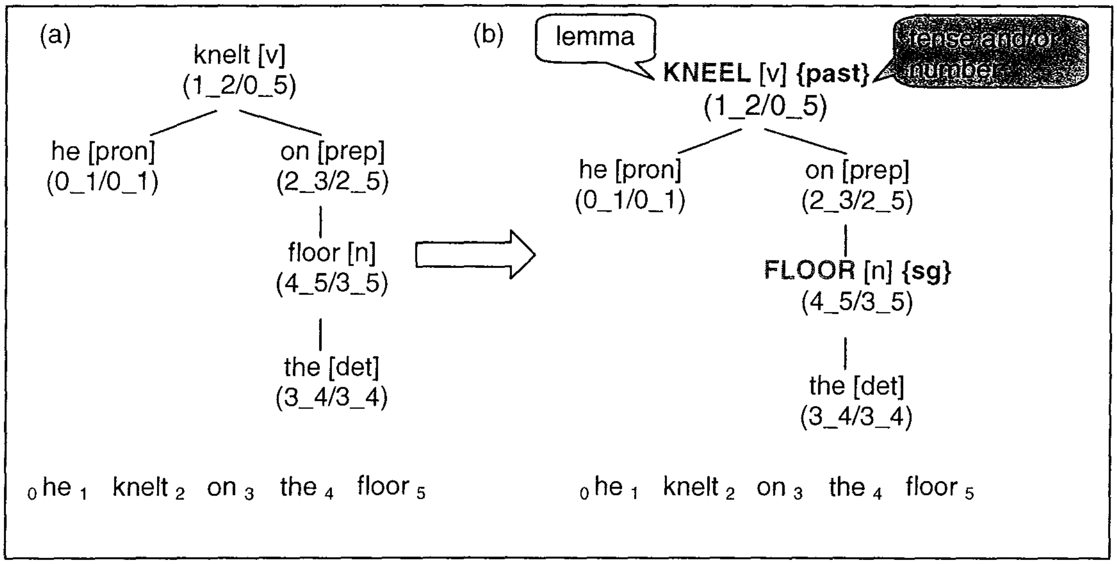


Figure 3.2: SSTC from Al-Adhaileh's (*ibid.*) work (see (a)) and our enriched SSTC (see (b)) for the English sentence "**he knelt on the floor**"

3.1.3 Subcategorization of Determiner

In English, a determiner normally precedes a noun. However, in Malay, a determiner may precede or follow a noun. Hence, to tune our system, we can subcategorize English determiners in relation to the position of their Malay equivalents in Malay noun phrases. Based on this criterion, we can categorize English determiners into two types[6]:

[5] see Appendix A for the parts of speech used in our system
[6] see Appendix B for the English determiners collected from our BKB

a) type 1 (det1), where Malay determiner precedes the noun, e.g. 'every' in

(14) every car → setiap kereta
'every' 'car'

b) type 2 (det2), where Malay determiner follows the noun, e.g. 'the' in

(15) the car → kereta itu
'car' 'the'

3.2 Indexing of the Bilingual Knowledge Bank

As mentioned in Section 2.2.3(b), not only similarity between input sentence and example source sentence is important but also adaptability from example source sentence to example target sentence is important. Hence, we index examples based on source language text segments that have correspondence links. The examples in our Bilingual Knowledge Bank (BKB) are indexed based on two criteria: word and structure. The indexing pseudocode is attached in Appendix C.

3.2.1 Word Index

In our system, the word index is built from lexical correspondences recorded in S-SSTCs. The lexical correspondences (ℓ_{sn}) are denoted by SNODE pairs (see Section 2.1.2). For example, an S-SSTC is shown in Figure 3.3.

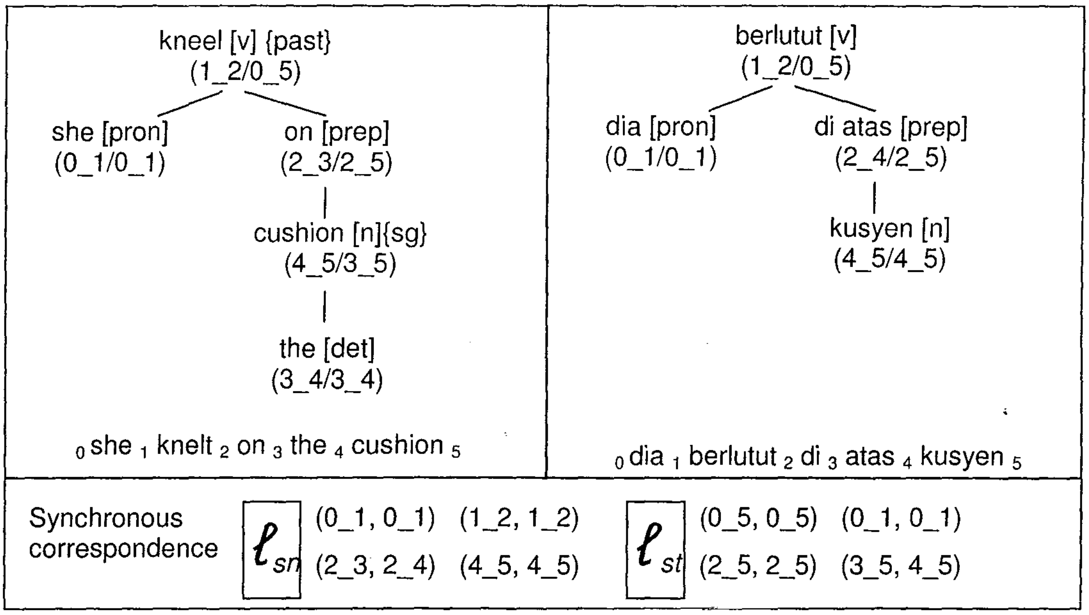


Figure 3.3: An S-SSTC for the English sentence "**she knelt on the cushion**" and its Malay translation "**dia berlutut di atas kusyen**"

Based on SNODE pairs in the S-SSTC, lexical correspondences are extracted (as shown in Figure 3.4).

• she [pron]	• dia [pron]	• kneel [v] {past}	• berlutut [v]
• on [prep]	• di atas [prep]	• cushion [n]{sg}	• kusyen [n]

Figure 3.4: Lexical correspondences extracted from the S-SSTC shown in Figure 3.3

However, if a lexical correspondence spans more than one node on one of the two trees, the correspondence will not be considered for word indexing. For example, the English phrase “in terms of” which corresponds to the Malay phrase “dari segi” will not be considered for word indexing because the phrase “in terms of” spans three nodes in its tree representation.

Table 3.1 shows a word index built from the lexical correspondences. Our word index contains six fields. Among them, the field “source lemma” can be empty, the field “S-SSTC IDs” contains identities (IDs) of original S-SSTC, and the field “target string” is useful for breaking down the field “source string” according to different translation equivalents. In general, a word index should contain at least the source word and the reference IDs. Other information will help us in choosing the correct target word.

Table 3.1: Word index built from the lexical correspondences shown in Figure 3.4

Source String	Source Lemma	Source POS	Target String	S-SSTC IDs	Frequency
she	-	pron	dia	2356	1
knelt	kneel	v	berlutut	2356	1
on	-	prep	di atas	2356	1
cushion	cushion	n	kusyen	2356	1

Apart from lexical correspondences, we have another type of correspondences: subtree correspondences. In the next section, we describe our structural index which is based on the subtree correspondences.

3.2.2 Structural Indexes

We classify structural indexes according to different examples and structures. Examples are categorized according to different levels of generalization: (1) fully lexicalized (i.e. no generalization), (2) partially generalized, and (3) fully generalized. Figure 3.5 shows an example of generalization for an English sentence's representation tree. Note that tree nodes are generalized by POS. Furthermore, we have 3 types of partially generalized examples. We will give more details in the following subsections.

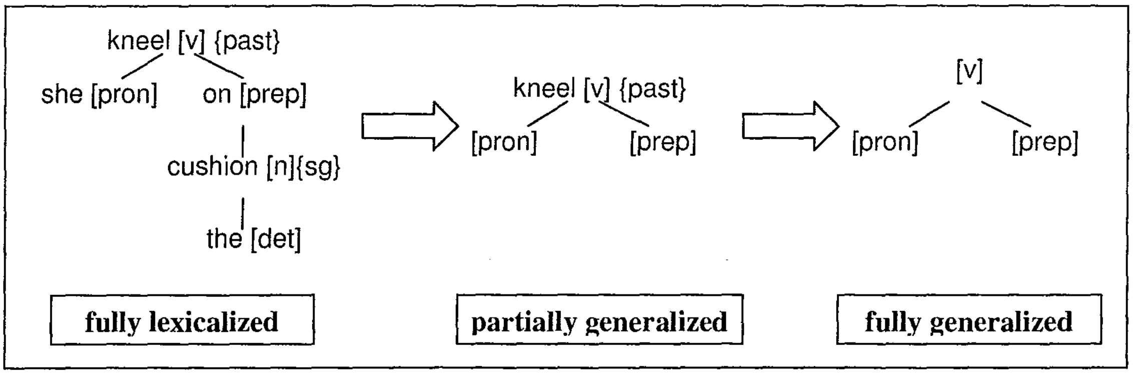


Figure 3.5: An example of different levels of generalization for the representation tree of the English sentence "**she knelt on the cushion**"

On the other hand, structures are categorized according to position of a structure in a representation tree. We have categorized structures into 3 types: (1) root, (2) intermediate, and (3) terminal. Figure 3.6 shows an example of different structures for an English sentence's representation tree.

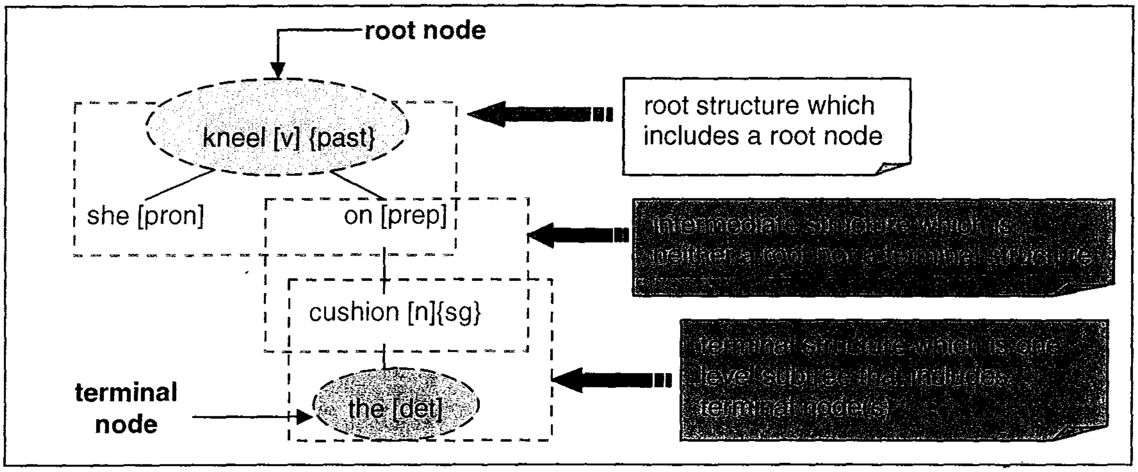

root node
kneel [v] {past}
she [pron]
on [prep]
cushion [n]{sg}
terminal node
the [det]
root structure which includes a root node
intermediate structure which is neither a root nor a terminal structure
terminal structure which is one level subtree that includes terminal node(s)

Figure 3.6: An example of different types of structures for the representation tree of the English sentence "**she knelt on the cushion**"

Classification of structural indexing can be summarized into Table 3.2. In this table, an empty cell (□) means that structural indexing applies for a particular combination of example and structure. However, as fully lexicalized examples contain no generalization and can stand by themselves (i.e. no need replacement within the examples), we do not associate these examples with different structures. There are different types of partially generalized examples, viz. type I, type II, and type III. As type II and III examples involve more than one-level subtree, terminal structure for the examples is not applicable (see Section 3.2.2(b)).

Table 3.2: Classification of structural indexing

Examples / Structures	Fully Lexicalized	Partially Generalized			Fully Generalized
		Type I	Type II	Type III	
Root					
Intermediate					
Terminal			☒	☒	

3.2.2(a) Fully Lexicalized Examples

Fully lexicalized examples consist of pairs of source and target phrases / sentences. Hence, structural indexes for this kind of examples can be considered as phrasal and sentential indexes. For easier discussion, we will call these indexes as phrasal index from this point onward. The phrasal index can be built from subtree correspondences recorded in S-SSTCs. The subtree correspondences (ℓ_{st}) are denoted by STREE pairs (see Section 2.1.2). For example, based on STREE pairs in the S-SSTC shown in Figure 3.3, the following subtree correspondences are extracted (as shown in Figure 3.7).

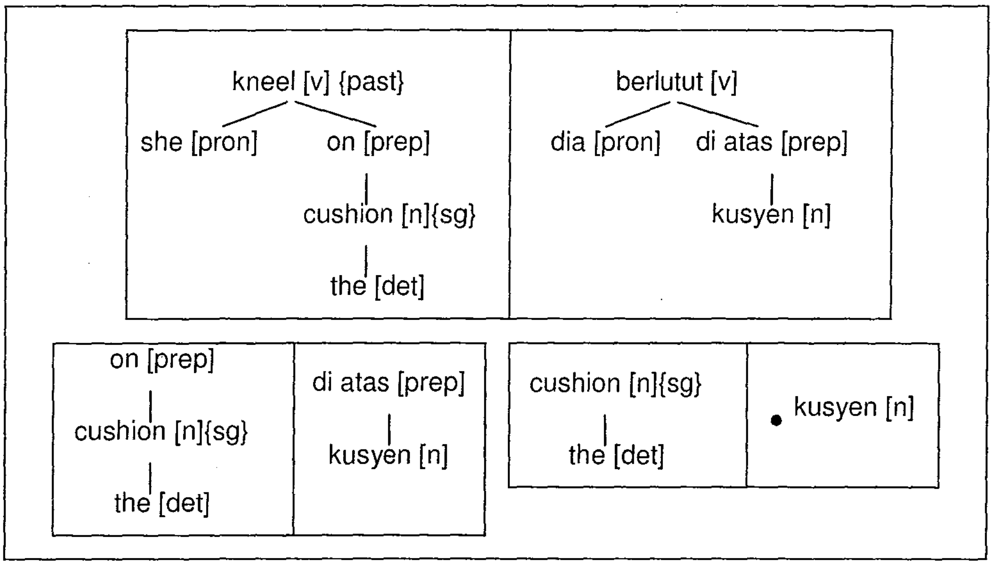


Figure 3.7: Subtree correspondences extracted from the S-SSTC shown in Figure 3.3

However, if a subtree correspondence consists of only one node in a source tree, the correspondence will not be considered for phrasal indexing. For example, the node "she [pron]" which corresponds to the node "dia [pron]" should be considered only in the word index.

A phrasal index built from the subtree correspondences is shown in Table 3.3. Note that the field "source POS" (which exists in our word index) is excluded from this phrasal index because the field is of no use for phrasal matching. In general, a phrasal index should contain at least the source phrase and the reference IDs. Other information will help us in choosing the correct target phrase.

Table 3.3: Phrasal index built from the subtree correspondences shown in Figure 3.7

Source String	Source Lemma	Target String	S-SSTC IDs	Frequency
she knelt on the cushion	she kneel on the cushion	dia berlutut di atas kusyen	2356	1
on the cushion	on the cushion	di atas kusyen	2356	1
the cushion	the cushion	kusyen	2356	1

3.2.2(b) Partially Generalized Examples

Partially generalized examples are actually templates. So, a structural index for this kind of examples can be considered as a template index. The format of a template index is shown in Table 3.4. Compared to a phrasal index, a template index contains 5 extra fields. Fields “type” and “structure” are needed because we have classified templates into different types of templates and structures. Fields “root word”, “root lemma”, and “root POS” are needed for structural matching. On the other hand, the field “target string” is excluded from the template index in the current implementation due to complexity. In general, a template index should contain at least the source template. Other information will help us in choosing the best source and target templates.

Table 3.4: Format of a template index

Root Word	Root Lemma	Root POS	Source String	Source Lemma	Type	Structure	S-SSTC IDs	Frequency

In the following subsections, we will describe different types of templates. For each type of template, we will also describe templates with different structures.