USING CONCEPTUAL VECTORS TO IMPROVE TRANSLATION SELECTION

LIM LIAN TZE

UNIVERSITI SAINS MALAYSIA

2006

USING CONCEPTUAL VECTORS TO IMPROVE TRANSLATION SELECTION

by

LIM LIAN TZE

Thesis submitted in fulfilment of the requirements

for the degree of

Master of Science

June 2006

ACKNOWLEDGEMENTS

It has been quite a journey undertaking my M.Sc. research. Now that that stage is over, I would like to express my heartfelt gratitude to *my loving parents, Lim Yoo Kuang and Gan Choon*, and *Wong Mun Choong*, my husband, for their never-wavering love, confidence, faith, encouragement and support. Thank you for reassuring me, time and again, that it is perfectly all right to have academic and "geeky" tendencies.

My supervisor, *Dr Tang Enya Kong*, showed me the ropes of academic research, and introduced me to the field of Natural Language Processing. This thesis would never have come into existence but for his knowledgeable guidance and immense patience. He, together with other lecturers at UTMK, including *Dr Chuah Choy Kim* and *Dr Bali Ranaivo-Malançon*, also showed me that contrary to popular belief, researchers *are* entitled to a "life" too, complete with hobbies and pastimes outside the research lab.

I have had the good luck to meet many intelligent and generous people, whose help was indispensable for my research. *Mosleh Al-Adhailah* and *Ye Hong Hoe* first showed me the innards of the Machine Translation system developed at UTMK, while *Lim Beng Tat* explained his work on sense disambiguation to me. I thank *Dr Francis Bond*, of *NTT Communications Science Laboratories*, for granting me permission to use the *GoiTaikei* concept hierarchy, and *Dr Mathieu Lafourcade* for sharing his work and insights on conceptual vectors. *Liu Muk Moi, Kho Chow Thee* and *Anna Teng Howe Ying* deserve special mention for painstakingly performing the initial Japanese-to-English translation of *GoiTaikei* concept labels, and recreating the concept hierarchy as a *Protégé* ontology. In addition, *Dr Chuah*, *Dr Ranaivo-Malançon*, and *Prof. Christian Boitet* have all given their feedback on my presentations and writings on different occasions. I must thank *Mr Tan Ewe Hoe* for helping me with the Malay abstract, while *Nur Hussein* inducted me

into the ways of *GNU/Linux*, all the while trying to convince me that *Star Trek* was really where this whole universal translator affair started, and that we should add support for Klingon as soon as possible.

Doing research in a non-"researcher-friendly" environment can be gruelling. Fortunately, this was not something that I have to endure, thanks to the emphasis on research culture and fine facilities provided by the *School of Computer Sciences* in general, and *all the staff at UTMK* in particular, especially *Mdm Rohana Omar* and *Mr Tan*. I also have the whole bunch of research students on Level 2 to thank, for keeping me sane by helping me go insane every now and then. *Hong Hoe, Adib, Nur Hana, Sara, Gan, Hussein, Siew Kin, Sze Ling,* you guys rock.

If there is anyone that I forgot to thank, my apologies and gratitude to you, and please be assured that it is unintentional. Thank you all, however brief our meeting might have been, for I am sure every little encounter helped me get to where I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BKB bilingual knowledge bank

CV conceptual vector

EBMT example-based machine translation

GT *GoiTaikei*

MT machine translation

NLP natural language processing

POS part-of-speech

SL source language

SSTC Structured String-Tree Correspondence

S-SSTC Synchronous Structured String-Tree Correspondence

TL target language

WSD word sense disambiguation

PENGGUNAAN VEKTOR KONSEPSI UNTUK MEMPERBAIKI KAEDAH PEMILIHAN TERJEMAHAN

ABSTRAK

Dalam bahasa tabii, sesuatu kata yang mempunyai makna yang berlainan adalah dikatakan *taksa*. Penyahtaksaan makna kata (WSD) merupakan suatu tugas untuk menentukan makna sebenar bagi sesuatu kata taksa dalam konteks. Ini merupakan satu tugas penting dalam aplikasi pemprosesan bahasa tabii (NLP) termasuk terjemahan berkomputer (MT), memandangkan kata dalam bahasa sasaran mesti dipilih supaya makna teks asal dapat disampaikan dengan tepat.

Penyelidikan ini bermatlamat untuk memperbaiki kaedah pemilihan terjemahan bagi satu sistem terjemahan berkomputer berasaskan contoh (EBMT) yang sedia ada dengan mengubah suai suatu algoritma WSD bagi tujuan khusus pemilihan terjemahan. Model vektor konsepsi (CV) digunakan untuk mewakili tema-tema konsep yang tersirat dalam bahan-bahan leksikal bagi membolehkan persamaan tematik antara dua bahan leksikal diukur dengan jarak sudut di antara CV yang dikaitkan dengannya.

Prosedur pemilihan terjemahan yang bakal dibangunkan memerlukan penyediaan data semantik daripada dua jenis sumber: kamus dan korpus terjemahan. Terlebih dahulu, makna-makna perkataan dari sebuah kamus bahasa sumber perlu dianotasi dengan label-label kategori am dari suatu hierarki konsep. Satu CV *profil* kemudian dihitung bagi setiap unit terjemahan dalam korpus terjemahan itu, dengan menggunakan label kategori

yang telah diberikan kepada makna-makna kata bahasa sumber tadi, dan juga konteks-nya di dalam korpus itu. Pada masa jalanan terjemahan, sistem EBMT akan menghitung CV *petunjuk* untuk setiap segmen teks input. Ia kemudian memilih terus daripada senarai unit-unit terjemahan, kata terjemahan dalam bahasa sasaran yang CV profilnya berhubung kait baik sekali dengan CV petunjuk.

Hasil ujian yang melibatkan penterjemahan teks Bahasa Inggeris ke Bahasa Melayu secara automatik menunjukkan bahawa penambahan pengetahuan semantik (dalam bentuk CV) telah membantu sistem EBMT menghasilkan output yang lebih berkemungkinan mengandungi terjemahan yang betul bagi kata taksa. Sistem baru juga dapat mengendalikan pemilihan terjemahan untuk kata isi dari semua golongan kata dengan baiknya, berbanding dengan sesetengah sistem lain yang hanya boleh mengendalikan kata nama dan kata kerja dengan kedah berlainan.

Dengan kejayaan percubaan pertama kami dalam menggabungkan analisis semantik dengan EBMT ini, hala tuju penyelidikan masa depan akan diarahkan kepada pengatasian kelemahan-kelemahan yang telah dikenalpasti termasuk keperluan tenaga manusia dalam penyediaan data dan juga penggabungan jenis-jenis maklumat semantik lain untuk mencapai ketepatan yang lebih tinggi.

USING CONCEPTUAL VECTORS TO IMPROVE TRANSLATION SELECTION

ABSTRACT

In natural language, a word having different meanings is said to be *ambiguous*. Word sense disambiguation (WSD) refers to the task of determining the correct meaning or sense of an ambiguous word in context, a crucial one in many natural language processing (NLP) applications. These include machine translation (MT), where words in the target language must be chosen such that the meaning of the original input text is correctly conveyed.

This research aims to improve translation selection in an existing example-based machine translation (EBMT) system by adapting a WSD algorithm specifically for translation selection. The Conceptual Vectors (CV) model is used to represent conceptual themes of lexical items, such that thematic similarity between two lexical items can be measured by the angular distance between their associated CVs.

The translation selection procedure requires semantic data to be prepared from two sources: a dictionary, and a translation corpus. Word senses from a source language dictionary are first annotated with general category labels from a conceptual hierarchy. A *profile* CV is then computed for each translation unit in the translation corpus, using these category labels assigned earlier to the senses of the source language words, and that of its context in the corpus. At translation run-time, the EBMT system computes *clue* CVs for each input text segment. It then selects directly, from the list of translation units,

translation words in the target language, whose profile CVs correlate best with the clue CVs.

Results from tests, in which English text is translated to Malay, show that this addition of semantic knowledge has enabled the EBMT system to produce outputs in which ambiguous words are more likely to be translated correctly. The improved system is also able to handle translation selection for content words of all parts-of-speech, rather than being limited to only nouns and verbs, using different approaches, as in some work by others.

With our first successful attempt at incorporating semantic analysis into the EBMT system, future work will be directed towards overcoming identified weaknesses, including the need for manual data preparation, and incorporating other types of semantic information to achieve higher accuracy rates.

CHAPTER 1

INTRODUCTION

This introductory chapter highlights the motivation and problem statement of this research, besides providing a "road-map" to the organisation of the chapters in this thesis.

1.1 Research Background

This research belongs in the domain of the use of computer systems for processing human languages, or what is called *natural language processing* (NLP). The word "natural" indicates languages that are spoken naturally by humans, and evolve dynamically as time passes. NLP encompasses a wide range of tasks, spanning speech synthesis and recognition, automatic translation, text summarisation, dialogue modelling, knowledge learning and management, and more.

Natural language is highly dynamic, flexible and expressive. It is, therefore, full of ambiguities, which presents much problem in NLP, including the translation of natural language texts.

1.1.1 The Dream of Machine Translation

Machine Translation (MT) refers to the application of computers to the problem of translation from one natural language to another (Hutchins, 1986). Automatic translation of texts and speech has long been a dream of humanity, as envisioned by *Star Trek*'s "universal translator", which made its first appearance in the TV series' episode *Metamorphosis*

in 1967 (Okuda *et al.*, 1999).

Fast forward to the present day. The need for MT has become increasingly acute, thanks to the onset of globalisation in almost every field imaginable, including socio-economy and technology. This is even more so with the immense amount of information brought about by the advent of the Internet. People of all nationalities need access to business reports, technical manuals, academic and technical documents, as well as news stories, all of which may not be available in their respective mother tongues.

Translating such material is often time-consuming, mundane, and requires consistency, but not tremendous professional effort of human translators. Rather, their expertise would be better spent on translating sensitive diplomatic documents, or cultural and literary works. Therefore, MT is seen to be able to fill in for human translators to speedily produce translations of domain-specific (often technical) documents (Hutchins, 1986; Hutchins and Somers, 1992).

It is a dream that has not quite been realised. Current MT systems have not been able to yield satisfactory translations: the outputs are often grammatically incorrect, or convey meanings that are entirely different from the original text. It is the latter problem that this thesis will address.

1.1.2 The Problem of Lexical Ambiguity

In natural language, a word with different meanings is said to be *ambiguous*. While it comes naturally to humans, deciding what an ambiguous word means in a particular discourse can be very problematic for machines. Consider the English word *log*. A computer might wrongly translate the English sentence

*The computer **logs** have been deleted.*

into the Malay sentence

* ***Balak***	*komputer*	*telah*	*dipotong.*
(timber	computer	already	been cut)

The problem of overcoming this lexical ambiguity is known as *word sense disambiguation* (WSD). It refers to the task of determining the correct meaning of an ambiguous word in context. This requires first establishing a list of all different meanings, or *senses*, for all the words under consideration. Disambiguation is then performed by evaluating the context of an occurrence of an ambiguous word, and comparing it against sense entries in the said list, in order to assign the correct sense to the ambiguous word instance under consideration (Ide and Véronis, 1998). The process of assigning sense labels from some sense repository to word occurrences in a text is also known as *sense-tagging*, which can be performed by either humans (manual tagging) or machines (automatic tagging). As word senses are typically given as a numbered list in a dictionary, the sense labels used in sense-tagging are usually the number identifiers of each sense, or more concisely *sense numbers*.

WSD is crucial in many NLP tasks, such as information retrieval, machine translation and speech processing. It is therefore unsurprising that so much research work has been (and is still being) done in this area, ever since the problem of lexical ambiguity was anticipated as early as 1949 by Warren Weaver (in Hutchins, 1986). Unfortunately, WSD remains largely unsolved to this day.

This thesis deals with WSD in the context of MT, i.e. that of translating ambiguous words. The selection of equivalent words in a target language to translate ambiguous words in a source language, as in the *log* example above, is often termed *translation selection*.

1.2 Research Overview

Example-based Machine Translation (EBMT) systems are MT systems that translate new input texts based on "translation examples" stored in a bilingual knowledge bank (BKB). This is a database which contains pairs of parallel text, where each pair is made up of an input text in a source language and its translation in a target language, both preferably annotated with some linguistic information. The main objective of this research is **to improve translation selection in an EBMT system, such that target language words, as appearing in the BKB, are chosen to correctly translate ambiguous words in the input text**.

Although this problem is closely related to the problem of WSD, the approach taken here is to disambiguate between *translations* in a target language, rather than between *sense numbers* of an ambiguous word in the source language, as set out by some dictionary or lexicon. This is because neither the *input word* $\leftrightarrow$ *sense number* relationship, nor that of *sense number* $\leftrightarrow$ *translation word*, is a simple, one-to-one mapping. This means even if an ambiguous word in some language A has been assigned a sense number, there may still be several possible words in the target language B, only one of which is acceptable as the translation of the input word. (To differentiate between the two situations, *word sense ambiguity* is used to refer to the case where a word has multiple meanings; while *translation ambiguity* indicates that a word has multiple translations in some chosen language.) Therefore, the path taken by this research is to "short-circuit" the two-stage *input word* $\rightarrow$ *sense number* $\rightarrow$ *translation word* resolving process, preferring instead to select a

translation word in language B for an input word in language A directly.

1.2.1 Research Outline

EBMT systems operate by retrieving translation fragments and related information from a BKB, as introduced in the previous section. (Chapter 3 describes the EBMT paradigm in more detail.) To achieve the above-mentioned research objective, the BKB needs to be enriched with semantic information, especially for ambiguous words.

Dictionaries are popular knowledge resources from which such semantic information about words can be extracted, since they are easily obtainable and contain much definitive knowledge about words, set down by lexicographers. On the other hand, recent advances in computing machinery have popularised the use of large amounts of corpora, such as articles in newspapers, magazines or webpages, in NLP tasks. Corpora contain actual usages of words that are not necessarily found in dictionaries, as it would require significant effort and expertise to exhaustively include all word usages in dictionary entries.

WSD approaches employing dictionaries, lexicons and thesauri are called *knowledge-based* approaches, while those drawing their knowledge from previously disambiguated corpora are known as *corpus-based* ones. This research intends to supplement the information extracted from dictionary sources with that gleaned from the corpus stored in the EBMT system's BKB. However, as both the dictionary definitions and the corpus are written in natural language, some NLP processing is required to extract the relevant information. This will involve (a) representing the meanings of lexicon sense entries by their related concepts, as well as (b) sense-tagging the BKB corpus. The resulting data will then be used to construct conceptual knowledge about the translation units in the BKB, i.e. translation pairs (between the source and target words) at the word- and phrase-levels.

Once the EBMT system's knowledge base is equipped with this knowledge (*profiles*) about words or phrases and their translations, it will be used by the EBMT system to select translation pairs with similar meanings to the fragments of an input sentence. To see how this works, let us view translation selection as a task involving some detective work. First, think of a crime scene, where a crime (say murder) has been committed, and the investigators need to pin it to one of several suspects. The circumstances and forensic evidences found at the scene often provide hints and clues to the criminal's *modus operandi*, behaviours and habits, aiding investigators to profile the criminal after analysing these circumstances and evidences (Meyer, 2000; Turvey, 2001).

Now consider an input sentence containing words with multiple possible translations as the "crime scene", the "crime" being translation ambiguity. There exist multiple words in the target language — the "suspects" — which can be used to construct the final translation output. To solve the "crime", i.e. translate the sentence, the EBMT system identifies the most probable "suspect" for each "crime" (ambiguous word occurrence), by gathering "clues" from the "crime scene" (the input sentence) and matching them against the "profiles" from a "database of suspects" (list of translation pairs in the database).

To summarise, this research has the following detailed objectives, which are necessary to achieve the main objective stated earlier:

- To represent the meanings of word senses from a lexicon using their related concepts;
- To sense-tag translation examples in the EBMT system's BKB;
- To compile "profiles" of translation units based on results from the first two objectives;
- To modify the EBMT system's matching and selection mechanism to use these "profiles" in selecting translation units that are closer in meaning to the input sentence

fragments.

The first three objectives belong to a *data preparation* phase, while the final objective is to be achieved during the *EBMT run-time*. While the run-time phase deals with translation ambiguities only (*word* ↔ *translation*), the data preparation phase is itself a two-stage process (see Figure 1.1). Entries from a lexicon are treated on a sense-number level (*word* ↔ *sense number*), whereas profiles are computed for *word* ↔ *translation* pairs from the translation example knowledge base, using the earlier *word* ↔ *sense number* level information.

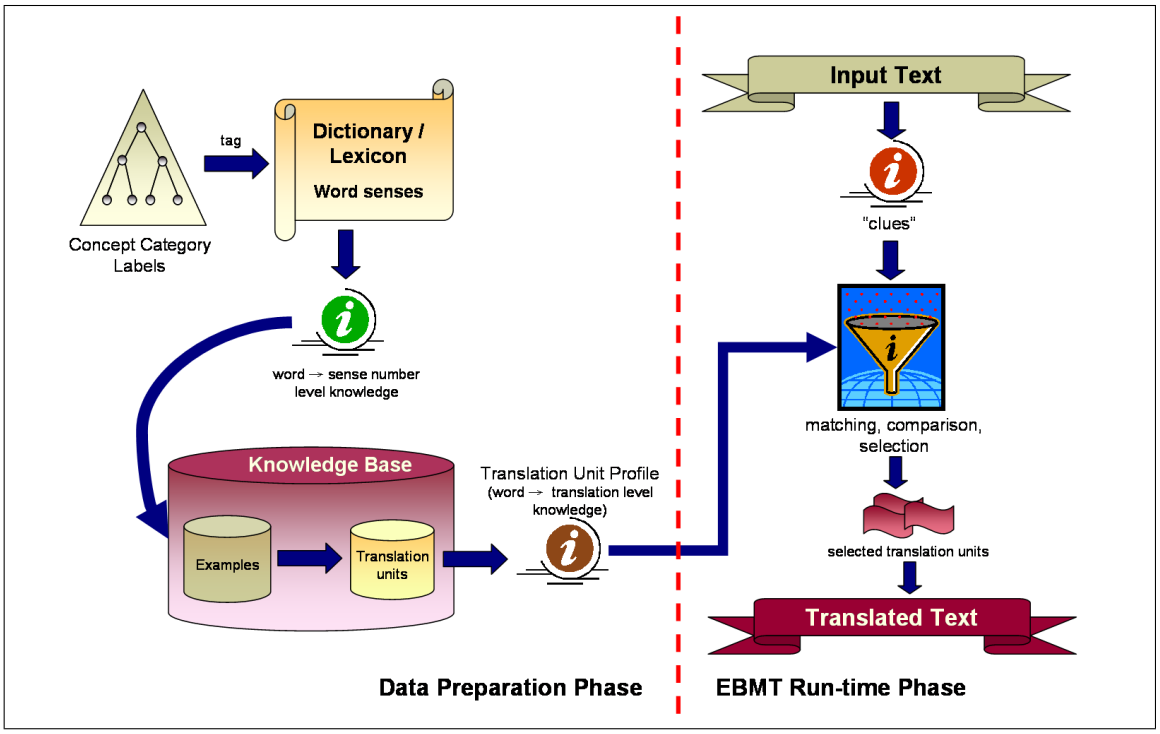


Figure 1.1: Outline of Data Preparation and EBMT Run-time Phases

At this point, some form of representation is required to encapsulate the conceptual knowledge about words. A model for representing themes or concepts related to lexical items, called the conceptual vector model (introduced in Chapter 4), is chosen for this purpose. Conceptual vector operations are conventional mathematical vector operations (with NLP interpretations), which are easy to implement and manipulate, and avoids

the combinatorial effect of multiple ambiguous words when used in a WSD approach. Some important ideas and operations on conceptual vectors are reviewed in Chapter 4. In Chapter 5, we propose guidelines for determining the appropriate concepts for senses of content words, i.e. nouns, verbs, adjectives and adverbs.

Test results show that the EBMT system has indeed benefited greatly from the addition of semantic knowledge for translation pairs, producing output in which ambiguous words in the input text are more likely to be translated correctly. Although the approach taken in this research is not without drawbacks, the biggest of which is the need for manual data preparation, there are already ideas as to how they can be overcome or avoided in future research.

1.2.2 Research Contributions

The contributions of this research are listed below:

- Adaptation of a WSD approach for the specific aim of translation selection, so that it is better suited for that purpose.
- Proposal of specific guidelines for determining related concepts for word meanings from dictionaries or lexicons.
- Application of the same concept hierarchy to the semantic tagging of nouns, verbs, adjectives and adverbs, instead of using different hierarchies for words of different POS, as practised by most of the current systems using hierarchies, e.g. *WordNet* (Miller *et al.*, 1990) and the *KSMSA* project (Ševčenko, 2004). We thus enable all content words to be disambiguated (both sense-number and translation ambiguity) in the same manner, as opposed to systems that deals only with nouns, e.g. (Resnik, 1995a; Agirre and Rigau, 1996; Jiang and Conrath, 1997; Hirst and St-Onge, 1998).

- Production of machine-usable information about word meanings on two different levels, i.e. the *word* ↔ *sense number* level (as found in dictionaries), and the *word* ↔ *translation* level (as found in texts and their translations).

1.3 Thesis Organisation

Having presented a brief overview about the research background, objectives and approach, the rest of this thesis is organised in the following chapters to report on this research:

Chapter 1 sets the stage for the research work undertaken.

Chapter 2 reviews past approaches to WSD and translation selection, and their influences on the direction adopted in this research.

Chapter 3 contains some background information about MT in general, and EBMT in particular. It also takes a look at the Synchronous Structured String Tree Correspondence (S-SSTC) annotation schema, which is used to annotate the BKB of the EBMT system that is to be improved, and which facilitates the overall translation process.

Chapter 4 reviews briefly the conceptual vector model, which represents semantic concepts related to lexical items with mathematical vectors. This includes a short description of mathematical vector operations involved, and their interpretations in NLP tasks.

Chapter 5 describes a methodology to enrich the knowledge base with semantic information, encoded with the conceptual vector model reviewed in Chapter 4. The chapter goes on to describe an algorithm that uses this semantic information to improve translation outputs of the EBMT system reviewed in Chapter 3.

Chapter 6 highlights some considerations and issues encountered during the implementation of the methodology design in Chapter 5.

Chapter 7 presents and discusses some experiment results from the improved EBMT system.

Chapter 8 sums up the research, and raises some issues that merits future work and research.

In addition, the **appendices** contain material on all prepared data and test results, as well as a short account of an index mapping procedure that was used during the construction of conceptual vectors.

As an alternative "roadmap", Figure 1.2 is a mind map showing the various aspects of this research, in which the corresponding chapter numbers are shown in small circles.

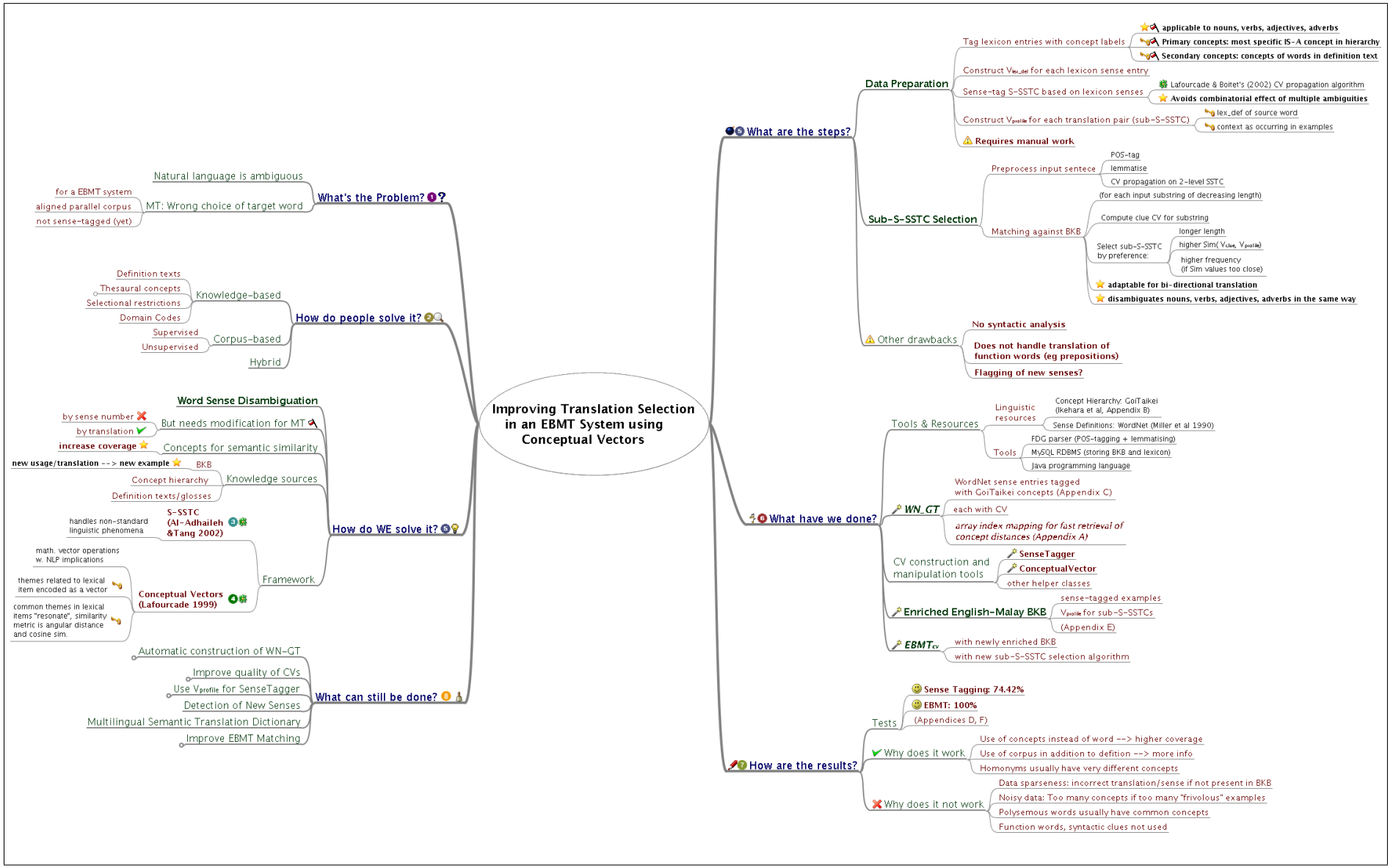


Figure 1.2: Thesis Contents as a Mind Map

CHAPTER 2

SURVEY OF WORD SENSE DISAMBIGUATION APPROACHES

A word sense disambiguation (WSD) procedure typically associates a particular occurrence of an ambiguous word in some context, with a particular sense from a lexicon (Wilks and Stevenson, 1997). As WSD has been a research area of much interest since the earliest days of machine translation, many different approaches have been developed and attempted. This chapter surveys the more recent and popular approaches to WSD and translation selection.

2.1 Role of Morphosyntactic Analysis

While WSD is largely concerned with semantic analysis, morphosyntactic analysis can provide very helpful information, some of which is described below. Such analysis is therefore often used as a pre-processing step in many systems.

The input text to be disambiguated are usually tokenised, and the words stemmed (or *lemmatised*) to their root forms. While this can usually be done safely for inflectional word forms (such as noun plurals and verb senses), derivational word forms often have different meanings from their root forms. Overzealous stemming, or over-dependence on root forms, will then lead to disambiguation error (Sanderson, 2000).

Part-of-speech (POS) tags are useful cues for WSD, as they can reduce ambiguities drastically (Wilks and Stevenson, 1998). For example, the word *handle* has five senses

as a verb, but only one sense as a noun in *WordNet* (Agirre and Martinez, 2001). Given the reliability and robustness of current POS taggers, POS-tagging is usually one of the first steps performed in WSD and widely used. This is also the reason why current WSD research is focused on ambiguous words from the same POS (Ide and Véronis, 1998), and typically on content words.

Syntactic dependency and relations between an ambiguous word and its context, resulting from syntactic analysis, are also often used. This includes whether a verb is transitive or intransitive, as well as various syntactic relations, such as *subj-verb, verb-obj, adj-noun* and others.

Nevertheless, morphosyntactic analysis alone is insufficient to handle all ambiguities. The next section will describe some of the more popular WSD approaches, taking different routes to tackle the semantic analysis required.

2.2 WSD Approaches

WSD usually involves comparing learned knowledge about an ambiguous word and the information from its surrounding input context. Therefore, all approaches to WSD and translation selection need some kind of knowledge source to "learn" from. WSD approaches may hence be broadly classified into two categories depending on the type of source used: knowledge- and corpus-based. Hybrid approaches that combine both learning sources also exist, and the *SENSEVAL* competitions (SENSEVAL, 2005) provide both types of resources for participants to train their systems.

2.2.1 Knowledge-based Approaches

Knowledge-based disambiguation approaches extract knowledge from lexical resources like dictionaries and thesauri. This was motivated by the fact that hand-crafting rules (morphological, semantic or otherwise) for each word is very expensive. However, much of these information is already set down in dictionaries by lexicographers — albeit written in natural language, therefore requiring prior processing by computers to extract the implicitly "encoded" (in natural language) knowledge. As research interest in semantic web technologies became more widespread in recent years, an increasing number of researchers also make use of ontologies in WSD efforts.

2.2.1(a) Dictionary Definitions

Dictionary definitions (usually taken from machine readable dictionaries, or MRDs) of senses have been used in different ways in various WSD work. Lesk (1986) counted overlapping words in the definition for senses of an ambiguous word w, with the definitions of the context surrounding w, to determine the most probable sense. One problem with this method is that if the input text contained multiple ambiguous words, the algorithm would have to test all word sense combinations: an operation that is very computationally expensive. This was solved by Cowie *et al.* (1992) using a simulated annealing technique. Lesk's WSD method was further improved by Wilks and Stevenson (1997) to compensate for the bias towards senses with longer definitions, while Wilks *et al.* (1993) and Pedersen *et al.* (2005) used "gloss vectors" and the cosine of the angle between them to measure the degree of overlapping.

Other researchers used dictionary definitions in different ways in their WSD endeavours. To make dictionaries more machine-tractable, Wilks *et al.* (1993) extracted a natural set of semantic primitives, i.e. words in the definition found to be "atomic" using the

"defining cycle" method, from the *Longman Dictionary of Contemperary English* (LDOCE, 2003). Lim (2003) applied a similar procedure on *WordNet* (see next subsection), then used the overlaps of semantic primitives among *WordNet* senses to derive a measure of relatedness between senses. The combinatorial effect of multiple word senses during a WSD process was overcome using a dynamic programming technique.

Elsewhere, Shirai and Yagi's (2004) WSD model learns hypernyms (`is-a` relations) from definition sentences. Gaume *et al.* (2004) used definitions to build a graph representing a whole dictionary, an approach that resembles the use of semantic networks in §2.2.1(b).

While dictionary definitions are undoubtedly invaluable resources for WSD, there is only so much information that can be contained in a short definition sentence. In addition, Lesk-like measures of overlapping words depend much on how the definitions are worded (Levow, 1997). WSD approaches that use dictionary definitions would do well to complement it with information from other sources.

2.2.1(b) Edge-counting in Semantic Hierarchies and Networks

The electronic lexical database *WordNet* (Miller *et al.*, 1990) became an extremely popular resource for WSD work soon after its introduction. *WordNet* groups synonym words in "synsets", which are organised differently depending on their parts-of-speech (POS). For example, the noun synsets constitute an `is-a` hierarchy, while adjective synsets form clusters with a axis–satellite structure.

Given taxonomical hierarchies such as that of the *WordNet* noun synsets, many WSD researchers compute the semantic similarity between two word senses using the shortest path length between the two corresponding nodes in the hierarchy, or the shortest path

length to their least common subsumer (Sumita and Iida, 1991; Wu and Palmer, 1994; Leacock and Chodorow, 1998). These simple metrics were further enhanced by others by taking into consideration the conceptual density in a particular region of the hierarchy (Agirre and Rigau, 1996), or information content of the two concepts, i.e. how specific they are (Jiang and Conrath, 1997; Lin, 1998; Resnik, 1995b, 1999).

However, one problem with edge-counting in this way is that concept similarity, which uses only `is-a` relations, may be too restrictive: *labourer* and *strike action* are related, but are in subtrees that are far apart in an `is-a` hierarchy (under HUMAN and EVENT respectively). Researchers started to look for *relatedness*, where paths between word senses can cut across regions of hierarchies and even POS.

As the synsets in *WordNet* are also linked by many other semantic relations (e.g. `has-part`, `member-of` and many more), Sussna's (1993) and Hirst and St-Onge's (1998) measures of relatedness incorporate such links as well. Elsewhere, Gaume *et al.* (2004) built a graph representation of a whole dictionary, using word occurrence in the sense entry definitions, and the hierarchy of sub-senses.

More elaborate and expressive semantic networks became available with the recent research interest in knowledge engineering and ontology technologies. Ontologies, which are richer than mere taxonomies, have been developed for various domains, including medicine, bioinformatics, military, and also NLP.[1] Examples of NLP-targeted ontologies include the *Suggested Upper Merged Ontology (SUMO)* (Niles and Pease, 2001; Pease, 2005) and the *OpenCyc* project (Reed and Lenat, 2002; OpenCyc, 2005), which has attracted some researchers' attention to incorporate them into WSD models (Legrand *et al.*, 2003; Navigli

[1] See `http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary` and `http://protege.stanford.edu/plugins/owl/owl-library/` for a selection of existing ontologies, and `http://protege.stanford.edu/community/conferences.html` for example applications that make use of ontologies.

and Velardi, 2005).

Since semantic hierarchies are primarily about concepts, much of the *WordNet*-related approaches concentrate on nouns only. Currently, most existing NLP-related ontology resources have only established the hierarchical structures, or classifying words of different POS using different sets of concepts, e.g. the *KSMSA* project (Ševčenko, 2003, 2004). The various ontological *properties* of the instances, which are more likely to involve verbs, adjectives and adverbs, have yet to be filled in. This often leads to the disambiguation of only nouns, as words of different POS are associated with very different sets of concepts. However, disambiguation of verbs, adverbs and adjectives is possible when semantic hierarchies are used in conjunction with selectional restrictions (see next subsection), or by using a separate, different semantic hierarchy for verbs. For example, Resnik and Diab (2000) looked into how similarity between verbs can be measured, *vis-à-vis* the verb hierarchy in *WordNet*. On the other hand, concept hierarchies provided by thesauri e.g. *Roget's Thesaurus are* applied to words of different POS, although *Roget*'s concept hierarchies, which were designed in late eighteenth century Victorian England, might be awkward to work with today.

2.2.1(c) Selectional Restrictions

Semantic hierarchies are also used in conjunction with selectional restrictions (also known as semantic formulae) in some WSD work, especially in the context of transfer-based MT systems (McRoy, 1992; Wilks and Stevenson, 1998; Viegas *et al.*, 1999; Bond, 2001). Selectional restrictions specify semantic class types of nouns that can be taken on as "arguments" by verbs and adjectives. Example phrase patterns with selectional restrictions for the Japanese verb 取る *toru* from *GoiTaikei* (Ikehara *et al.*, 1999), the semantic dictionary used in (Bond, 2001), look like the following:

1. **J:** N1がN2を取る。	**E:** *N1* ***have*** *N2.*	N1: AGENT; N2: EATING EVENT
2. **J:** N1がN2を取る。	**E:** *N1* ***subscribe to*** *N2.*	N1: AGENT; N2: PUBLICATION

In the following input Japanese sentence:

私 が	雑誌 を	取る。
(*watashi ga*	*zasshi o*	*toru*)
I	magazine	**have? / subscribe?**

According to *GoiTaikei*'s semantic hierarchy, 私*watashi* (I) is subsumed by AGENT, while 雑誌*zasshi* (magazine) is subsumed by PUBLICATION. Pattern 2 is therefore found to be more applicable here, and 取る*toru* is translated as *subscribe to*. Some researchers (Resnik, 1997; McCarthy *et al.*, 2001) have also taken a statistical approach to selectional restrictions.

The disadvantage of the selectional restriction approach is in the expertise and time required to hand-craft a small set of rules, let alone achieving one of reasonable coverage. Although this can be alleviated by automatic learning using machine learning methods (Allmuallim *et al.*, 1994; Resnik, 1997; Ciaramita and Johnson, 2000), the approach itself can be cumbersome (Somers, 1999): see the descriptions in (Viegas *et al.*, 1999) and (Bond, 2001, §3.2.1).

2.2.1(d) Subject Codes

Some dictionaries and thesauri, such as *LDOCE* and *Roget's Thesaurus*, use subject codes to annotate domain-specific keywords, which can be useful for WSD purposes, especially for technical documents. The adjective *high* can then be resolved to mean a high tone in the MUSIC domain; or being high on drugs in the MEDICAL domain; or a high pressure area in METEOROLOGY (Buitelaar, 2001).

WSD approaches that use subject codes include work by Wilks and Stevenson (1998) and Magnini *et al.* (2001), who manually annotated *WordNet 1.6* synsets with about 200 subject field codes in (Magnini and Cavaglià, 2000). Annotation with subject codes is possible across different POS and concept hierarchy regions: an advantage over the edge-counting methods described in §2.2.1(b). Unfortunately, they are not as useful when disambiguating non-technical text (Magnini *et al.*, 2002).

2.2.2 Corpus-based Approaches

As with the development in other NLP research areas, advances in computer processing speed, storage capacity and the World Wide Web has made the use of large amounts of corpora feasible as WSD learning sources. Corpus-based methods avoid the human effort required in building knowledge sources, and many of them employ numerical or statistical models in their training phases, which are more empiric in nature. This is in contrast with knowledge-based methods which place more emphasis on formal linguistic rules (Ide and Véronis, 1998).

2.2.2(a) Supervised Training

WSD approaches with a supervised training phase construct disambiguation models from sense-tagged corpora using various learning methods, such as Bayesian classifiers (Leacock *et al.*, 1993; Chao and Dyer, 2001), k-th nearest neighbour (Ng and Lee, 1996) and support vector machines (Cabezas *et al.*, 2001; Lee *et al.*, 2004).

Yarowsky's (1993) work is considered to best exemplify the "classic" supervised learning paradigm (Cabezas *et al.*, 2001), in which the training procedure calculates the probability distribution of word senses for collocations $Pr(Sense|Collocation)$ in sense-tagged corpora. This work led to the famous "one sense per collocation" observation, which formed the basis to many subsequent corpus-based WSD approaches.

The most common complaint about supervised WSD approaches is the difficulty of manually sense-tagging a training corpora (Ide and Véronis, 1998). While this problem could be alleviated somewhat by bootstrapping from a small hand-tagged sample (Hearst, 1991; Basili *et al.*, 1997), other researchers looked elsewhere for solutions.

2.2.2(b) Unsupervised Training

To reduce the effort needed for manual tagging, Schütze (1992; 1998) first clusters words in unprocessed training text, then assigns senses to the *clusters*, rather than to individual word occurrences. Elsewhere, following from the "one sense per discourse" (Gale *et al.*, 1992) and "one sense per collocation" (Yarowsky, 1993) heuristics, Yarowsky (1995) showed that it was possible to use raw, unprocessed corpora as WSD training material (although the initial samples still need to be clustered).

A different approach makes use of parallel text (Resnik and Yarowsky, 1997; Ide, 1999;

Ide *et al.*, 2002; Ng *et al.*, 2003), where translations of ambiguous words are regarded as sense distinctions. As it is not always possible to acquire parallel corpora, independent second-language corpora have also been shown to be suitable for training WSD models, either by purely numerical means (Brown *et al.*, 1991; Fung and Lo, 1998; Kaji and Morimoto, 2002; Li and Li, 2004) or with the aid of syntactic relations (Dagan and Itai, 1994; Zhou *et al.*, 2001; Kim *et al.*, 2002). Such cross-lingual resources (or simply target-language co-occurrences) are especially suitable for translation selection, which has direct consequences for machine translation systems.

Another major obstacle for corpus-based approaches (both supervised and unsupervised) is the problem of data-sparseness. Information about word senses that are not frequently used in corpora might go unnoticed or under-observed in the WSD models. Schütze (1992; 1998) uses the idea of "second-order co-occurrence" to overcome this problem, where the overall contexts of words in a target context are captured (see §4.1). Another popular solution is to use class-based (or concept-based) methods, where observations of words belonging to some common category are combined. Such classes of words can be derived from the statistical properties of the corpora itself (Brown *et al.*, 1992; Pereira *et al.*, 1993), or taken from external resources, thereby resulting in hybrids of both knowledge- and corpus-based approaches, which are discussed in the next subsection.

2.2.3 Hybrid Approaches

Knowledge- and corpus-based WSD (and translation selection) approaches are often combined to gain the best of both worlds. For example, Lee and Kim (2002) combined scores for sense preference from dictionaries, and word probability from a target language corpus for their work in translation selection. Many researchers use both semantic hierarchies and corpora to overcome data-sparseness of purely corpus-based methods, by grouping words

of similar meanings under common categories in the semantic hierarchy. Works that adopt such an approach include (Yarowsky, 1992; Resnik, 1995a; Leacock *et al.*, 1998; Stevenson and Wilks, 2001; O'Hara *et al.*, 2004; Tufiş *et al.*, 2004).

2.3 Discussion

Several observations can be made from the previous descriptions of various WSD approaches:

- Dictionary definitions contain concise and "authoritative" information about word senses. However, definition text alone cannot be expected to capture relevant information exhaustively (Hearst, 1991). The complementary use of corpora can provide additional collocating information, since they represent actual usages of word senses "on the ground".

- Surface-level lexical matching, either in approaches similar to Lesk's (1986) overlaps of definition words or observations of collocation in corpora (Yarowsky, 1993), often suffer from coverage and data-sparseness problems. This can be overcome by adopting a class-based model, where words of similar semantic concepts are grouped under common categories. Several lexical knowledge resources provide such concept labels, usually arranged in a hierarchy, including *Roget's Thesaurus*, *WordNet* and *GoiTaikei*.

- Most semantic hierarchies organise concepts in taxonomy-like structures. Consequently, most WSD approaches using such resources (most notably those using *WordNet*) only apply the concept labels to nouns, thereby leading to disambiguation of nouns only (Agirre and Rigau, 1996). Disambiguation of verbs, adjectives and adverbs are usually done with hierarchies for verbs, selectional restrictions, or

other strategies. In other words, WSD for words of different POS is undertaken with different methods.

- In general, measures of *relatedness* between word senses gives higher coverage than measures of *similarity*.

These observations lead to the conclusion that a WSD approach combining knowledge- and corpus-based learning sources is promising. For higher coverage, a relatedness metric between word senses should be used. This metric can be derived by applying a class-based model to Lesk's idea of overlapping definitions, thus forming the basis to the methodology that will be chosen in this research.

We do not opt for a statistical modelling method as it does not sit well with the MT framework that we seek to improve (see Chapter 3). We are also interested to see how the same semantic hierarchy can be used for content words of all POS, such that their disambiguation can be done with essentially the same strategy.

In general, it is difficult to evaluate and compare WSD systems (Ide and Véronis, 1998), as their accuracy is relative to the lexical repository and evaluation corpus being used (Wilks and Stevenson, 1997, 1998). If the lexicon used does not list a particular sense of an ambiguous word, the WSD system should not be held wrong for being unable to select it. The choice of test words may also have a big effect on the accuracy results. There are two types of lexical ambiguity: homonymy and polysemy. *Homonyms* are words spelled the same way, but have unrelated meanings (e.g. *money* ***bank*** and *river* ***bank***), while *polysemes* have multiple related meanings (e.g. ***high*** *air pressure* and ***high*** *musical note*). It is usually easier to disambiguate homonyms than polysemes.

The *SENSEVAL* Workshops (SENSEVAL, 2005) attempt to address this evaluation problem by organising WSD competitions, in which the training resources, test data and scoring methods are standardised. Participants are provided with a sense-tagged training corpus and asked to disambiguate the same evaluation corpora, based on *WordNet* senses. Top-ranking WSD systems typically report accuracy rates of over 90%.

SENSEVAL Workshops evaluate WSD systems in isolation with special benchmarks, what Ide and Véronis (1998) termed *in vitro* evaluation. However, as mentioned earlier, WSD is essentially an intermediate task to many other NLP applications, all of which require different degrees of granularity. For example, several sense entries of the same word may have the same translation in a target language, thereby reducing (at times eliminating) the required disambiguation effort. *In vivo* evaluations of WSD methods consider their contribution to the improvement of a particular application. This is in contrast to *in vitro* evaluations, which can be pedantic at times, especially if the sense distinctions are very fine-grained.

This research approaches WSD in the context of MT (as translation selection), in the belief that this is more beneficial and has direct, pragmatic consequences to an actual NLP application. This also implies that we may need to adapt WSD approaches for other specific NLP tasks.

2.4 Summary

WSD approaches can be broadly categorised into knowledge- and corpus-based methods. The two approaches need not be mutually exclusive, and this research will adopt a hybrid approach of both by using dictionary definitions, concept hierarchies and parallel text. This research will also address WSD in the context of translation selection in MT, since

the outputs and results, having consequences in an actual NLP application, will be easier to evaluate from a practical perspective. To understand further where WSD stands in MT, specifically EBMT, the next chapter describes the EBMT system that this research seeks to improve.

CHAPTER 3

EXAMPLE-BASED MACHINE TRANSLATION AND THE SYNCHRONOUS STRUCTURED STRING-TREE CORRESPONDENCE ANNOTATION SCHEMA

This chapter will describe the Example-Based Machine Translation (EBMT) system which we seek to improve. We start by introducing past MT approaches in general and the more recent EBMT paradigm. This is followed by a brief review about the Structured String-Tree Correspondence (SSTC) and Synchronous SSTC (S-SSTC) annotation schemas. We then look into how the EBMT system in question uses S-SSTC-annotated translation examples to produce translation outputs. Finally, we discuss the current weaknesses of this system in translating ambiguous words in the input text, and how it can be improved.

3.1 Past Approaches of Machine Translation

Machine Translation (MT) refers to the (sometimes semi-) automatic translation of natural language texts by computerised systems, from a *source language* (SL) to a *target language* (TL) (Hutchins, 1986). Hutchins and Somers (1992) classifies past MT approaches into *first generation systems* using a "direct" approach, and *second generation systems* adopting an "indirect approach", where the main difference is the level of SL text analysis involved. This is summarised in the Vauquois pyramid (Vauquois, 1968, see also Figure 3.1 on the next page).

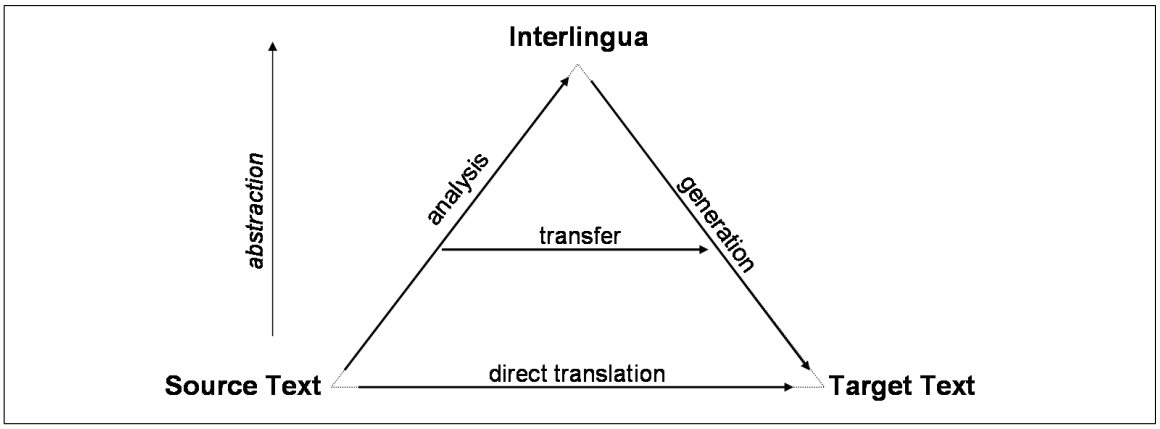


Figure 3.1: The Vauquois Pyramid (modified from Hutchins and Somers, 1992, p. 107)

3.1.1 Direct Approach

Early attempts at MT in the 1960s used a *direct* approach, in which the SL text was transliterated directly, word-for-word, following some morphological analysis, a dictionary look-up and some word-reordering (Hutchins and Somers, 1992). No structural analysis of the SL text is done. This approach was soon found to be unsatisfactory, in that it both produced grammatically incorrect outputs, as well as failed to handle the translation of ambiguous words correctly.

3.1.2 Indirect Approach

Next came the *indirect* approaches, in which the SL text is analysed to some intermediate representation, before the final translation is produced. Its two main variants are the *interlingua* approach and the *transfer* approach (*ibid.*).

3.1.2(a) Interlingua Approach

In interligua-based MT, the SL text is first analysed to an abstract representation that is language-independent, i.e. the *interlingua*. The TL translation is then generated from this interlingua representation, without any reference to the SL text.

This approach was deemed attractive and elegant, as it separated the analysis phase completely from the generation phase, thus enabling the two phases to be implemented in a modular fashion. In addition, given n languages, the interlingua approach requires the development of only $O(n)$ analysis and generation modules (one analysis module and one generation module per language) to perform bi-directional translation, compared to $O(n^2)$ modules (one per language *pair* per direction) for the direct approach.

Unfortunately, there is one major hurdle: the definition of an interlingua that is truly language-independent and universal. It is very difficult to develop an interlingua that captures semantic nuances across cultures and languages, such that it can be a "bridge" between any two languages.[1] Moreover, a completely language-independent intermediate representation means that all language-dependent features (including syntactic and morphological information) of the source text, is discarded, when in fact such surface information is often necessary to generate the target text correctly.

3.1.2(b) Transfer Approach

The transfer approach is less ambitious than the interlingua approach. Rather than going all the way to a language-independent abstraction, the SL text is only analysed "half-way" to produce a SL-dependent representation, with some relevant surface information still intact. A *transfer module* then *transfers* this representation to a TL-dependent version, from which the translation is generated.

While the number of modules needed in a transfer-based MT system is $O(n^2)$, the complexity of the analysis and generation modules is much lower than that of interlingua systems, since the intermediate representations here are not as far removed from the

[1] See (Sérasset and Boitet, 2000) for an account of the Universal Networking Language, an on-going interlingua project.

surface texts. It also avoids the need to formulate a universal representation that needs to capture all linguistic phenomena and semantic information. In addition, the retention of surface information from the SL text aids the correct generation of TL translation.

3.2 Example-Based Machine Translation

Earlier approaches to MT required linguistic rules, including morphological, syntactic and at times semantic frames, to be hand-crafted by linguistic experts, which can be expensive. With the cheap availability and abundance of computer memory and storage since the late twentieth century, researchers started to look into corpus-based methods to solve NLP problems. Nagao (1984) proposed the "translation by analogy" principle, or what has since become known as *example-based* machine translation (EBMT). Its most prominent characteristic is the use of a database of translation examples, i.e. texts in different languages that are translations of each other, to translate new inputs. The essence of EBMT is best described by Nagao himself:

> Man does not translate a simple sentence by doing deep linguistic analysis, rather, Man does translation, first, by properly decomposing an input sentence into certain fragmental phrases [...], then, by translating these fragmental phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference [...]. (Nagao, 1984)

There are thus three main stages in an EBMT process (*ibid.*, Somers 1999):

1. **matching** fragments against database of examples,

2. **identifying** or selecting corresponding translation fragments, and

3. **recombining** fragments to produce the target text.

An important requirement in the EBMT paradigm is a parallel aligned corpus, where the alignments or correspondences between the SL and TL texts are specified. Annotated tree structures were popular for storing examples in early attempts, including work by Sato and Nagao (1990), Al-Adhaileh and Tang (1999) and Wong *et al.* (2004). Later research sought to mine generalised, template- or pattern-like structure pairs from the translation examples (McTait, 2003).

3.3 An Annotation Schema for Specifying Correspondences

Al-Adhaileh and Tang (1999) proposed a flexible annotation schema called the Synchronous Structured String-Tree Correspondence (S-SSTC), which they used to annotate examples in their EBMT knowledge base. Ye (forthcoming) performed structural indexing on this knowledge base, to allow for faster and more accurate retrieval of S-SSTC substructures during the translation process.

The S-SSTC schema is a synchronisation of two SSTCs, and describes the relation between the two SSTCs at different levels. Its advantage lies in its flexibility to handle non-standard correspondences between natural languages. We will describe these annotation schemas in the next two subsections.

3.3.1 Structured String-Tree Correspondence (SSTC)

An SSTC is a general structure that can associate a string in a language to an arbitrary tree structure, as chosen by the annotator to be the interpretation structure of the said string (Boitet and Zaharin, 1988). This choice of an arbitrary tree structure, together

with the facility to specify non-projective correspondences between the string and the tree, are especially desirable in handling non-standard linguistic phenomena, e.g. cross-dependencies (Tang and Zaharin, 1995).

The following definitions are taken from (Al-Adhaileh *et al.*, 2002):

- An **SSTC** is a triple (st, tr, co) where st is a string in one language, tr its associated representation tree structure, and co the correspondence between st and tr.
- The correspondence co is made up of two interrelated correspondences:
  (a) between nodes and (possibly discontinuous) substrings, and
  (b) between (possibly incomplete) subtrees and (possibly discontinuous) substrings.
- co is encoded on the tree by attaching a pair of **interval sequences**, SNODE and STREE, to each node n in the tree:
  (a) SNODE(n) is the sequence of intervals of the substring in st that corresponds to n in tr, and
  (b) STREE(n) is the sequence of intervals of the substring in st that corresponds to the subtree in tr having n as its root.

Each node in the tree can be further annotated with other relevant information, including POS, morphological information and sense number.

Figure 3.2 shows an SSTC recording the correspondences between the sentence *The old gardener watered the flowers* and its dependency tree. Here, the node with SNODE interval 2_3 corresponds to the substring *gardener*, while the subtree rooted at the node with STREE interval 4_6, corresponds to the substring *the flowers*.

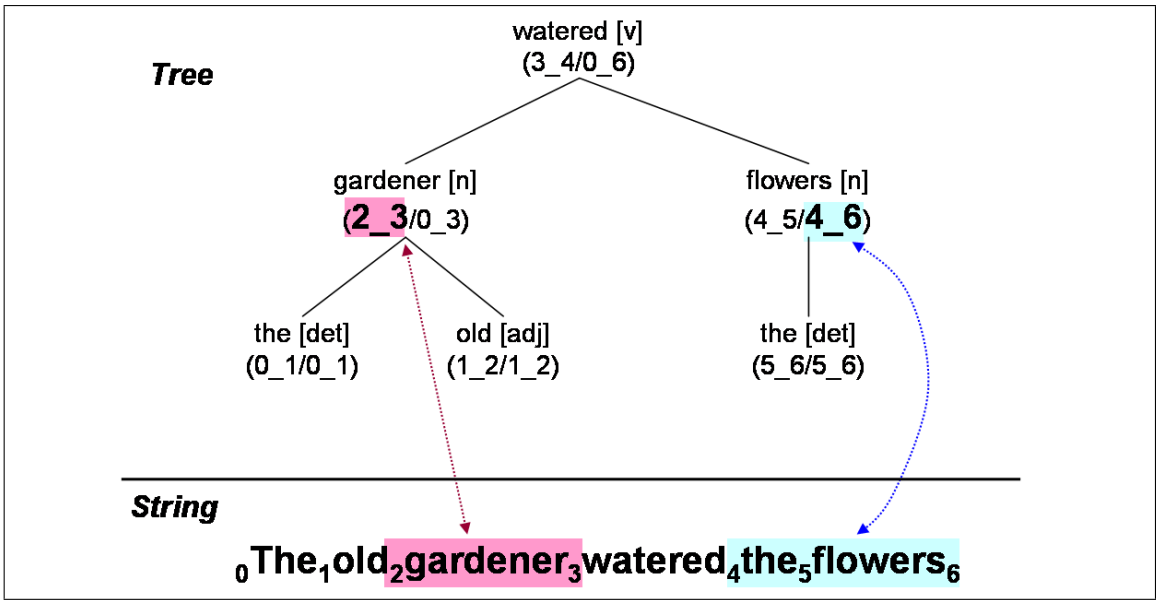


Figure 3.2: An SSTC recording the correspondences between the sentence *The old gardener watered the flowers* and its dependency tree.

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

While SSTC describes the relation between a text and its representation structure in one language, the S-SSTC annotation schema specifies the correspondences on different levels (i.e. lexical and structural) *across languages*. This proves useful for the purpose of marking up the alignment between the SL and TL text in a translation example. Al-Adhaileh *et al.* (2002) gives the following definitions:

- An **S-SSTC** is a triple $(S, T, \varphi_{(S,T)})$, where S and T are SSTCs as defined in the previous section, and $\varphi_{(S,T)}$ is a set of links defining the synchronous correspondences between S and T at different internal levels.

- A synchronous correspondence link $\ell \in \varphi_{(S,T)}$ can be either of type ℓ_{sn} or ℓ_{st}:

  (a) $\ell_{sn} = (X_1, X_2)$ records the synchronous correspondences between S and T at the node level (i.e. lexical correspondences). X_1 and X_2 are sequences of SNODE intervals from co_S and co_T respectively.

  (b) $\ell_{st} = (Y_1, Y_2)$ records the synchronous correspondences between S and T at

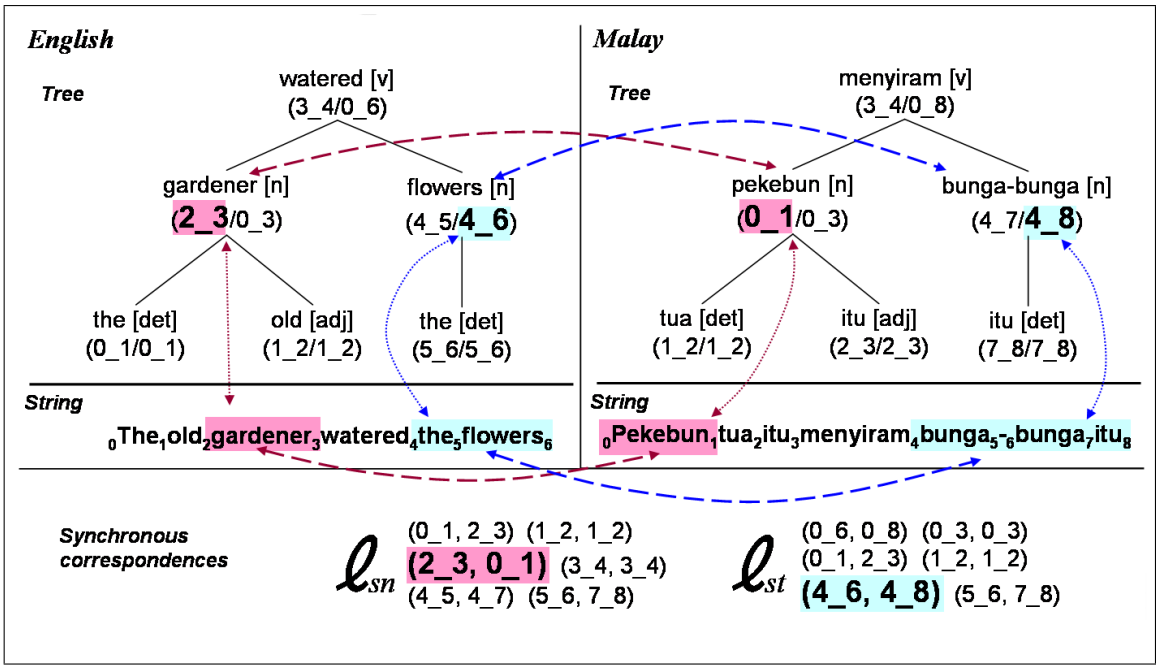


Figure 3.3: An S-SSTC for the English sentence *The old gardener watered the flowers* and its Malay translation *Pekebun tua itu menyiram bunga-bunga itu*.

the subtree level (i.e. structural correspondences). Y_1 and Y_2 are sequences of STREE intervals from co_S and co_T respectively.

Consider the S-SSTC given in Figure 3.3, for the English sentence *The old gardener watered the flowers* and its Malay translation *Pekebun tua itu menyiram bunga-bunga itu*. Its synchronous correspondence pairs, ℓ_{sn} and ℓ_{st}, specify the lexical and structural correspondences between the two sentences. For example, the ℓ_{sn} pair $(2\_3, 0\_1)$ expresses the fact that the English *gardener* corresponds to the Malay *pekebun*, while the alignment between the noun-phrases *the flowers* and *bunga-bunga itu* is recorded by the ℓ_{st} pair $(4\_6, 4\_8)$. For more examples of S-SSTCs, especially in handling non-standard linguistic phenomena, see (Al-Adhaileh and Tang, 2002).

3.4 An EBMT System Based on the S-SSTC

Al-Adhaileh and Tang's (1999) EBMT system draws on a database of annotated parallel text to translate new inputs. They created a database of English-Malay translation

examples (Tang and Al-Adhaileh, 2001), known as a Bilingual Knowledge Bank (BKB),[2] annotated in the S-SSTC schema, using dependency trees as the linguistic representation structure for the strings. Each annotated example is similar to the S-SSTC shown in Figure 3.3, with extra morphological information tagged to each node. To improve the translation output quality, Ye (forthcoming) indexed the BKB based on the S-SSTC structures, thereby making possible the accurate retrieval of sub-S-SSTCs, i.e. matching at the sub-sentence level.

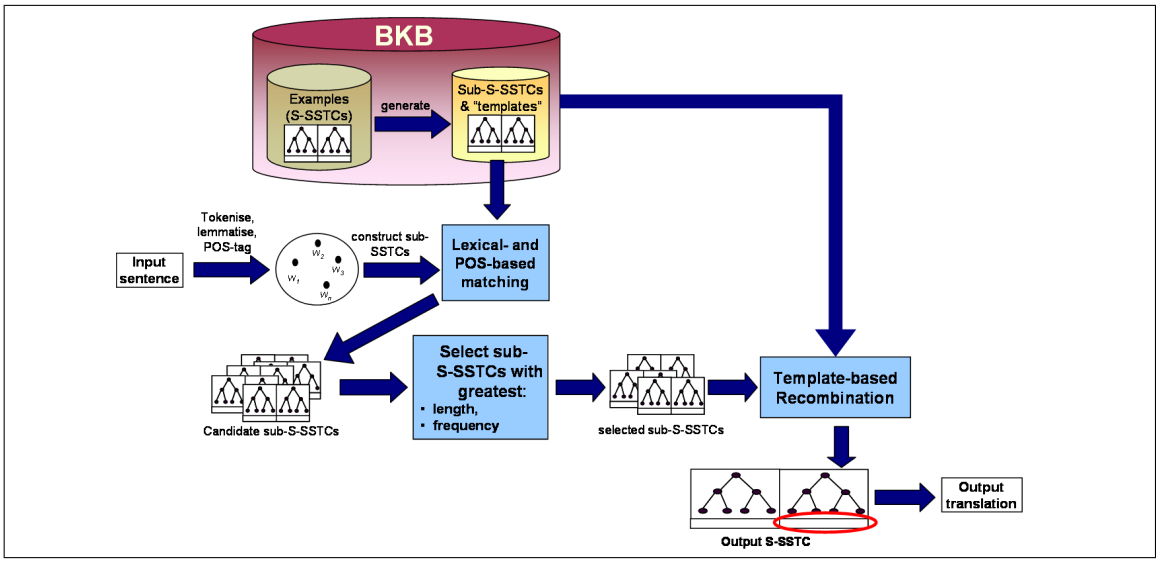
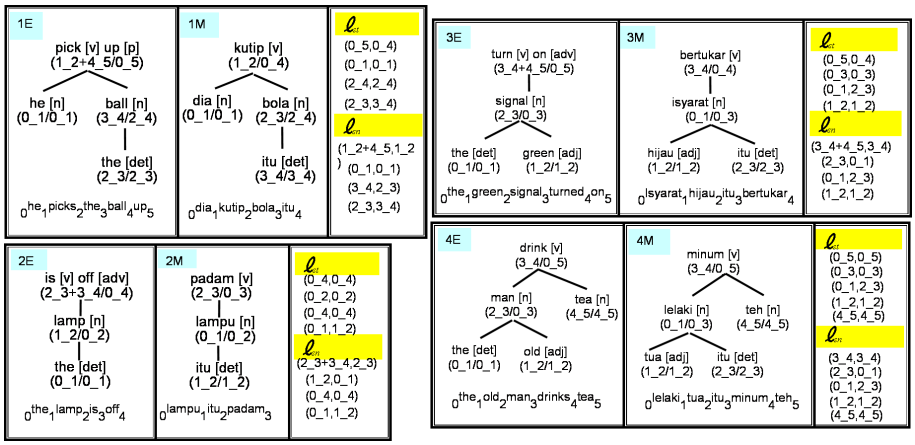


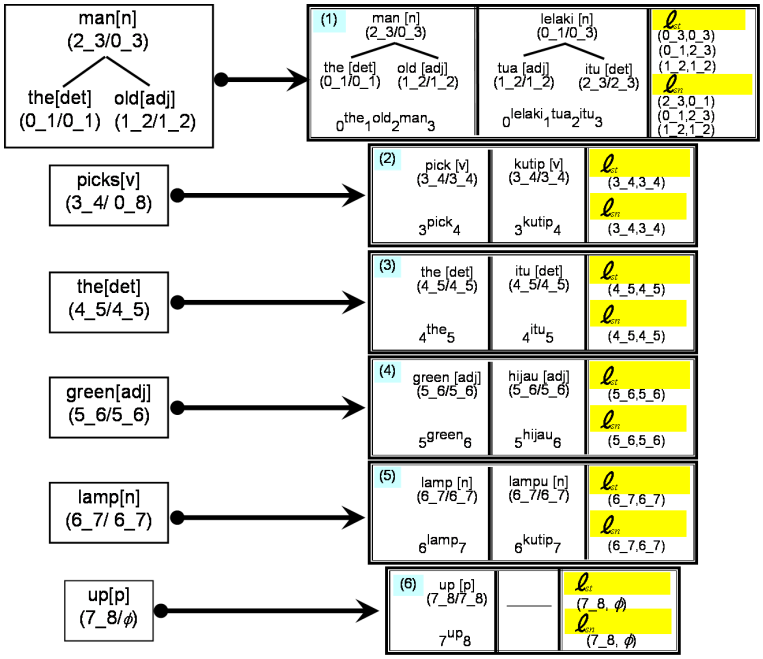
Figure 3.4: EBMT System Based on an S-SSTC Annotated BKB

A typical translation process is shown in Figure 3.4. After tokenising, POS-tagging and lemmatising the input sentence, the system constructs sub-SSTCs from these processed tokens, and retrieves sub-S-SSTCs from the BKB via lemma- and POS-based matching. A list of "best" sub-S-SSTCs are selected, with preference given to those with longer string lengths (i.e. greater coverage), and those that occur more frequently in the BKB, in that order. Finally, by referring to the appropriate generalised templates, the system constructs a complete S-SSTC from this list of sub-S-SSTCs. The translation of the input sentence is simply the string of the target language SSTC of the resulting S-SSTC.

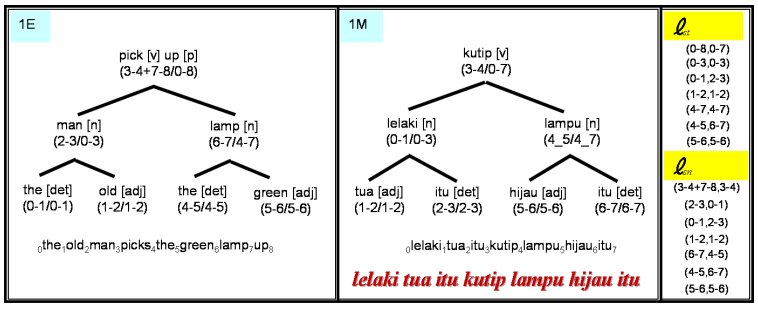
[2]defined in (Sadler and Vendelmans, 1990) as a "syntactically and referentially structured pair of corpora, one being a translation of the other, in which translation units are cross-coded between the corpora".



(a) Set of S-SSTCs in the BKB



(b) Matching sub-SSTCs from the input *The old man picks the green lamp up* against sub-S-SSTCs in BKB



(c) Final Ouput S-SSTC after Template-based Recombination

Figure 3.5: Translating *The old man picks the green lamp up* (modified from Al-Adhaileh and Tang, 1999)

Figure 3.5 shows the translation process of the English sentence

The old man picks the green lamp up

given a BKB containing four annotated translation examples (Figure 3.5a). The EBMT system selects sub-S-SSTCs that best match the input fragments, based on coverage and number of occurrences in the BKB (see Figure 3.5b). Finally, the selected sub-S-SSTCs are recombined, giving the final translation

Lelaki tua itu kutip lampu hijau itu

as simply the string of the resulting S-SSTC.

3.5 The Need for Sense Disambiguation

While this EBMT system gives translation outputs that are structurally satisfactory, it may output incorrect translations for input words with multiple meanings (c.f. §1.1.2), like the following:

Input: *He fell into the river from the* ***bank****.*

Output: **Dia jatuh ke dalam sungai dari* ***bank****.*

if there are ten occurrences of the translation pair *bank* ↔ *bank* (a financial institution) in the BKB, and only one occurrence of *bank* ↔ *tebing* (the slope beside a body of water, which should have been selected in this case).

It is obvious that WSD capabilities is a highly desirable feature if the EBMT system is to be improved. Many researchers have tried various approaches for WSD and translation

selection (see Chapter 2), including knowledge- and corpus-based methods, as well as hybrids of both.

In the context of EBMT, most literature give more attention to example matching based on structure and fragment coverage length, and semantic similarity is usually handled following Nagao's (1984) suggestion of using a thesaurus.[3] Thesauri group near-synonymous words under concept headings, often organised as a taxonomy-like hierarchy, such as Roget's (2002) Thesaurus. Therefore, given the following English-Japanese translation examples and input text taken from (Nagao, 1984):

Examples:

1. **E:** *A man* ***eats*** *vegetables.* **J:** 人は野菜をたべる。
   (consume food)
2. **E:** *Acid* ***eats*** *metal.* **J:** 酸は金属を侵す。
   (erode)

Input: *He* ***eats*** *potatoes.*

たべる*taberu* from example 1 will be selected correctly to translate *eats* in the input sentence, as the thesaural distances between *man* and *he*, *vegetables* and *potatoes* are smaller than that of between *acid* and *he*, *metal* and *potatoes*. For other examples see (Sato and Nagao, 1990).

In such approaches, the similarity of the substituted items are typically based on their concepts (as well as their contexts) *are*, instead of what their concepts are *related to* (see also §2.2.1(b)). This may cause selection "misses". Consider the following English–Malay translation examples and input:

[3]The use of translation examples and a thesaurus makes this similar to a hybrid WSD approach.

Examples:

1.	**E:** *He drowned in the river.*	**M:** *Dia mati lemas dalam sungai.*
2.	**E:** *We walked along the river* ***bank****.*	**M:** *Kami berjalan di sepanjang* ***tebing*** *sungai.*
3.	**E:** *I deposited my money at the* ***bank****.*	**M:** *Saya menyimpan wang saya di* ***bank****.*

Input: *He drowned near the* ***bank****.*

Even though the correct translation for *bank* (*tebing*) is present in the BKB, the simple word similarity measure described above may not be sufficient for *tebing* to be selected. However, *bank* $\leftrightarrow$ *tebing* and *drown* share concepts related to WATER, a piece of information that can be extracted from dictionary definitions, thesaural concepts and the BKB (translation example 1 above) itself. If an EBMT system has access to this knowledge about *bank* $\leftrightarrow$ *tebing*, then the EBMT system will be more likely to translate such ambiguous sentences correctly. This is the idea of Schütze's (1998) "second-order co-occurrence", and can be encapsulated using vectors, as discussed in §4.1. Lafourcade's (2001) conceptual vector model will be used to represent thematic information about lexical items in this research.

To summarise, in order to incorporate semantic considerations into the EBMT matching phase, the BKB described in this chapter needs to be enriched with semantic information, which may be achieved as follows:

- Sense-tagging S-SSTCs. The S-SSTC can accommodate any extra information that may be attached to each tree node, including sense numbers for words.
- Extracting semantic information for each translation unit (sub-S-SSTC) from the BKB and a lexicon.
- A means to encode the lexical semantic information, namely conceptual vectors.

Each of these requirements will be dealt with in the next two chapters.

3.6 Summary

The S-SSTC is a flexible annotation schema for specifying correspondences between parallel corpora on both lexical and structural levels. It is used to annotate the BKB in an EBMT system that we reviewed, one which performs English–Malay translation. The lack of sense disambiguation features was identified as one weakness of the system. To overcome this weakness, some suggestions were made to enrich the BKB with semantic information. The next two chapters follow up on these suggestions by describing the conceptual vector model (Chapter 4), and presenting in detail the methodology of improving the BKB and translation unit selection (Chapter 5).

CHAPTER 4

CONCEPTUAL VECTOR MODEL AND OPERATIONS

We survey the use of vector representations in some WSD work, and go on to describe the conceptual vector model proposed by Lafourcade (2001), which can be used for lexical semantic analysis in NLP. Conceptual vectors play an important role in our design for the enrichment of a BKB with semantic information. We deal mainly with mathematical operations on conceptual vectors in this chapter, while their construction from dictionary sources is discussed in the next chapter.

4.1 Vector Representations in WSD

In NLP applications, semantic knowledge and information needs to be represented in a form that is machine-tractable. As the basic idea in this research is to exploit the overlapping of multiple concepts related to word senses, inspiration can be drawn from many similar work using mathematical vectors.

WSD researchers adopting machine learning methods often construct vectors containing context features, including n-grams, POS, and semantic features. Additional processing is sometimes carried out to filter out noise and reduce the vector dimensions. Nevertheless, this research is more interested in the use of vectors as exemplified in (Salton *et al.*, 1975; Schütze, 1992; Wilks *et al.*, 1993; Schütze, 1998; Lafourcade, 2001; Pedersen *et al.*, 2005).

Salton *et al.* (1975) used a vector space model for indexing documents in information retrieval. Given t index terms, they constructed a index vector of t dimensions for a document, with each index term regarded as an axis in the t-dimensional vector space. Each document is a point in this vector space, the position of which is represented by its index vector. Two documents can then be compared by computing some similarity coefficient from their index vectors.

Wilks *et al.* (1993) created a weighted vector of gloss words for each dictionary sense entry s by taking the sum of the vectors for related words, of each word appearing in the definition of s. A vector is similarly created for the context of the instance of an ambiguous word. Schütze (1992, 1998) also constructed context vectors, albeit from corpora instead of dictionary definitions, capturing "second-order co-occurrence". As Schütze (1998) described it:

> Instead of forming a context representation from the words that the ambiguous word directly occurs with in a particular context (first-order co-occurrence), we form the context representation from the words that these words in turn co-occur with in the training corpus. Second-order co-occurrence information is less sparse and more robust than first-order information.

Pedersen *et al.* (2005) adapted the ideas from Wilks *et al.* (1993) and Schütze (1998) in constructing vectors for *WordNet* synsets from their glosses and a corpus. The relatedness of a pair of synsets is measured by the cosine of the angle between their vectors. The dimension of Pedersen *et al.*'s (2005) vectors is enormous with 15,000 elements, though they are relatively sparse. As a comparison, Schütze (1998) used singular value decomposition to reduce the vector dimension to 100.

Lafourcade's (2001) conceptual vectors are more similar in nature to Salton *et al.*'s (1975) index vectors, although his vectors describe semantic themes related to any lexical item (including senses, words, phrases, sentences and texts). Class labels from a concept hierarchy are used as the vector base. Lafourcade's aim was to represent themes related to a lexical item in a vector form, without explicitly specifying how they are related. Two lexical items are considered thematically close if the angular distance between their conceptual vectors is less than 45°.

Lafourcade's approach is attractive in that it is a concept-based model using a pre-defined knowledge-based resource, which is more robust than surface word matching methods. A suite of mathematical operations on conceptual vectors and their interpretations in NLP have been investigated in subsequent years, and applied in various NLP tasks (see page 44). Lafourcade and Boitet (2002) also described a WSD method using conceptual vectors that avoids the combinatorial effect of multiple ambiguous words in the same input. The construction of conceptual vectors from a handful of concept labels exploit the hierarchy structure of the concept taxonomy. This produces dense vectors, i.e. all vector elements have non-zero values, which helps to overcome data-sparseness.

Based on these considerations, Lafourcade's conceptual vector model is chosen for representing semantic information in this research. However, as Lafourcade did not explain in detail how the initial set of concept labels are assigned to word senses — apart from their thesaural category — some systematic guidelines will have to be drawn up to do this (see Chapter 5).

4.2 Conceptual Vectors and Thematic Projection

Lafourcade (2001) proposed conceptual vectors (CV) as a "thematic representation of textual segments". The *thesaurus hypothesis* (see Schwab and Lafourcade, 2003) considers a set of concepts as a language generator. In the CV model, lexical meaning is seen to be a projection of semantic fields in a vectorial space, the base of which is a set of pre-defined concepts $\mathcal{C} = \{c_1, c_2, \ldots c_N\}$. In other words, the meaning of a lexical item (including word, phrase, sentence, text) is annotated with concepts — and the "intensity" of each concept — as its themes.

Given $\mathcal{C}$, a set of D concepts, a CV $V(w)$ for a lexical item w is a D-uple, i.e. "a linear combination of elements c_i of $\mathcal{C}$" (Lafourcade, 2002):

$$V(w) = (v_1, v_2, \ldots, v_D) \qquad D = |\mathcal{C}| \tag{4.1}$$

Each vector component v_i corresponds to a concept c_i, and the numerical value of v_i is regarded as the "intensity" of c_i for w. For example, if we specified "bank" (as a financial institution) to be related to the themes (amongst others) CORPORATION, WORKPLACE, PUBLIC_ECONOMY, MONEY and LENDING, its CV might be

$$(\ldots, \text{CORPORATION}_{0.1673}, \text{WORKPLACE}_{0.2747}, \text{PUBLIC\_ECONOMY}_{0.1903}, \text{MONEY}_{0.1944}, \text{LENDING}_{0.1513}, \ldots).$$

All conceptual vectors are normalised so that they are unit vectors.

This vector model makes it possible to apply mathematical operations on the CVs, and much of Lafourcade's research was on understanding the linguistic (and semantic)

interpretations of such manipulations, as well as their implications from a computational linguistics point of view. The CV model has been used in NLP tasks requiring semantic analysis, including the following:

- text indexing and information retrieval (Lafourcade and Boitet, 2002),
- sense disambiguation and lexical transfer (Lafourcade, 2001; Lafourcade and Boitet, 2002),
- aligning sense entries from different dictionaries (Lafourcade, 2002),
- discovering semantic relations between words, e.g. synonymy, antonymy, hypernymy[1] and meronymy[2] (Lafourcade, 2003; Schwab and Lafourcade, 2003).

In the following sections, we present a selection of measures and operations involving CVs, as well as the vector propagation algorithm used for the semantic analysis of texts.

4.3 Thematic Promixity

The thematic distance between two lexical items x and y is defined as the cosine similarity (CSim) and angular distance ($\mathcal{D}_{\mathcal{A}}$) between their conceptual vectors, $V(x)$ and $V(y)$:

$$\text{CSim}(x, y) = \frac{V(x) \cdot V(y)}{|V(x)| \times |V(y)|} \tag{4.2}$$

$$\mathcal{D}_{\mathcal{A}}(x, y) = \arccos(\text{CSim}(x, y)) \tag{4.3}$$

Two lexical items are considered to be *thematically close* if the angular distance between the CVs representing them is less than or equal to $\frac{\pi}{4}$ or 45° ($\mathcal{D}_{\mathcal{A}}(x, y) \leq \frac{\pi}{4}$). By definition, $\mathcal{D}_{\mathcal{A}}(\overrightarrow{\mathbf{0}}, \overrightarrow{\mathbf{0}}) = 0°$ and $\mathcal{D}_{\mathcal{A}}(X, \overrightarrow{\mathbf{0}}) = 90°$ where $\overrightarrow{\mathbf{0}}$ is the null vector representing the "empty idea". Here are some examples of $\mathcal{D}_{\mathcal{A}}$ values between the words *river, lake, shop, canoe*

[1] the `is-a` relation, e.g. *tiger* `is-a` *cat* `is-a` *animal*.
[2] the `part-of` or `member-of` relation, e.g. *room* `part-of` *building*, *soldier* `member-of` *squad*.

and *paddle*:

$$\mathcal{D}_{\mathcal{A}}(\textit{river, river}) = 0^\circ \qquad \mathcal{D}_{\mathcal{A}}(\textit{river, lake}) = 34.39^\circ$$
$$\mathcal{D}_{\mathcal{A}}(\textit{river, shop}) = 80.95^\circ \qquad \mathcal{D}_{\mathcal{A}}(\textit{canoe, paddle}) = 20.89^\circ$$

Lafourcade (2005) used the "starry sky" metaphor to explain the angular distance as a thematic proximity measure. Consider the space of all word senses as a sky full of stars. From the point of view (the origin point) of an observer on the Earth, the actual Euclidean distance between two stars (word senses) cannot be known: instead, what is referred to as "proximity" between the stars is the angular distance between them, as observed by the earth-bound star-gazer. $\mathcal{D}_{\mathcal{A}}$, therefore, cannot be used to judge if one lexical item (e.g. *crimson*) is a more "intense" version of another, e.g. *red*. It can only indicate that they are highly similar in their themes, i.e. $\mathcal{D}_{\mathcal{A}}(\textit{crimson, red})$ may be equal or very close to 0° (almost collinear).

4.4 Conceptual Vector Operations

This section describes a selection of useful operations on CVs, which will be employed in our design and implementation later. See Prince and Lafourcade (2003) or Lafourcade (2005) for a more comprehensive discussion.

4.4.1 Magnitude

The magnitude of a CV V is defined as

$$\begin{aligned} |V| &= \sqrt{{v_1}^2 + \cdots + {v_n}^2} \\ &= \sum_{i=1}^{n} {v_i}^2. \end{aligned} \tag{4.4}$$

4.4.2 Normalisation

A normalised vector $\widehat{V} = (\widehat{v_1}, \ldots, \widehat{v_n})$ of $V = (v_1, \ldots, v_n)$ is a unit vector (i.e. vector with magnitude 1).

$$\widehat{V} = N(V) \qquad | \qquad \widehat{v_i} = \frac{v_i}{|V|} \tag{4.5}$$

4.4.3 Sum

Given vectors X and Y, their classical vector sum is:

$$V = X + Y \qquad | \qquad v_i = x_i + y_i. \tag{4.6}$$

To generalise:

$$V = \sum_{k=1}^{n} X_k \qquad | \qquad v_i = \sum_{k=1}^{n} x_{ki} \tag{4.7}$$

4.4.4 Normalised Sum

Given vectors X and Y, their mean or normalised sum is

$$X \oplus Y = \frac{X + Y}{|X + Y|}. \tag{4.8}$$

To generalise:

$$\bigoplus_{i=1}^{n} V_i = \frac{\sum_{i=1}^{n} V_i}{|\sum_{i=1}^{n} V_i|} \tag{4.9}$$

With $\overrightarrow{\mathbf{0}}$ (the "empty idea") being the neutral element, we also have the following

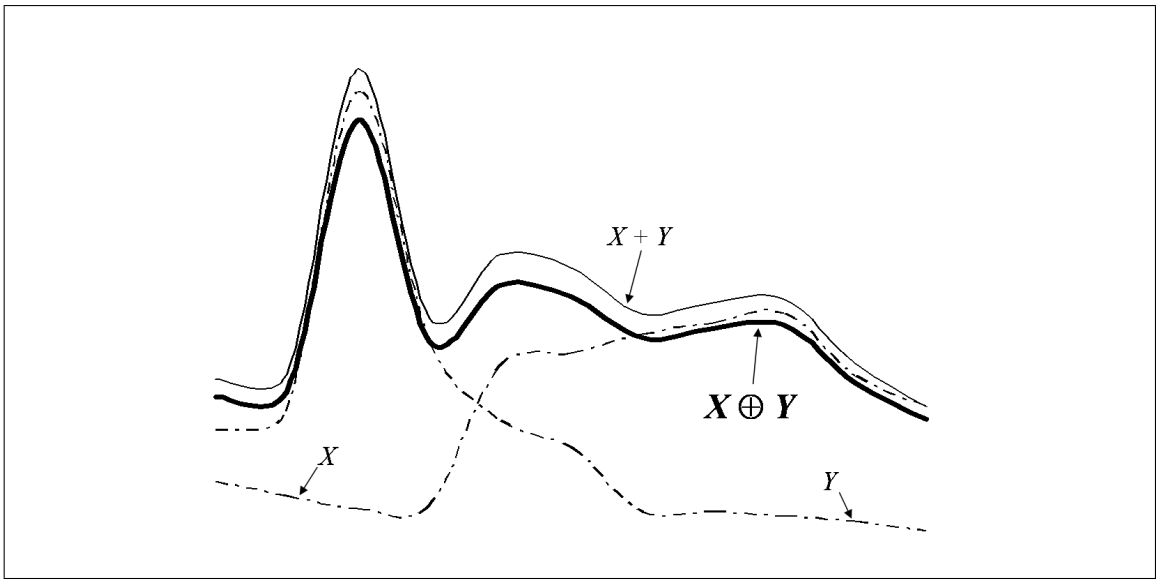


Figure 4.1: Normalised Sum of Two Conceptual Vectors

properties:

$$\overrightarrow{\mathbf{0}} \oplus \overrightarrow{\mathbf{0}} = \overrightarrow{\mathbf{0}} \qquad X \oplus X = X$$

$$X \oplus \overrightarrow{\mathbf{0}} = \overrightarrow{\mathbf{0}} \oplus X = X$$

The normalised sum can be interpreted as the *mean* of the operand vectors, and is often used to create conceptual vectors for an ambiguous word from its different senses.

4.4.5 "Normed" Term-to-Term Product

Given vectors X and Y, their "normed" term-to-term product is

$$V = X \otimes Y \qquad | \qquad v_i = \sqrt{x_i \cdot y_i} \tag{4.10}$$

This operator has the following properties:

$$X \otimes X = X \qquad X \otimes \overrightarrow{\mathbf{0}} = \overrightarrow{\mathbf{0}} \otimes X = \overrightarrow{\mathbf{0}}$$

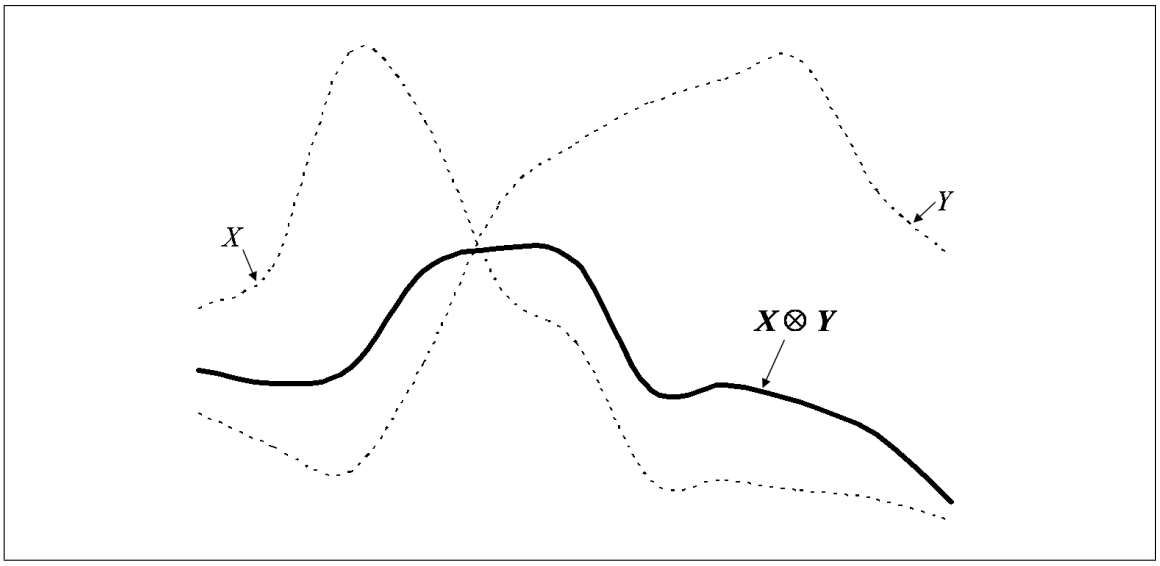


Figure 4.2: "Normed" Term-to-Term Product of Two Conceptual Vectors

The "normed" term-to-term product can be considered an "intersection" function, as it accentuates concepts common to both operand vectors. Contrary to other operations, the result here is not usually normalised to the unit vector, as its magnitude is a good indicator of how highly the two operand vectors correlate.

4.4.6 Weak Contextualisation

Placing a vector Y in the context of X, i.e. contextualisation (γ function) is defined as

$$\gamma(X, Y) = X \oplus (X \otimes Y) \tag{4.11}$$

The γ functions is not symmetrical, and has the following properties:

$$X \neq Y \Rightarrow \gamma(X, Y) \neq \gamma(Y, X) \qquad \gamma(X, X) = X$$

$$\gamma(X, \overrightarrow{\mathbf{0}}) = X \oplus \overrightarrow{\mathbf{0}} = X \qquad \gamma(\overrightarrow{\mathbf{0}}, X) = \overrightarrow{\mathbf{0}} \oplus \overrightarrow{\mathbf{0}} = \overrightarrow{\mathbf{0}}$$

X, the context, acts as a "filter" to pick out dominant concepts in Y that "resonate"

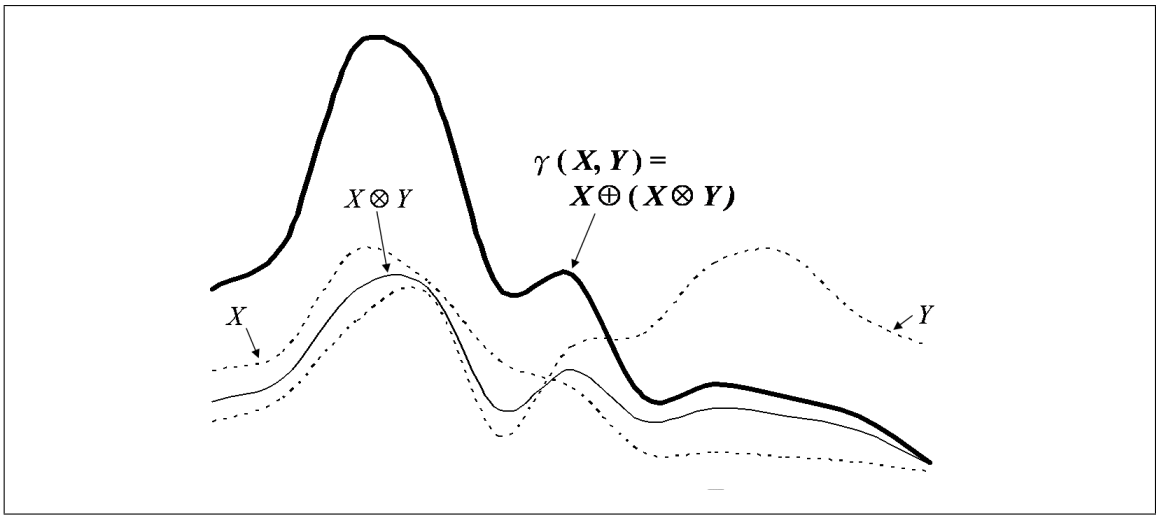


Figure 4.3: Contextualisation of One Conceptual Vector by Another

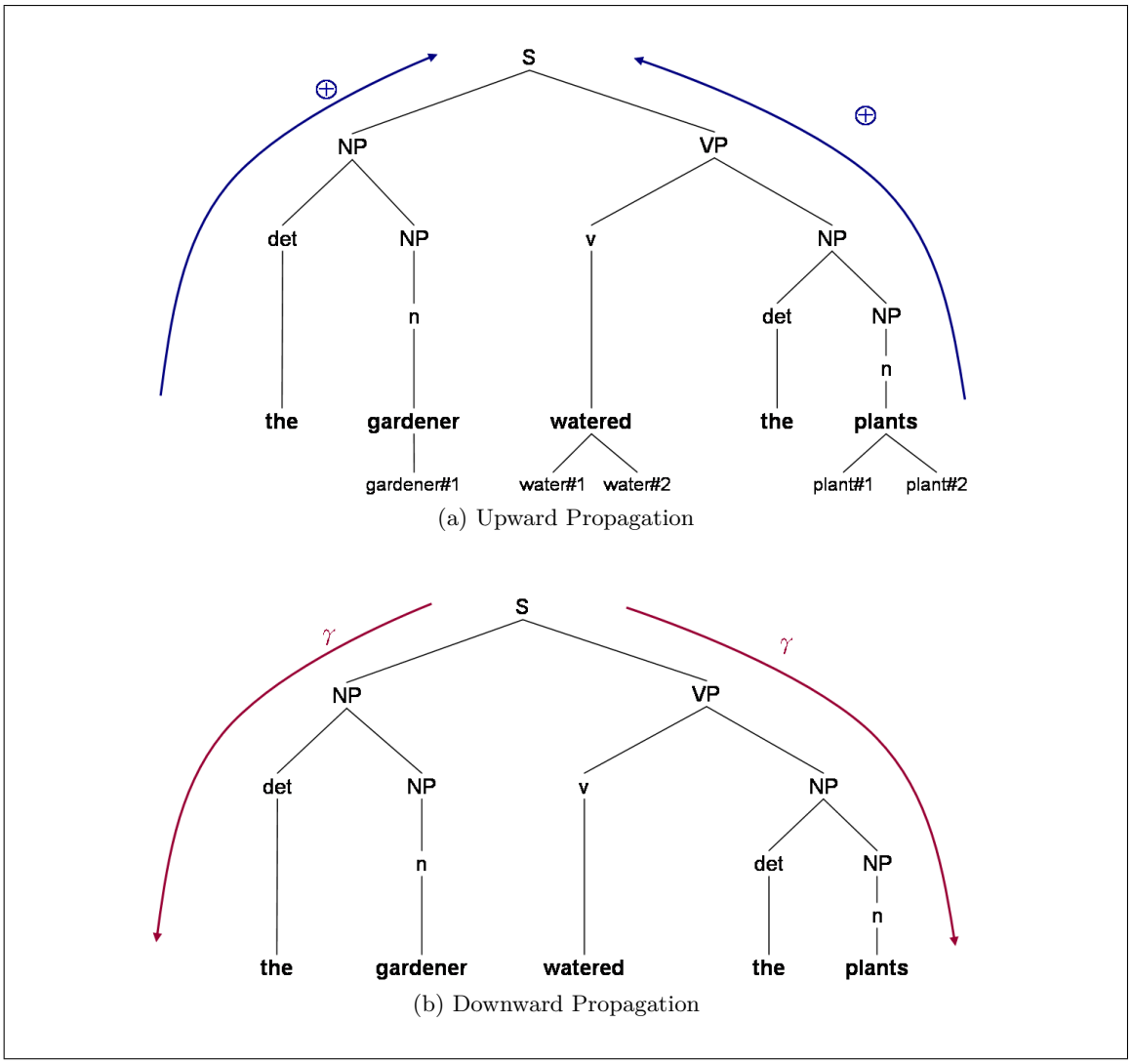
with its (X's) concepts, propotionally to their intersection. As an example, the conceptual vector representing the ambiguous word *bank* comprises concepts related to FINANCE and GEOGRAPHICAL FEATURE. Contextualising it with *river*, i.e. $\gamma(bank, river)$, will dim the concepts related to FINANCE but highlight GEOGRAPHICAL FEATURE.

4.5 Semantic Analysis with Conceptual Vectors

It is possible to perform a semantic analysis on a text using CVs, given the syntactic tree resulting from a syntactic analysis. The following *vector propagation algorithm* can be applied to phrase-structure trees (Lafourcade, 2004) and UNL[3] graphs (Lafourcade and Boitet, 2002), where surface lexical items are located at the leaf nodes of the tree. The algorithm first computes the overall context of the text, and then "activates" correlating concepts in each lexical item using the overall context.

1. **Initialisation.** For each leaf node p containing content word w, tag p with concep-

[3]Universal Networking Language. See (Sérasset and Boitet, 2000) for more details.



(a) Upward Propagation

(b) Downward Propagation

Figure 4.4: The Vector Propagation Algorithm on a Phrase-Structure Tree

tual vectors for all possible senses of $w\{s_1, \ldots, s_n\}$, such that

$$V(p) = \bigoplus_{i=1}^{n} V(s_i). \tag{4.12}$$

2. **Upward Propagation.** Starting from the nodes nearest the leaves and working towards the root, the vector for each internal node p is computed as the normalised sum of its children's vectors (Figure 4.4a):

$$\uparrow V(p) = \bigoplus_{q \in p\text{'s children}} V(q) \tag{4.13}$$

3. **Downward (Back) Propagation.** Starting from the root and working towards the leaves, propagate the vector of each internal node downwards, weakly contextualising those of the children (Figure 4.4b):

$$\downarrow V'(p) = \gamma(V(p), V(parent(p))) \tag{4.14}$$

The vector propagation algorithm has the effect of making the (polysemous) vectors in the text "resonate" with one another, thereby activating word senses with common themes or semantic meanings. This is easily extended to be a WSD approach, where the sense s_i that minimises $\mathcal{D}_{\mathcal{A}}(V'(p), V(s_i))$ at each leaf node p containing polysemous word w is selected.

A WSD solution using the vector propagation algorithm handles multiple occurrences of ambiguous words in an attractive manner. Given an input text containing m ambiguous words, each with n possible senses, a "brute force" approach would need to consider all n^m possible sense combinations to identify the optimum sense "configuration" to maximise some disambiguation score. The vector propagation algorithm, in contrast, needs $O(m)$ time to propagate the conceptual vectors up and down the tree, and $O(mn)$ time to select senses for all ambiguous words.

4.6 Summary

Conceptual vectors represent the semantic concepts and themes related to the meanings of lexical items as mathematical vectors. A number of useful mathematical operations on these vectors have been defined, together with brief descriptions of their implications in NLP tasks, including an application of a vector propagation algorithm in WSD.

In the next chapter, we will discuss how this conceptual vector model can be employed to enrich an S-SSTC-annotated BKB (as described in Chapter 3) with semantic information, and how the enriched BKB supports translation selection in the reviewed EBMT system.

CHAPTER 5

ENRICHING THE BKB WITH CONCEPTUAL VECTORS TO IMPROVE SUB-S-SSTC SELECTION DURING EBMT

At the end of Chapter 3, we concluded that translation selection in the discussed EBMT system can be improved by incorporating sense disambiguation. This requires the BKB used by the EBMT system to be enriched with semantic information. In this chapter, we will outline how this can be carried out using the conceptual vectors model described in Chapter 4. We also describe the algorithm for selecting appropriate sub-S-SSTCs for ambiguous input text segments, using the enriched BKB.

5.1 Design Considerations

To recap briefly from Chapter 3, the reviewed EBMT system relies on frequency count when selecting translation units (i.e. sub-S-SSTCs in the S-SSTC framework) from the BKB. This often gives unsatisfactory outputs, since no consideration is given to sense disambiguation.

To illustrate, consider a BKB in which the translation pair *bank* $\leftrightarrow$ *bank* (a financial institution) occurs 10 times, and *bank* $\leftrightarrow$ *tebing* (the slope beside a body of water) just once. Given the input sentence *he fell into the river from the* ***bank***, the current EBMT system will always select the translation pair *bank* $\leftrightarrow$ *bank* erroneously.

We seek to improve the EBMT system by incorporating semantic disambiguation, but

decided against approaching the problem as one of assigning sense numbers to ambiguous words in the input sentence, *a lá SENSEVAL* WSD exercises (SENSEVAL, 2005). This is because the relationship between *source word* → *sense* → *target translation* is not a simple one-one mapping: even if an ambiguous word has been sense-tagged correctly, it may still be unclear which target language word should be used in the final translated text. For example, the *WordNet* sense

> **circulation#6 (n)** the spread or transmission of something (as news or money) to a wider group or area

would have the Malay translation *penyebaran* for describing the spread of news or gossip, and *peredaran* for the transmission of money.

On the other hand, there is actually no selection to be done on the part of the MT system, if two different senses of an SL word are translated to the same TL word, i.e. when the ambiguity is carried over to the TL. For example, *channel* can be translated to *saluran* in Malay for both *television channel* and *communication channel*. In other words, MT systems do not need to completely understand text as in an artificial intelligence sense (Hutchins, 1986, §19.3).

It would therefore be more beneficial to disambiguate between *translations*, instead of sense numbers as proposed in (Al-Adhaileh and Tang, 1999), when working in the context of an MT application. Some researchers (Ide, 1999; Ide *et al.*, 2002; Ng *et al.*, 2003; Cabezas and Resnik, 2005) regarded this as WSD using translations as sense-tags. This approach constrasts sharply with Lee and Kim's (2002) translation selection based on 'word-to-sense and sense-to-word', as the sense selection stage is bypassed entirely during the MT run-time. With this in mind, our research objective is, given an input text, to

select the most appropriate sub-S-SSTCs from the BKB that match the input fragments, for building the translation.

5.2 Design Overview

We use a criminal investigation analogy to illustrate the task at hand. At a crime scene, where a crime has been committed, the investigators need to pin it to one of several suspects. The circumstances and forensic evidences found at the scene often provide hints and clues to the criminal's *modus operandi*, behaviours and habits, aiding investigators to profile the criminal after analysing these circumstances and evidences (Meyer, 2000; Turvey, 2001).

Now consider an input sentence containing words with multiple possible translations as a "crime scene", the "crime" being translation ambiguity. There exists multiple sub-S-SSTCs — the "suspects" — which can be used to construct the translation output. To solve the "crime" i.e. translate the sentence, our mission is to identify the most probable "suspect", by gathering "clues" from the "crime scene" (the input sentence) and matching them against the "profiles" from a "database of suspects" (the BKB).

This means that the selection of sub-S-SSTCs will be based on the semantic similarity of the input sentence and the examples in the BKB, which we measure using the conceptual vector model described in Chapter 4. The BKB will first be enriched with semantic information drawn from a lexicon, which in turn will have its sense entries tagged with concept labels. The reason for tagging entries in a lexicon with concepts is to increase coverage, as opposed to methods using word features or overlapping of words in definition texts (e.g. Lesk, 1986; Lim *et al.*, 2002; Lim, 2003).

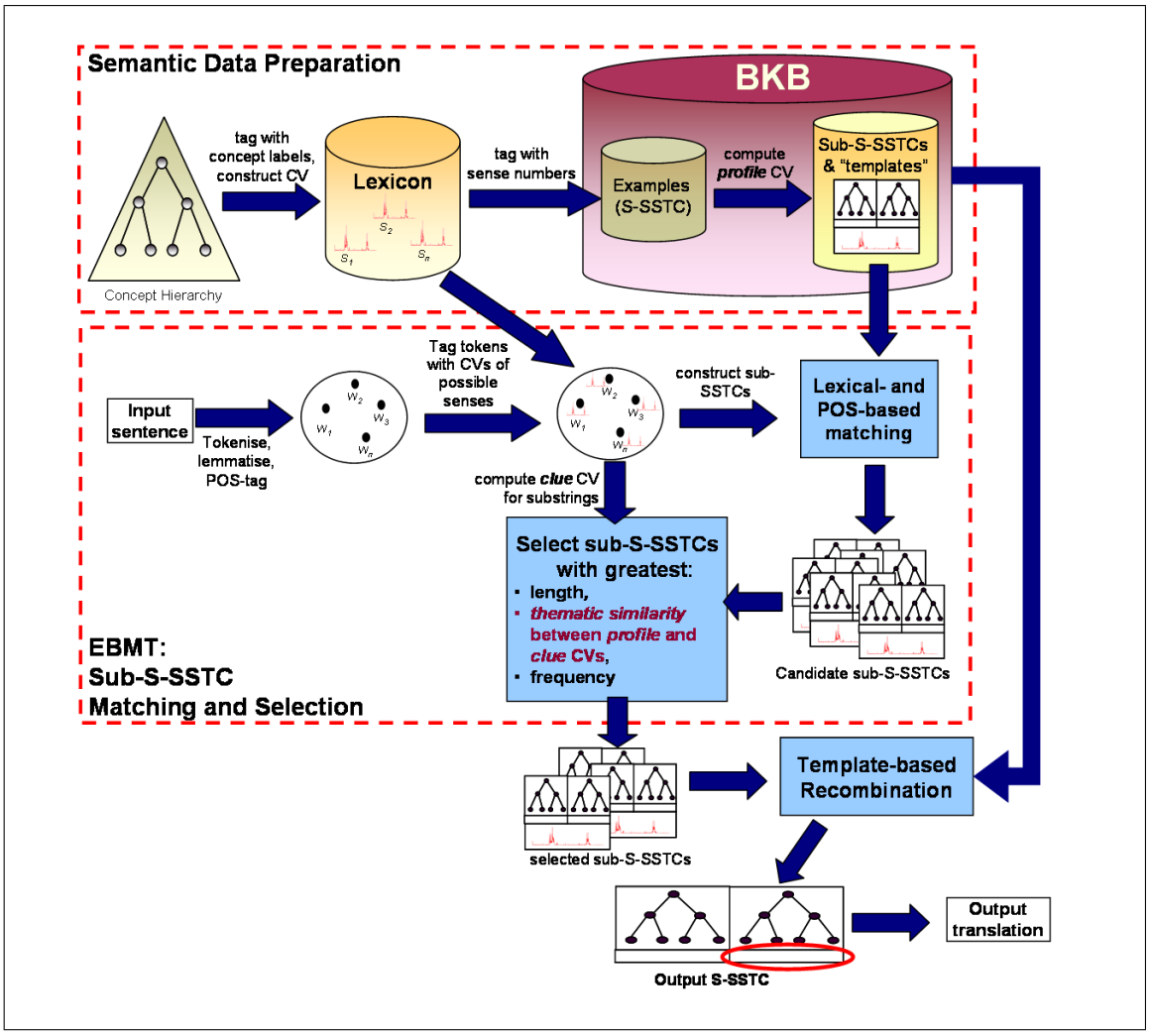


Figure 5.1: Design Overview: Improving Sub-S-SSTC Selection in EBMT with Semantic Similarity Measure

Our methodology is divided into two major stages, as illustrated in Figure 5.1 and described in the following sections:

- **semantic data preparation**, where the BKB is tagged with semantic information;
- **sub-S-SSTC selection during EBMT**, where the sub-S-SSTC selection phase in an EBMT system is enhanced with the semantic information from the BKB.

5.3 Semantic Data Preparation

To facilitate semantic analysis, the BKB needs to be enriched with semantic information – in this case, conceptual vectors (CVs). The idea is to conceptually *profile* each sub-S-

SSTC, based on lexicon definitions, as well as its usage as found in the BKB. This makes the method used here a hybrid of knowledge-based and corpus-based approaches.

The first aim is to construct CVs for word senses as found in a dictionary or lexicon. The automated CV construction method described in (Lafourcade *et al.*, 2004) was suitable for building CVs for head words of a *target language*, if CVs for a *source language* already exist. Since we do not have access to such resources, we start by manually tagging lexicon entries with concept labels instead.

The preparation of the semantic data involves the steps below, each of which will be described in the ensuing sections:

1. Tagging a lexicon $\mathcal{L}$ with concepts,
2. Constructing conceptual vectors (CV) for the sense entries in $\mathcal{L}$,
3. Sense-tagging examples in the BKB with respect to $\mathcal{L}$,
4. Computing "profile" CVs for sub-S-SSTCs in the BKB.

These steps have some similarities with the machine-tractable dictionary construction process described in (Wilks *et al.*, 1993; Lim *et al.*, 2002; Lim, 2003). However, instead of extracting semantic primitives from the lexicon itself, a set of fixed, pre-defined concept labels forming a semantic hierarchy, is used. This preference was made because

(a) concepts labels provide wider coverage than words, which is what Lim *et al.*'s (2002) and Lim's (2003) semantic primitives really are; and

(b) Lafourcade's (2001) conceptual vectors, constructed based on a concept hierarchy, are dense: all vector elements have non-zero values. Dense vectors provide higher coverage than sparse vectors.

In addition, the aim of (Wilks *et al.*, 1993; Lim *et al.*, 2002) was to make a dictionary more machine-tractable by essentially reducing definitions to a smaller set of words, whereas the steps above seek to gather thematic concepts that are related to parallel text translations across languages.

5.3.1 Tagging Lexicon with Concepts

We start the data preparation phase with the following resources:

- **A lexicon $\mathcal{L}$,** containing head words of one of the languages in the BKB. We assume this to be English for discussion purposes. Let the set of word senses listed in it be $\{s_1, s_2, \ldots, s_M\}$. Each sense entry s specifies a head word w, a sense number n (i.e. the nth sense of word w), a part-of-speech *pos*, and definition text *def*. We denote the set of senses of a lexical item w listed in $\mathcal{L}$ as $\mathcal{L}(w)$.
- **A concept hierarchy $\mathcal{C}$,** containing D concepts $\{c_1, c_2, \ldots, c_D\}$. This could be the concept hierarchy as defined by a thesaurus or an ontology.

Each sense entry s in $\mathcal{L}$ is tagged with a selection of concepts, $\mathcal{C}_s \subset \mathcal{C}$ that reflect the meaning and "themes" of s. Note that this involves more than selecting just one semantic class for each sense, as was done in (Bond *et al.*, 2004): to fully exploit the conceptual vector model, each sense entry should be tagged with as many relevant classes as possible.

Both Lafourcade and Boitet (2002) and Schwab and Lafourcade (2003) automatically tagged lexicon sense entries with concept labels by bootstrapping from a manually-prepared data set, but did not elaborate if any principles were used to guide human annotators during the manual tagging stage, apart from the use of a concept immediately related to a headword (the "genus") in a thesaurus (Lafourcade, 2001), domain codes

(Lafourcade *et al.*, 2004), and the discretion of the annotators (Lafourcade, 2001). We complement the notion of "primary concept" (that of the "genus" term of a headword) with "secondary concepts", which will help human annotators to determine the set of concept labels to be assigned to a sense entry, based on its definition text or glosses.

We can therefore determine $\mathcal{C}_s$ based on the definition text of s and the following guidelines:

- **primary concepts:** These are the concepts under which s would be classified in the concept taxonomy or hierarchy.

  Sense entries for verbs are assigned the primary concept as the noun sense entry having the same (or morphologically-related) head word and the same "core" senses (as in "the act of doing ‹verb›"). Adjective and adverb senses, on the other hand, will take on primary concepts of their morphologically-related counterparts. Therefore, both the verb and the noun *walk* would have the same primary concept WALK, while both adjective *angry* and noun *anger* would be assigned ANGER. However, the noun and verb *plant* would have different primary concepts: the former would be assigned PLANT, the latter CULTIVATION.

- **secondary concepts:** These are concepts that are thematically related to s, i.e. concepts of the word senses *appearing in the definition text of* s.

  This "second-level look up" method is inspired by Wilks *et al.* (1993), Lim *et al.* (2002) and Lim (2003), where "semantic primitives" and gloss words for a sense entry are extracted from the definition text of the head word, as well as those of the *words* in the definition text.

To see how both these guidelines are applied together, consider this sense entry:

bank#1 (n) a financial institution that accepts deposits and channels the money into lending activities.

The primary concepts of **bank#1** would be CORPORATION and WORKPLACE, while the secondary concepts might be PUBLIC_ECONOMY (for *financial*), MONEY (*deposit* and *money*), and LENDING (*lending*). Hence, **bank#1** would be tagged with the set of concepts

$$\mathcal{C}_{bank\#1} = \{\text{CORPORATION, WORKPLACE, PUBLIC\_ECONOMY, MONEY, LENDING}\}. \quad (5.1)$$

5.3.2 Constructing CVs for Lexicon Sense Entries

The next step in the data preparation phase is to construct CVs from $\mathcal{C}_s$ for each sense entry s.

Let the number of concepts in our concept hierarchy $\mathcal{C} = \{c_1, \ldots, c_D\}$ be D. Each CV $(v_1, \ldots, v_D)$ would then have D elements, where the value of each v_i denotes the "strength" of concept c_i.

If sense entry s is tagged with concepts $\mathcal{C}_s = \{c_p, \ldots, c_q\} \subset \mathcal{C}$, then the CV representing s, $V(s) = (v_1, \ldots, v_D)$ can be computed by the following steps:

Step 1. Initialise $V(s) = V^0$ as a boolean vector, where

$$v_i^0 = \begin{cases} 1 & \text{if } c_i \in \mathcal{C}_s \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

Lafourcade (2001) calls this a *raw vector*. Referring back to the sense entry in (5.1), if we plotted $V^0(bank\#1)$ or $v_i^0(bank\#1)$ against c_i, it may be visualised as in Figure 5.2.

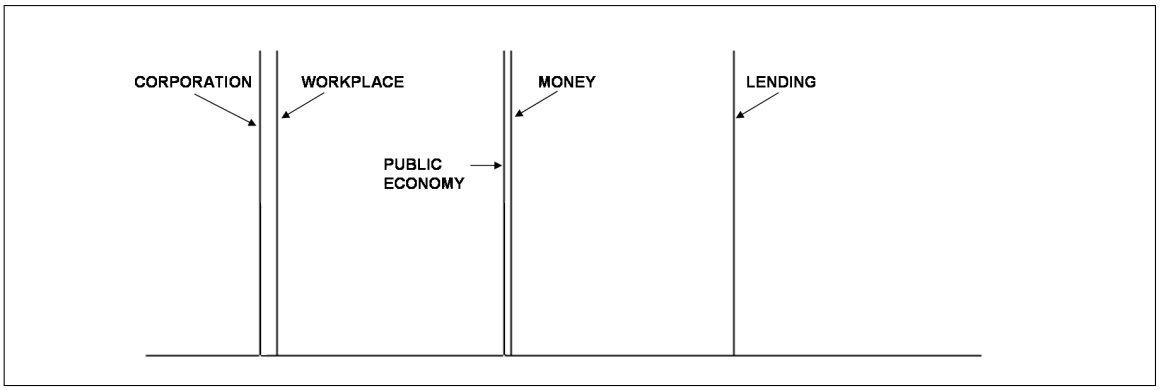


Figure 5.2: The raw vector $V^0(bank\#1)$ showing concepts related to *bank*#1

Step 2. Iteratively compute V^j, that is, the jth iteration of V. In each iteration, as the concepts send out "ripples" along the tree paths in $\mathcal{C}$, each v_i^{j+1} receives "strength contributions" from other elements in V_j, based on the distance between c_i and other concepts in $\mathcal{C}$:

$$v_i^{j+1} = v_i^j + \sum_{r=1}^{D} \frac{v_r^j}{2^{\text{dist}(c_i,c_r)}} \tag{5.3}$$

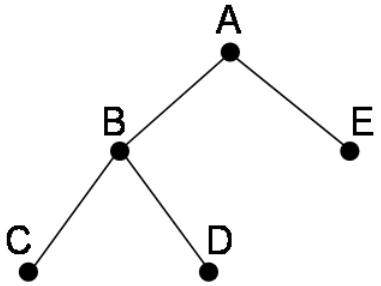
where $\text{dist}(c_i, c_r)$ is the length of the shortest path between concepts c_i and c_r in the concept hierarchy $\mathcal{C}$. Figure 5.3 demonstrates how (5.3) is applied on a raw vector V^0 to produce V^1, based on a small $\mathcal{C}$ with five concepts. Figures 5.4(a) and 5.4(b) shows $V^1(bank\#1)$ and $V^2(bank\#1)$ respectively, iteratively computed from $V^0(bank\#1)$ in Figure 5.2.

Such an iterative process of computing vectors from a set of concepts is termed V^j *augmentation* in (Lafourcade, 2001). Each V^j with $j > 0$ is an *augmented vector* resulting from this process.

Step 3. Stop at some $j = T$ and set $V(s) = \widehat{V^T}$.

5.3.3 Sense-Tagging Examples in the BKB

The next step in the data preparation stage is to sense-tag the English part of the BKB examples, based on the sense entries in $\mathcal{L}$. This can either be done manually, or semi-



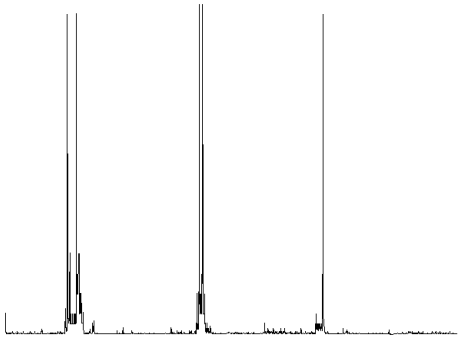
dist	**A**	**B**	**C**	**D**	**E**
A	0	1	2	2	1
B	1	0	1	1	2
C	2	1	0	2	3
D	2	1	2	0	3
E	1	2	3	3	0

(a) Small Concept Hierarchy with Five Concepts

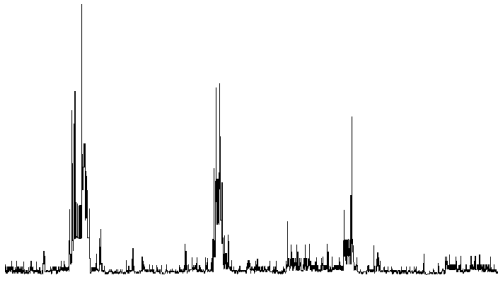
			v_A	v_B	v_C	v_D	v_E
$V^0 = ($			0 ,	0 ,	1 ,	0 ,	1)
	$\Vert$						
	$\frown$						
v_A	0	$\times$	$\frac{1}{2^0}= 0$	$\frac{1}{2^1}= 0$	$\frac{1}{2^2}= 0$	$\frac{1}{2^2}= 0$	$\frac{1}{2^1}= 0$
v_B	0	$\times$	$\frac{1}{2^1}= 0$	$\frac{1}{2^0}= 0$	$\frac{1}{2^1}= 0$	$\frac{1}{2^1}= 0$	$\frac{1}{2^2}= 0$
v_C	1	$\times$	$\frac{1}{2^2}= 0.25$	$\frac{1}{2^1}= 0.5$	$\frac{1}{2^0}= 1$	$\frac{1}{2^2}= 0.25$	$\frac{1}{2^3}= 0.125$
v_D	0	$\times$	$\frac{1}{2^2}= 0$	$\frac{1}{2^1}= 0$	$\frac{1}{2^2}= 0$	$\frac{1}{2^0}= 0$	$\frac{1}{2^3}= 0$
v_E	1	$\times$	$\frac{1}{2^1}= 0.5$	$\frac{1}{2^2}= 0.25$	$\frac{1}{2^3}= 0.125$	$\frac{1}{2^3}= 0.125$	$\frac{1}{2^0}= 1$
	$\smile$						
$V^1 = ($			0.75 ,	0.75 ,	2.125 ,	0.375 ,	2.125)

(b) Computing V^1 from V^0

Figure 5.3: Applying (5.3) on a Raw Vector with 5 Concepts



(a) $V^1(bank\#1)$



(b) $V^2(bank\#1)$

Figure 5.4: jth iteration in computation of $V(bank\#1)$

automatically by applying the CV propagation algorithm (Lafourcade, 2001, also described in §4.5) on the SSTC English text, followed by manual checking.

The vector propagation algorithm was modified to work on dependency trees, which are used as the SSTC trees stored in the BKB, as opposed to phrase-structure trees in

(Lafourcade, 2001) and UNL graphs in (Lafourcade and Boitet, 2002). The modification is necessary because lexical items occur only as leaves in both phrase-structure trees and UNL graphs, while all nodes (both terminals and non-terminals) in dependency trees contain lexical items. We describe the modified algorithm below, and illustrate it graphically in Figure 5.5.

Given an SSTC $\mathcal{S} = (st, tr, co)$ where st is a string in one language (English in this case), tr its associated tree, and co the correspondence between st and tr:

Step 1. Initialise $V(p) = V_{\mathcal{L}}(w_p)$ for each node p in tr and w_p the substring corresponding to p:

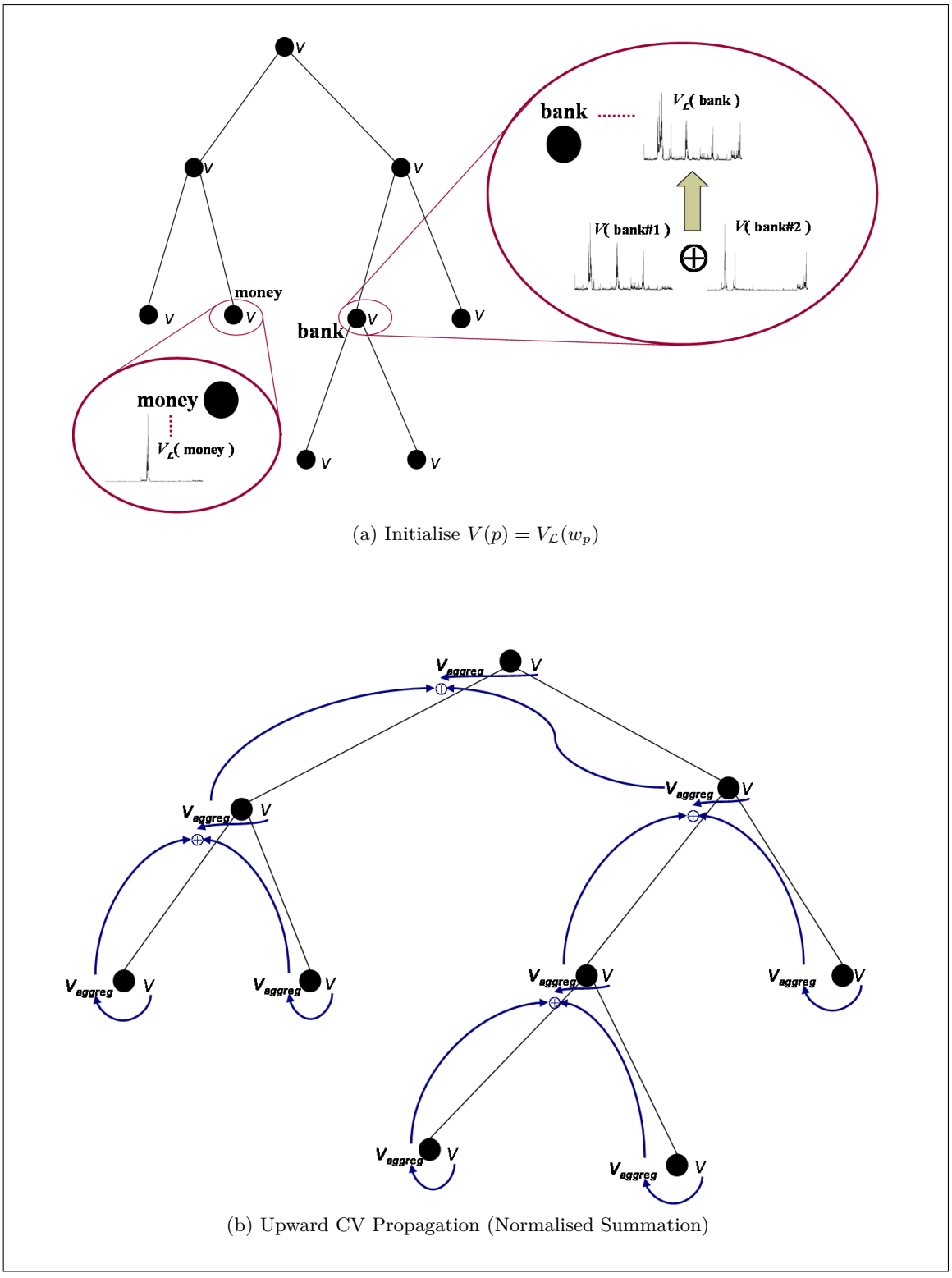
$$\begin{aligned} V(p) &= V_{\mathcal{L}}(w_p) \\ &= \bigoplus_{s \in \mathcal{L}(w_p)} V(s) \end{aligned} \tag{5.4}$$

$V(s)$ is the CV of sense entry s as defined in the previous section.

The SSTC shown in Figure 5.5 has the string representation *"... money ... bank ..."*. Here we assume that *money* has only one sense in $\mathcal{L}$, while *bank* has the following senses:

> **bank#1 (n)** a financial institution that accepts deposits and channels the money into lending activities.
>
> **bank#2 (n)** sloping land (especially the slope beside a body of water).

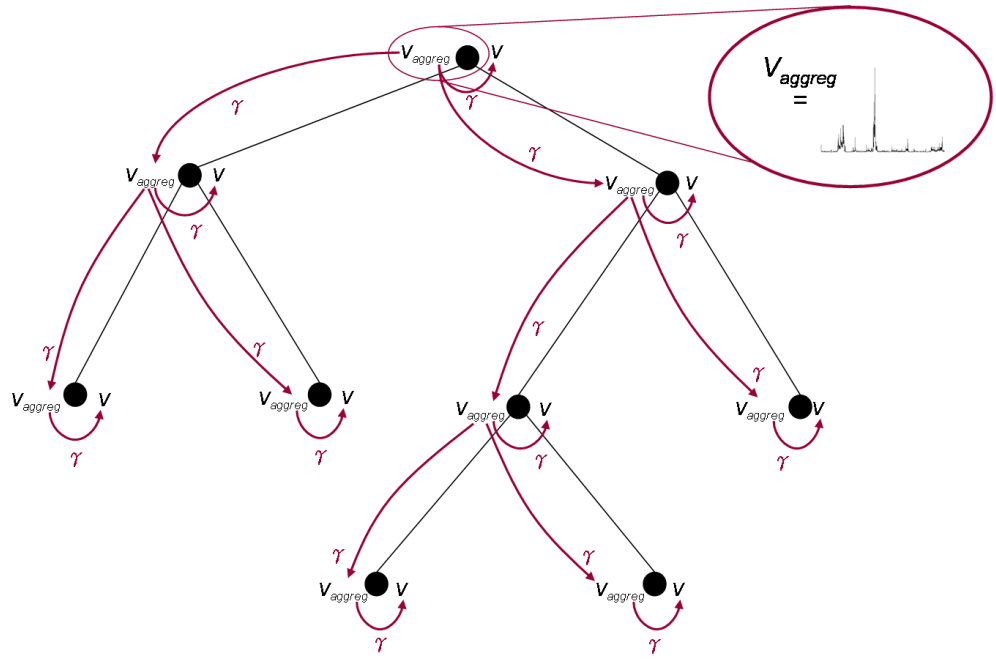
The node corresponding to *bank* therefore is tagged with the normalised sum of V(bank#1) and V(bank#2), while the node for *money* is tagged with V(money) (Figure 5.5a).



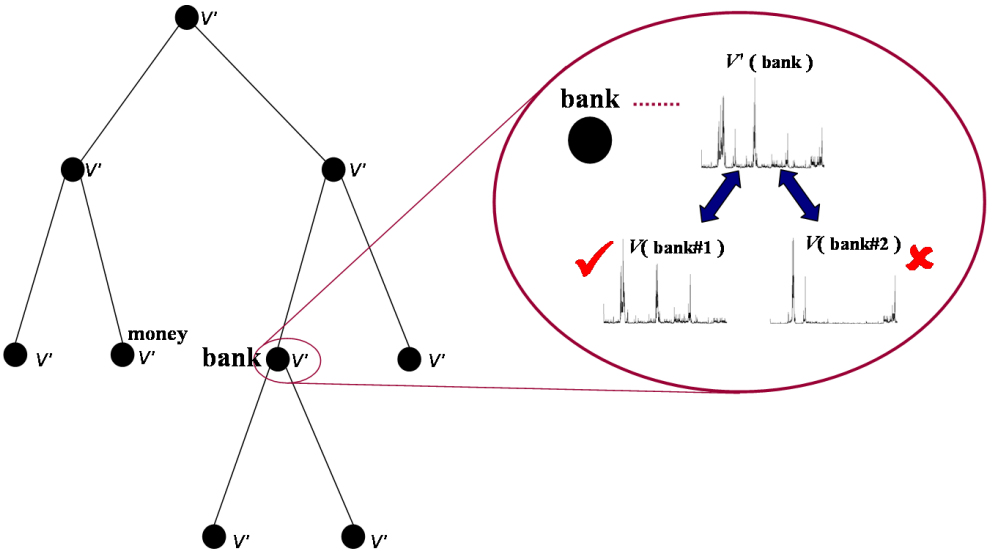
(a) Initialise $V(p) = V_{\mathcal{L}}(w_p)$

(b) Upward CV Propagation (Normalised Summation)

Figure 5.5: Modified Vector Propagation Algorithm for SSTC with dependency trees



(c) Downward CV Propagation (Contextualisation)



(d) Sense Selection Based on $\mathrm{CSim}(V(s), V'(p))$

Figure 5.5: Modified Vector Propagation Algorithm for SSTC with dependency trees (cont.)

Step 2. Propagate CVs upwards, starting from the terminal nodes towards the root (Figure 5.5b).

- For each terminal node p, set $V_{\text{aggreg}} = V(p)$.
- For each non-terminal node p, set V_{aggreg} to be the normalised sum of $V(p)$ and all V_{aggreg} of p's children.

$$V_{\text{aggreg}}(p) = \begin{cases} V(p) & \text{if } p \text{ is a terminal} \\ V(p) \oplus \bigoplus_{q \in p\text{'s children}} V_{\text{aggreg}}(q) & \text{otherwise} \end{cases} \tag{5.5}$$

Step 3. Propagate CVs downwards, starting from the root, performing CV contextualisation at each node (Figure 5.5c).[1]

- For each node p except the root, contextualise $V_{\text{aggreg}}(p)$ with $V\text{aggreg}$ of p's parent node:

$$V'_{\text{aggreg}}(p) = \begin{cases} V_{\text{aggreg}}(p) & \text{if } p \text{ is root node} \\ \gamma(V_{\text{aggreg}}(parent(p)),\ V_{\text{aggreg}}(p)\ ) & \text{otherwise} \end{cases} \tag{5.6}$$

- For each node p where $|\mathcal{L}(w_p)| > 1$, contextualise $V(p)$ with $V'_{\text{aggreg}}(p)$:

$$V'(p) = \begin{cases} V(p) & \text{if } |\mathcal{L}(w_p)| = 1 \\ \gamma(\ V'_{\text{aggreg}}(p),\ V(p)\ ) & \text{otherwise} \end{cases} \tag{5.7}$$

Step 4. Assign the sense $s \in \mathcal{L}(w_p)$ to w_p for each node p where $|\mathcal{L}(w_p)| > 1$, by selecting the s whose CV is closest to the $V'(p)$, i.e. select s that maximises

[1] The method described in (Lafourcade and Boitet, 2002) sets $V'(p) = \gamma(V(p), V(parent(p)))$ (Eq. 4.14). However, we found $\gamma(V(parent(p)), V(p))$ to give better results, and will use the latter calculation in our methods.

$\text{CSim}(V(s), V'(p))$.

In Figure 5.5d, $V(\text{bank\#1})$ is found to have a lower $\mathcal{D}_{\mathcal{A}}$ value (i.e. higher CSim value) with respect to $V'(\text{bank})$. Therefore, the sense *bank*#1 is assigned to *bank* in this particular SSTC.

It should be noted that all SSTCs sense-tagged using the vector propagation algorithm will still be manually checked by humans, as errors will most likely occur especially where the sense distinctions are quite fine and subtle. This is because such senses will have many common concept labels and themes. However, as our final aim is to work with *translations* rather than *sense numbers*, such sense-tagging errors may be ignored, particularly when both (incorrectly) assigned and actual correct senses resolve to the same translation in the target language.

5.3.4 Computing Profile CV for Sub-S-SSTCs

In all previous work using the CV model that we have reviewed, the CVs stored for lexicon entries were mostly constructed based on dictionary definitions. The knowledge about a lexical item can be enriched further by incorporating the *context* of its occurrences in corpora during the lexical item's CV construction.

A further note about the sub-S-SSTCs in the BKB is needed at this point. As shown in Figure 5.6, sub-S-SSTCs in the BKB are generated from the examples and indexed based on their English and Malay surface strings, POS and tree representation structures. Therefore, if the BKB contains occurrences of an English word (or phrase) that actually have different senses, but are translated to the same Malay text (i.e. sense ambiguity is carried over to Malay), then only one sub-S-SSTC is generated for the English-Malay translation. Hence, the BKB essentially indexes sub-S-SSTCs by the strings, POS and tree

structures (syntactic features) in both SL and TL, without looking at the sense numbers in either language (semantic features).

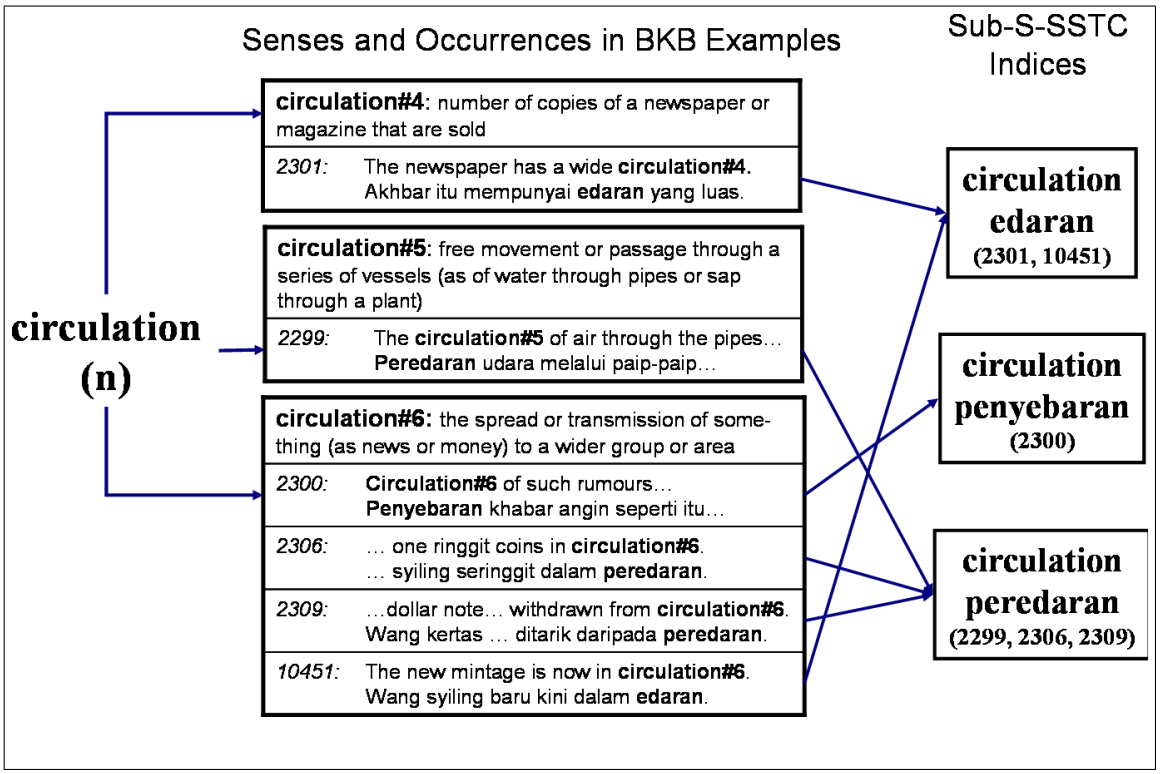


Figure 5.6: BKB Sub-S-SSTC Indices

We wish to enrich information about each sub-S-SSTC by compiling its résumé or *profile*, based on its "personal background" (lexicon definition) and "participation history" (usages in real examples as found in the BKB). A *profile CV* for a BKB sub-S-SSTC reflects the themes it encompasses. The profile CV encodes the themes of the lexical item associated with the sub-S-SSTC as defined in a lexicon, as well as the themes of the *context* in which the sub-S-SSTC appears in the BKB.

Let σ be a sub-S-SSTC in the BKB, and $\mathcal{E} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$ be the set of S-SSTC examples in which σ appears. context(ϵ_i, σ) is taken to be the SSTC nodes forming the context to σ in the English SSTC $\mathcal{S}$ of ϵ_i (see Figure 5.7). The whole tree structure is an SSTC tree representation of the English part of an example ϵ. Nodes enclosed in the

triangular shaded area form σ, and the remaining nodes constitute context(ϵ, σ). Let $R_{\mathcal{S}}$ and R_σ be the root nodes of the SSTC tree and the English subtree for σ respectively.

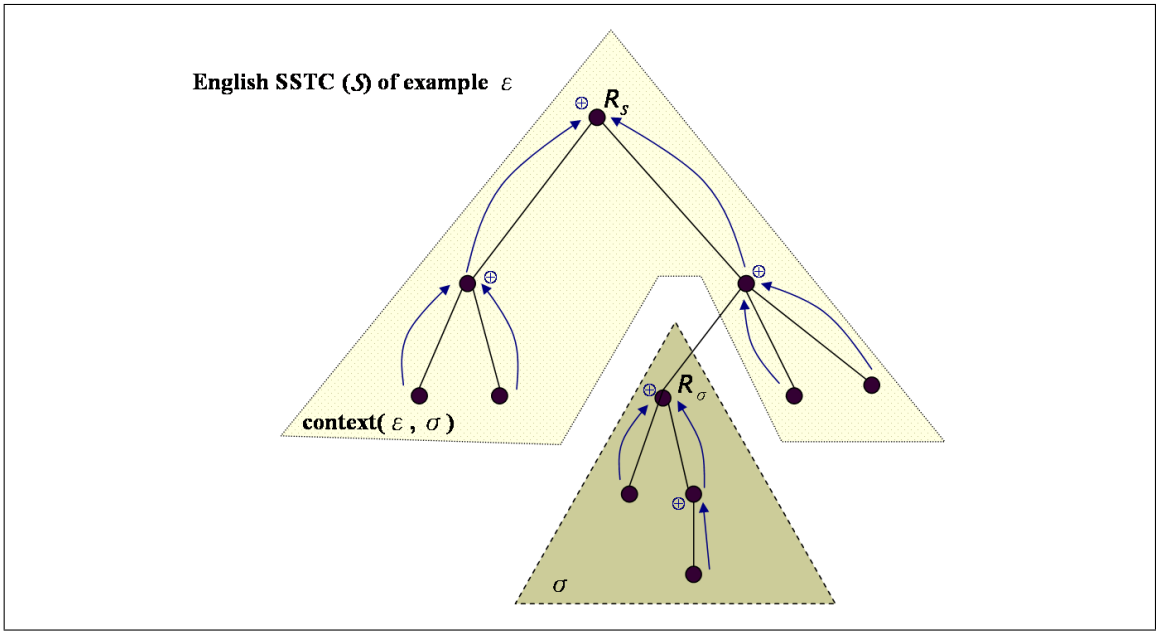


Figure 5.7: Computing a Profile CV for a Sub-S-SSTC from an Example

Assuming that $\mathcal{S}$ has been sense-tagged (see §5.3.3), we compute $V_{\text{context}}(\epsilon, \sigma)$ and $V_{\text{lex\_def}}(\epsilon, \sigma)$ using the upwards vector propagation method, starting from the terminal nodes towards the root:

- $V_{\text{context}}(\epsilon, \sigma)$:
  - For each terminal node $p \in$ context(ϵ, σ), set $V_{\text{aggreg}}(p) = V(p_s)$, where the occurrence of p in ϵ is tagged with sense s from $\mathcal{L}$, and $V(p_s)$ is the CV of s as described in §5.3.2.
  - For each non-terminal node $p \in$ context(ϵ, σ), set $V_{\text{aggreg}}(p)$ to be the normalised

sum of $V(p_s)$ and all V_{aggreg} of p's child nodes.

$$V_{\text{aggreg}}(p) = \begin{cases} V(p_s) & \text{if } p \text{ is terminal} \\ V(p_s) \oplus \bigoplus_{q \in p\text{'s children}} V_{\text{aggreg}}(q) & \text{otherwise.} \end{cases} \tag{5.8}$$

  - Set $V_{\text{context}}(\epsilon, \sigma) = V_{\text{aggreg}}(R_{\mathcal{S}})$. This is because the CV near at the root is representative of the themes contained in the text segment that we are interested in (Lafourcade and Boitet, 2002).

- $V_{\text{lex\_def}}(\epsilon, \sigma)$:
  - The above steps are repeated for the nodes *in* σ, i.e. the nodes enclosed in the grey area.
  - Set $V_{\text{lex\_def}}(\epsilon, \sigma) = V_{\text{aggreg}}(R_{\sigma})$.

Finally, we do not want the profile vector to skew towards senses that occur more frequently in the BKB, but when they *do*, we want to highlight the concepts that are more likely to appear in their contexts. Therefore, the profile vector for σ can now be expressed as the normalised sum of all $V_{\text{context}}(\epsilon, \sigma)$ and unique $V_{\text{lex\_def}}(\epsilon, \sigma)$.

$$V_{\text{profile}}(\sigma) = \bigoplus_{\epsilon \in \mathcal{E}} V_{\text{context}}(\epsilon, \sigma) \oplus \bigoplus_{\substack{\epsilon \in \mathcal{E} \\ V_{\text{lex\_def}} \text{ unique}}} V_{\text{lex\_def}}(\epsilon, \sigma) \tag{5.9}$$

Figure 5.8 illustrates the profile CV calculation process, with σ being the sub-S-SSTC *circulation* $\leftrightarrow$ *peredaran*. Given the three examples (numbered 2299, 2306 and 2309) in which this sub-S-SSTC occur, $V_{\text{lex\_def}}(\sigma, \epsilon)$ and $V_{\text{context}}(\sigma, \epsilon)$ is computed for each example. Their normalised summation will yield $V_{\text{lex\_def}}(\sigma)$ and $V_{\text{context}}(\sigma)$ respectively. Finally, V_{profile} is arrived at by summing and normalising these two CVs.

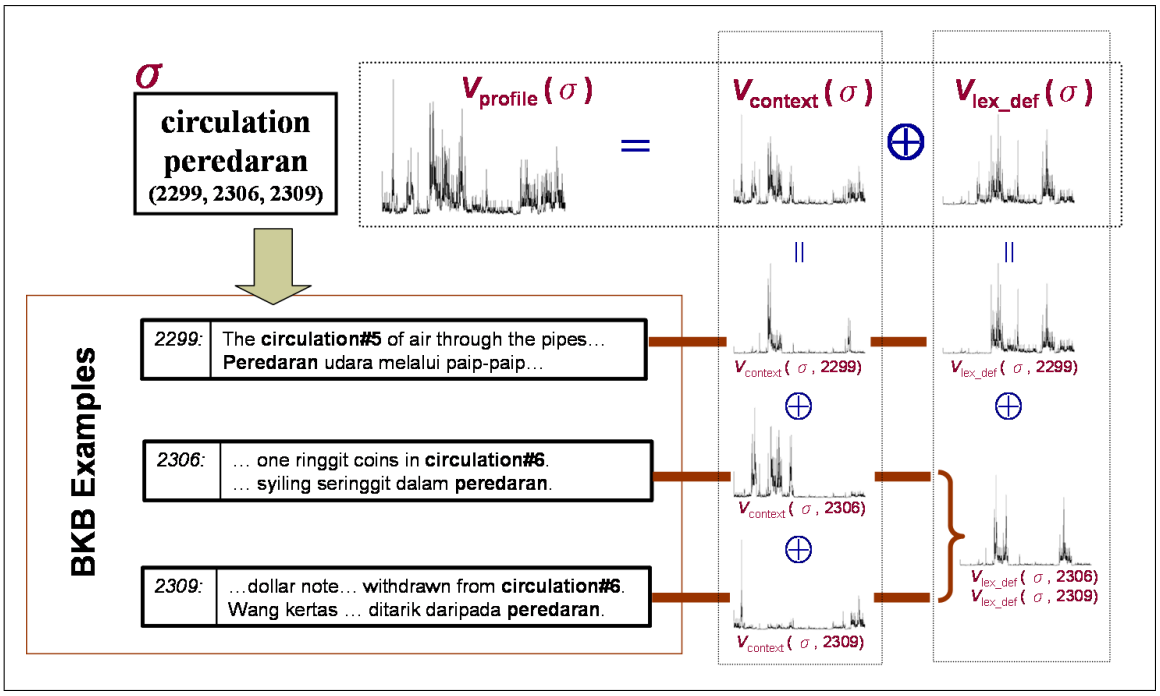


Figure 5.8: Computing V_{profile} for the Sub-S-SSTC *circulation* – *peredaran* from BKB examples

In our calculations, we consider the CV representing the lexicon definition ($V_{\text{lex\_def}}$) of a sub-S-SSTC and its context in the BKB examples (V_{context}) separately, so as to avoid "over-crowding" of concepts of frequently occurring word senses, which would give rise to an "biased" profile. Therefore, **circulation#5** and **circulation#6** contribute equally to $V_{\text{lex\_def}}$ for *circulation* ↔ *peredaran* in the example above.

5.4 Sub-S-SSTC Selection

In this section, we identify our "suspect" by running the "clues" gathered from the "crime scene" (input sentence) against the "suspect profile database" (profile CVs of sub-S-SSTCs). During the matching and selection phase of EBMT, sub-S-SSTCs in which the English parts match fragments in the input text are retrieved from the BKB. The need for disambiguation arises when sub-S-SSTCs with different Malay translations are returned. We will limit our scope to disambiguation of content words only. This include nouns, verbs, adjectives and adverbs: that is, head words that were tagged in §5.3.

Sato and Nagao (1990) states that two main factors determine the appropriateness of a translation unit:

- the size of the translation unit, and
- the similarity between the environments of the input fragment and that of the translation example.

The calculations and comparisons of CVs in this work pertains more to the second factor, although priority is given to translation units of longer sub-S-SSTCs during selection (see §5.4.2). We look for thematic closeness between the ambiguous[2] input words and the BKB contents. This is done by comparing the precomputed CV of the sub-S-SSTC and the CV of the input sentence, using the cosine similarity measure CSim between them (see Equation 4.2).

5.4.1 Input Text Pre-processing

The input text (usually a sentence) $Inp = (w_1, w_2, \ldots, w_n)$ is first POS-tagged, tokenised and lemmatised, and the syntactically matching sub-S-SSTCs are retrieved accordingly.

To prepare for the semantic similarity comparison, we create a trivial SSTC $\mathcal{S}_{\text{Inp}}$ for the input text: one in which the tree is made up of a root node with an empty label, and all w_i as its children nodes (see Figure 5.9). We then initialise $V(p)$ for each node p, and perform both upward and downward CV propagation on $\mathcal{S}_{\text{Inp}}$ (see steps 1–3 in §5.3.3). This makes related concepts in the input sentence "resonate" with one another and "stand out" in each $V'(p)$.

[2] again, "ambiguity" here refers to translation ambiguity, i.e. cases where more than one translation is possible. We therefore do not consider cases where sense ambiguity is carried over to the target language, as there would be only one possible translation.

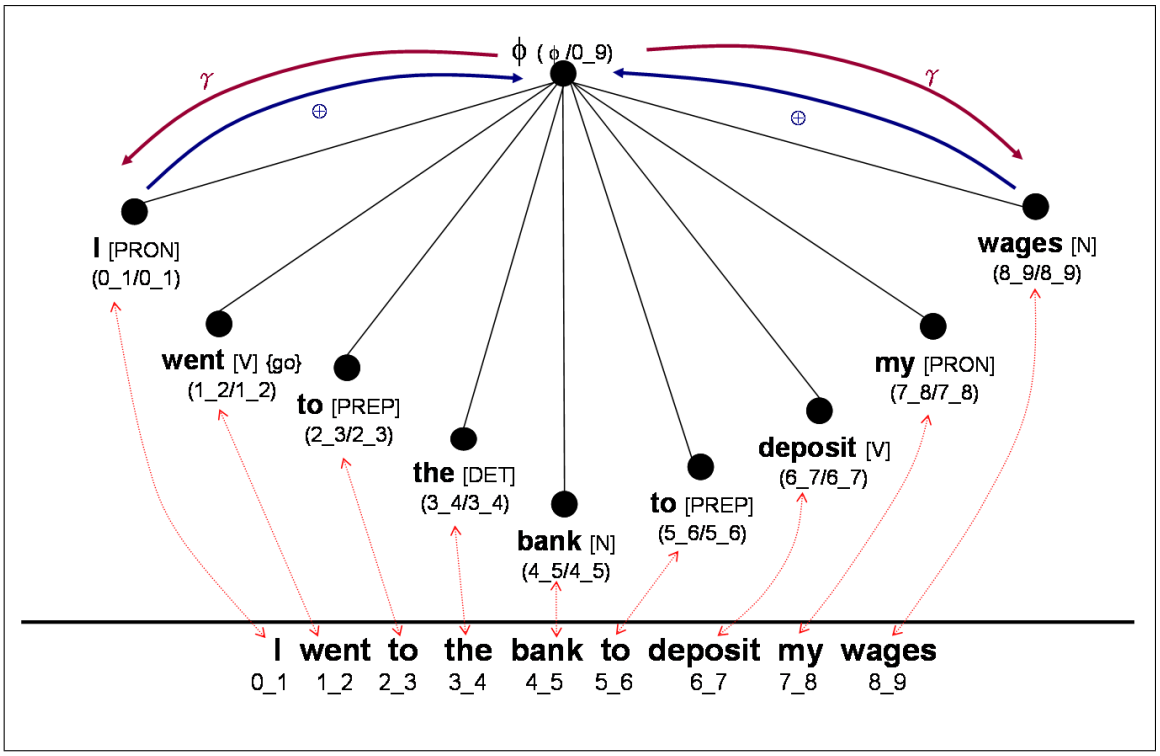


Figure 5.9: Pre-processing the Input Sentence

In the following discussion, we consider $V(w) = V'(p)$, where w is the substring corresponding to node p in $\mathcal{S}_{\text{Inp}}$, and $V'(p)$ is the CV tagged to node p after the vector propagation algorithm is executed.

5.4.2 Selection of Most Semantically Similar Sub-S-SSTC

The profile vectors can be considered as a database of "fingerprints" of the sub-S-SSTCs, against which the input sentence fragments will be matched during EBMT, when we attempt to identify the correct translations of words and phrases in the input sentence.

Disambiguation is performed for each substring of Inp, $h \subseteq subseq(w_1, w_2, \ldots, w_n)$ with multiple matching sub-S-SSTCs (and therefore multiple possible translations). The full algorithm is given in Figure 5.10.

In order to identify the best translation for a substring, we need to gather *clues* from the context words surrounding the substring in the input sentence. Priority is given to

POS-tag and tokenise input sentence Inp

Add all Inp tokens to *vTokens*

vOutput ← empty array

for l = length(Inp) to 1 **do**

  for all h = substring of Inp of length l constructed from *vTokens* **do**

    $\sigma_h \leftarrow$ null

    for all σ whose English string = h **do** /** Select the best σ for h **/

      if $\text{CSim}(V_{\text{profile}}(\sigma),\ V_{\text{clue}}(h)\ ) - \text{CSim}(V_{\text{profile}}\sigma_h,\ V_{\text{clue}}(h)\ ) > 0.001$ **then**

        $\sigma_h \leftarrow \sigma$

      else if $|\ \text{CSim}(\ V_{\text{profile}}(\sigma),\ V_{\text{clue}}(h)\ ) - \text{CSim}(\ V_{\text{profile}}(\sigma_h),\ V_{\text{clue}}(h)\ )| \leq 0.001$

          and freq(σ) > freq(σ_h) **then**

        $\sigma_h \leftarrow \sigma$

      end if

    end for

    if *vOutput* is empty or *vOutput* does not contain σ'

        whose English string h' overlaps with h **then**

      Add σ_h to *vOutput*

      Remove tokens $t \in h$ from *vTokens*

    else if length(h) = length(h') **then**

      if $\text{CSim}(V_{\text{profile}}(\sigma), V_{\text{clue}}(h)) - \text{CSim}(V_{\text{profile}}(\sigma'), V_{\text{clue}}(h')) > 0.001$ **then**

        Remove σ' from *vOutput*

        Add σ_h to *vOutput*

        Add tokens $t \in h' - h$ to *vTokens*

      end if

    else if $\text{CSim}(V_{\text{profile}}(\sigma), V_{\text{clue}}(h)) - \text{CSim}(V_{\text{profile}}(\sigma'), V_{\text{clue}}(h')) > 0.1$ **then**

      Remove σ' from *vOutput*

      Add σ_h to *vOutput*

      Add tokens $t \in h' - h$ to *vTokens*

    end if

  end for

end for

Recombine contents of *vOutput* into final output SSTC

Figure 5.10: Sub-S-SSTCS Selection Using Conceptual Vectors

sub-S-SSTCs of longer length (c.f. Sato and Nagao, 1990, §5.1). Starting with the longest substrings, for each h of varying length, we compute $V_{\text{clue}}(h)$ from the input sentence st as the normalised sum of CVs of words forming the context to h in Inp. However, if the context size is too small and contains less than a quarter of all content words in the input sentence, we take $V_{\text{clue}}(h)$ to be the normalised sum of CVs of *all* words.

$$V_{\text{clue}}(h) = \begin{cases} \bigoplus_{w_i \in Inp} V(w_i) & \text{if context is too small} \\ \bigoplus_{w_i \in Inp-h} V(w_i) & \text{otherwise} \end{cases} \tag{5.10}$$

Let $\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ be the set of matching sub-S-SSTCs for h, as well as other substrings of st of equal (or less) that overlap with h. To select sub-S-SSTCs that will be actually used to build the final output, we select σ_i such that CSim($V_{\text{profile}}(\sigma_i)$, $V_{\text{clue}}(h)$) is highest.

To illustrate, for the input sentence in Figure 5.9, Table 5.1 shows the possible sub-S-SSTCs retrieved for the ambiguous word *bank* and their CSim scores. Based on the CSim values, the sub-S-SSTC *the bank* ↔ *bank itu* will be selected for constructing the final output.

Table 5.1: Sub-S-SSTCs Matching ***bank*** in *I went to the* ***bank*** *to deposit my wages*

sub-S-S-SSTC ($\boldsymbol{\sigma}$)	CSim ($\boldsymbol{V}_{\text{profile}}(\boldsymbol{\sigma})$, $\boldsymbol{V}_{\text{clue}}(\boldsymbol{\sigma})$)
to the bank ↔ ke bank itu	0.8770
to the bank ↔ ke tebing	0.1957
the bank ↔ bank itu	**0.9029**
the bank ↔ tebing itu	0.2243

If the CSim measures of the top two ranking sub-S-SSTCs are too close, i.e. differing by less then a threshold value (say 0.001), this could mean that both sub-S-SSTCs refer

to the same sense, and are used in highly similar contexts (as stored in the BKB). The algorithm then selects the sub-S-SSTC with the higher frequency count, as it indicates that one translation is preferred over the other in similar contexts.

If, after this step, the sub-S-SSTC selected is for a substring h' that overlaps with h (but is not h itself), the tokens $h' - h$ will be returned to the pool of unmatched tokens or substrings for the next round of matching.

The process is repeated until we are left with a list of sub-S-SSTCs, the English SSTCs of which make up the original input sentence. The EBMT system can then take this list of sub-S-SSTCs, recombine them into a complete S-SSTC, and return its associated surface string as the final translation output.

5.5 Advantages

Some of the advantages of the approach used in this research include the following:

- **Ability to handle many-to-many *source word* → *translation* mappings.** This is achieved by having the translation selection algorithm choose from translations directly, bypassing sense numbers entirely. Sense distinction by sense numbers is used only in the data preparation phase.
- **Disambiguates and translates content words of all POS.** By tagging all content words with concept labels from a concept hierarchy, the sense-tagging and translation selection algorithm can process all of them in the same way.
- **Allows for bi-directional translation.** Even though only SSTCs of one language (say L_A) are sense-tagged, it is possible to use the same BKB for $L_A \rightarrow L_B$ translation as well as $L_B \rightarrow L_A$. The sub-S-SSTC selection algorithm described earlier

caters for $L_A \rightarrow L_B$ translation, but can be modified easily for $L_B \rightarrow L_A$ translation: sub-S-SSTCs with L_B substrings matching that of the input (L_B) sentence are retrieved. CSim scores are calculated using the same CVs associated with each sub-S-SSTC, even though the CVs were computed from the L_A SSTCs.

- **Production of lexical semantic data on two levels,** one on a sense-number level *a lá* dictionaries or lexicons, and another one on a translation-pair level. This separation makes it possible to add more information useful for each specific level.
- **New knowledge in the form of new examples.** The BKB can be updated with new usages and translations of a word or sense, by the addition of new translation examples in which they appear.
- **Avoids combinatorial effect of multiple ambiguities,** made possible by the CV propagation method.

5.6 Some Weaknesses

Apart from the advantages listed in the previous section, there are also some drawbacks to our approach:

- **Manual data preparation.** The biggest drawback of our approach is the need for manually prepared data, i.e. the tagging of lexicon sense entries with concept labels, and the checking of sense-tagged SSTCs.
- **Problematic translations of function words**. The approach here does not handle translation selection of function words, most notably prepositions.
- **Suitability of BKB examples.** An ambiguous word cannot be disambiguated or translated correctly if the desired sense and context of usage is not included in the

BKB (data sparseness). On the other hand, frivolously "throwing examples at the problem" might cause data "saturation", a situation in which the profile CV of a sub-S-SSTC contains too many prominent "peaks". This might cause the translation unit to be selected in any context.

- **Handling of new senses.** This is a problem common to all knowledge-based approaches. As natural language evolves, words may develop new senses and usages. If a sense is used in new contexts, possibly with a new translation, examples reflecting this usage can be added to the BKB, and the profile CVs updated automatically. This might, however, lead to data saturation if the new examples are not chosen appropriately (see previous item).

  If a word develops new senses, the lexicon would have to be updated as well, together with corresponding examples. These would all require some human effort, unless the system is able to detect the emergence of new senses, and attempts to create new entries and profiles for them.

5.7 Contributions

The main contributions of this work include:

- **Adaptation of a WSD approach for the specific aim of translation selection.** WSD is only an intermediate task that is common to many NLP problems, and WSD approaches usually have to be tailored to the needs of each problem. This research adapts Lafourcade's (2001) WSD approach using the CV model for the problem of translation unit selection in an EBMT system.

- **Proposal of primary and secondary concepts as concept-tagging guidelines.** These serve to help guide human taggers in choosing concepts to be assigned

to sense entries, rather than in an ad-hoc manner. Lafourcade's (2001) use of a thesaural concept associated with a head word is the equivalent to the "primary concept" used here, which is then supplemented by the "secondary concepts", determined based on the definition text or gloss. These guidelines can also be used for automated tagging of lexicons with concept labels in future.

- **Application of a concept hierarchy to the semantic tagging of verbs, adjectives and adverbs.** Most WSD work involving concept hierarchies tags only nouns with concepts or classes (usually the primary concepts), and therefore could only disambiguate content words of certain POS (usually nouns). By using the primary and secondary concept guidelines in this research, it was possible to assign concept labels to verbs, adjectives and adverbs as well. Therefore, the sense and translation of these words can also be disambiguated using the same sub-S-SSTC selection algorithm.

- **Production of lexical thematic information on two different levels,** i.e. one on a sense-number level as commonly found in dictionaries and lexicons, and another on a translation-pair level in a specific SL and TL. As different NLP tasks require different levels of granularity, the more suitable of the two semantic information repositories produced in our work may be reused as appropriate.

5.8 Summary

We have outlined an approach to enrich an S-SSTC-annotated BKB with semantic information, the aim of which is to facilitate translation selection during the EBMT process, as well as how the semantic information supports the latter process. The semantic information of lexical items, including that of lexicon sense entries and translation pairs, was expressed using Lafourcade's CV model. External linguistic resources, such as a lexicon

and a conceptual hierarchy, were used for this purpose.

Semantic information was treated on two levels during data preparation: one based on sense distinctions as listed by a lexicon (§5.3.1, §5.3.2 and §5.3.3), and another based on translations in a target language (§5.3.4). Semantic information on a translation-pair level is chosen over that of lexicon sense-number level for the translation selection task in the EBMT system. The translation selection task is performed on the basis of thematic similarity between the *profile* of a translation as gleaned from the BKB, and the *clues* yielded by the input sentence to be translated.

The next two chapters will discuss some issues and considerations during the implementation of the proposed methodology, as well as results of tests carried out on the implemented system.

CHAPTER 6

IMPLEMENTATION ISSUES AND CONSIDERATIONS

Having set down the design in the previous chapter, the implementation may now proceed. This chapter starts by describing the implementation environment, before going on to discuss some issues during implementation. Brief descriptions of external linguistic resources and tools used are also given.

6.1 Implementation Environment

The BKB, comprising the translation examples (stored as S-SSTCs), sub-S-SSTC indices and other relevant information, is stored as a relational database in the *MySQL*[1] database server environment. The EBMT system that interacts with the BKB, as well as manipulates the S-SSTC structures, is implemented using the Java programming language. POS-tagging and morphological analysis is performed using the commercial syntactic parser, *Machinese Syntax*, from Connexor.[2]

All implementation activities and experiments were carried out on an Intel Pentium 4 machine, with 2.8 GHz CPU and 512 MB RAM running the Linux 2.6.10 kernel and Sun Microsystem's Java 2 Platform Standard Edition (version 1.4.2).[3]

To implement the design outlined in Chapter 5, three main activities were undertaken:

[1] `http://www.mysql.com`
[2] `http://www.connexor.com/demo/syntax/`
[3] `http://java.sun.com/j2se/1.4.2/`

- preparing the concept-tagged lexicon as a flat text file,
- creating new *MySQL* tables and fields to store the CVs,
- writing or updating Java classes, which will perform the following tasks:
  - reading the flat text files,
  - constructing CVs,
  - propagating CVs on SSTCs,
  - populating *MySQL* fields and tables, and
  - selecting sub-S-SSTCs during the EBMT process.

6.2 Linguistic Resources

For want of a suitable lexicon tagged with semantic concepts, which was hard to come by, we developed one ourselves. As pointed out in §5.3.1, two additional linguistic resources are required for preparing the semantic data, namely a lexicon and a concept hierarchy. The resources chosen in our implementation are described below.

6.2.1 Concept Hierarchy: *GoiTaikei*

GoiTaikei (GT) (Ikehara *et al.*, 1999) is an electronic Japanese lexicon. It contains around 300,000 Japanese words categorised under 3,000 semantic classes in three hierarchies: common nouns, proper nouns, and "phenomena" (verbs, adjectives and adverbs). The Japanese words are marked with POS information and the semantic classes they belong to, while words in the "phenomenon" hierarchy are organised as a valency dictionary with selectional restrictions.

GT was developed for use with *ALT-J/E*, a Japanese-English transfer-based MT system (Ikehara *et al.*, 1991; Bond, 2001). Its common noun hierarchy (henceforth referred

to as the *GT hierarchy*) of 2,710 concepts is widely used by many Japanese researchers in NLP-related work, ranging from annotation of bilingual dictionaries (Bond *et al.*, 2001) to WSD (Baldwin *et al.*, 2001; Kawahara and Kurohashi, 2004; Shirai and Yagi, 2004) and machine translation (Ikehara *et al.*, 1991; Ogura *et al.*, 1999).

Each node in the *GT* hierarchy corresponds to a semantic class, and edges in the hierarchy represent `is-a` or `part-of` relations. Concepts which are antonyms to one another are usually sibling nodes. Figure 6.1 shows concepts from the top four (out of a maximum of twelve) levels in the hierarchy. The complete list, translated to English, is included in Appendix B.

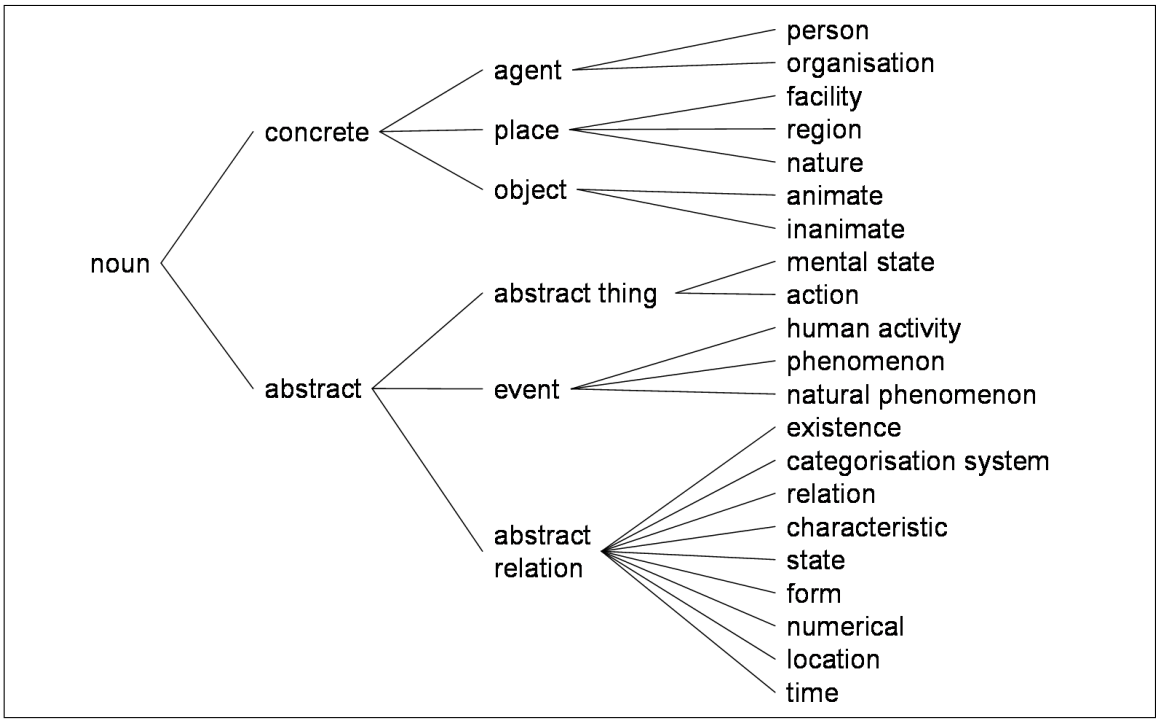


Figure 6.1: Top Four Levels of the *GT* Hierarchy

6.2.2 Lexicon: *WordNet*

WordNet (2005) is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory, developed by the Cognitive Science

Laboratory from Princeton University (Miller *et al.*, 1990). It is a popular lexical resource among the NLP research community, due to its broad coverage, rich lexical information, and free availability.

In *WordNet*, English nouns, verbs, adjectives and adverbs are organised into synonym sets (*synsets*), each representing one underlying lexical concept. The synonym sets are linked by various semantic relations, including hyponymy/hypernymy (`is-a`), meronymy/holonymy (`part-of`), antonymy and others. For example, the synsets containing the nouns *rhapsody* and *heroic verse* are hyponyms of the synset containing *epic*, as exemplified by Figure 6.2.

- {05983791} ‹noun.communication› epic poem#1, heroic poem#1, epic#1, epos#2
  — (a long narrative poem telling of a hero's deeds)
  ⇒ {05986524} ‹noun.communication› rhapsody#1
  — (an epic poem adapted for recitation)
  ⇒ {05989947} ‹noun.communication› heroic verse#1, heroic meter#1, heroic#1
  — (a verse form suited to the treatment of heroic or elevated themes; dactylic hexameter or iambic pentameter)
  ⇒ ...

Figure 6.2: A *WordNet* noun synset and its hyponyms

This research uses the sense repository and gloss texts provided by *WordNet 2.0*. Although the hypernymy/hyponymy structure for noun synsets can be regarded as a concept hierarchy (Agirre and Rigau, 1996; Asanoma, 2001), where the semantic conceptual class of each word sense is the synset containing it, the *WordNet* noun hierarchy was not used in this research. This is because the number of *WordNet* synsets, and hence number of concepts, is too large for CV construction and manipulation.

6.3 *WN-GT*: Tagging *WordNet* Entries with *GT* Concepts

The Japanese-English transfer lexicon used used in *ALT-J/E* (Ikehara *et al.*, 1991) contains entries tagged with *GT* concept labels, and Quah *et al.* (2001) produced a similar Malay-English lexicon. As we have no access to either lexicon, we had to develop our own concept-tagged lexicon from scratch.

The activity of tagging *WordNet 2.0* sense entries with *GT* concepts was done following the guidelines proposed in §5.3.1. To aid this tagging process, *XJDIC* (Breen, 2003), a two-way Japanese-English dictionary software application, was used to look up the Japanese translations of English words. The relevant concepts were then determined by looking up these Japanese translations in *GT*. This approach turned out well for determining the primary concepts of non-nouns. Derivations of nouns from adjectives, and of verbs from nouns, are very regular In the Japanese language, so much so that most Japanese dictionaries do not have individual entries for the derived forms (Breen, 2004). As *GT* also adopted this measure, the Japanese translations could all be found categorised in the common noun hierarchy.

The output is a flat text file containing entries similar to the following:

company 1 07568361 n 327,428,1892 [an institution created to conduct business]

where the format for each entry is

‹word› ‹sense number› ‹*WordNet* synset ID› ‹POS› ‹*GT* concepts›
[‹*WordNet* gloss›]

and ‹*GT* concepts› for each entry is determined following the guidelines described in

§5.3.1. The resulting text file is a lexicon with extra semantic information, in the form of *GT* concept tags. This lexicon (hereafter referred to as *WN-GT*[4]) can be reused in other NLP projects and tasks. The *WordNet* synset ID is retained so that the lexicon can be enriched with *WordNet* semantic relations, or be integrated with NLP projects already using *WordNet*, with ease in future.

As the tagging was done manually, only 130 *WordNet* entries were tagged and included in *WN-GT* for testing purposes, which are listed in Appendix C.

6.4 Construction of Conceptual Vectors from *WN-GT*

It is now possible to construct a CV for each entry in *WN-GT*, using its *GT* concept tags and the steps outlined in §5.3.2. To this end, the distances between *GT* concepts were pre-computed, and each CV is computed iteratively based on the resulting values.

6.4.1 Distances Between Concepts

The *GT* hierarchy was coded as an XML file, and distances between two concepts are computed as the shortest path connecting them in the hierarchy, following Rada *et al.*'s (1989) definition of conceptual distance.

There is the issue of storing these pre-computed conceptual distances for fast retrieval during CV construction. It took half an hour to construct one single CV with just two iterations, when each distance value is retrieved from a *MySQL* database table when needed.

To avoid this bottleneck, the conceptual distances are stored as a table in a plain text

[4]pronounced *'wing it'*

file, where each row contains numerical IDs of two concepts and the path length between them. During CV construction, the rows are read into memory as a single-dimension array for fast retrieval. In addition, since the GT hierarchy has a maximum depth of 12, which means that the maximum distance between any two concepts is 24, the array is declared to contain `unsigned byte` values. Each element occupies 1 byte = 8 bits in memory and has a value range of $[0, 2^8 - 1] = [0, 255]$, which is sufficient for storing the conceptual distances.

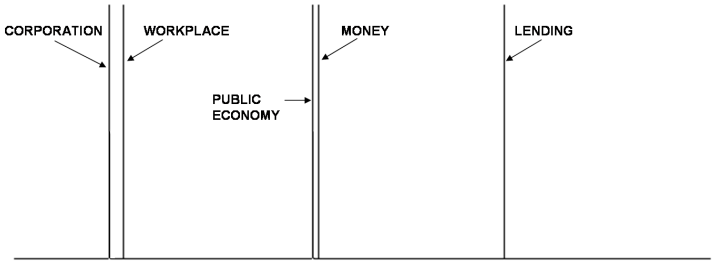
It is desirable to have a fast method for retrieving the distance between pairs of concepts in the chosen concept hierarchy. To this end, an efficient mapping procedure is devised and described in Appendix A. By using this mapping along with the in-memory array, the time taken to construct one CV with two iterations was successfully cut down to 0.5–1 second.

6.4.2 Iterative Computation of CVs

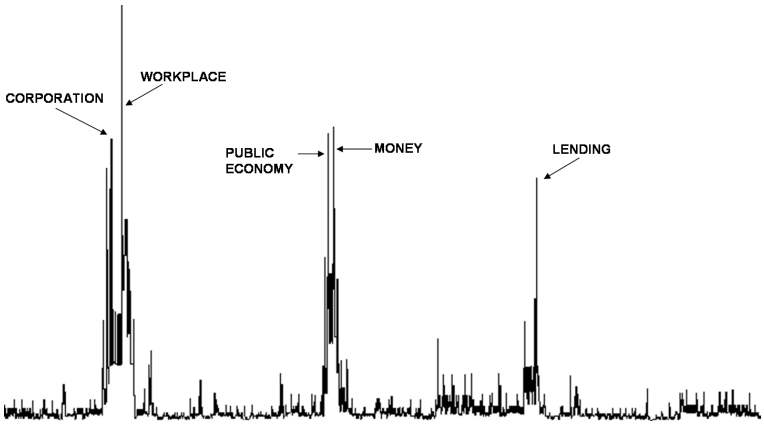
Since the top-level concepts are ancestors of all other concepts, their inclusion in $\mathcal{C}$, the set of concepts that forms the base of the CVs to be constructed, will cause non-trivial spikes corresponding to them in the resulting CVs. These spikes are actually noises that may skew the similarity measure (see §4.3). The following concepts are therefore excluded from $\mathcal{C}$, as there are very few lexicon entries that would have these as their related themes:

1: noun
- **2:** concrete
- **1000:** abstract
  - **1001:** abstract thing
  - **2422:** relation

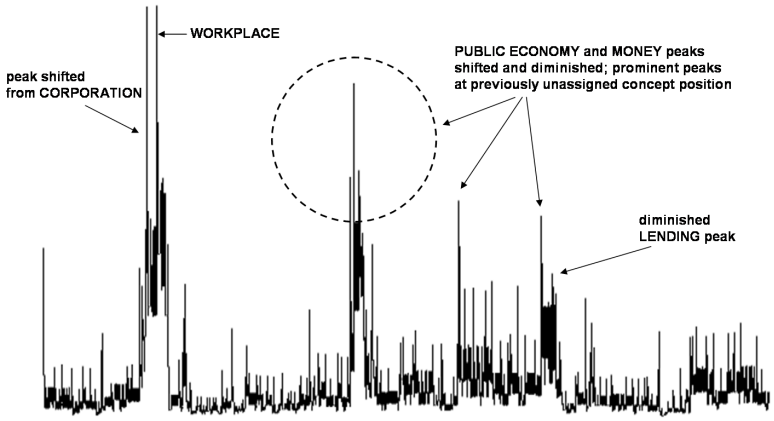
This leaves 2,705 concepts in $\mathcal{C}$, which means each CV will contain 2,705 elements.



(a) $V^0(bank\#1)$ as created from assigned GT concepts



(b) $V^2(bank\#1)$ shows good distribution of spikes where the most prominent ones are those in $V^0(bank\#1)$



(c) $V^3(bank\#1)$ shifts away from their initial positions in $V^0(bank\#1)$, or even vanished.

Figure 6.3: "Skewed" CVs after Too Many Iterations

§5.3.2 described how a CV can be computed iteratively from a lexicon entry tagged with $\mathcal{C}$ concepts (in this case, *WN-GT* entries). The iterative process causes the intensity of each concept to be propagated *a lá* ripples along the concept hierarchy, producing spikes in the CV. It is the presence of these spikes that makes the CSim measure (§4.3) possible.

It is expected that the highest spikes in each CV will occur at the positions corresponding to the *GT* concepts assigned to each lexicon entry. However, when the number of iterations is three or more, the highest spikes were found to have shifted away from their initial positions or are diminished, as Figure 6.3 demonstrates. Therefore, the number of iterations required to construct a CV is set to be *two* to optimise the "ripple effect" in CVs and to avoid deviation from the originally assigned concepts.

6.5 Extending SSTCs to Include Sense Numbers

The tree representation in each SSTC stored in the BKB is encoded in a Prolog-like string. Previously, each tree node is annotated with the base form, POS and inflectional tag of the lexical item that it contains. For example, the SSTC for the sentence

He applied to the bank for a loan.

is encoded as

```
applied[V]{apply@PAST}:1_2/0_9(He[PRON]{@SG3}:0_1/0_1,to[PREP]:2_3/2_5(
  bank[N]{bank@SG}:4_5/3_5(the[DET]:3_4/3_4)),for[PREP]:5_6/5_8(
  loan[N]{loan@SG}:7_8/6_8(a[DET]:6_7/6_7)),.[.]:8_9/8_9)
```

This encoding scheme is now extended to include the sense number (as listed in *WordNet* and *WN-GT*) for each lexical item, as in the following:

```
applied[V]{apply@PAST@3}:1_2/0_9(He[PRON]{@SG3}:0_1/0_1,to[PREP]:2_3/2_5(
  bank[N]{bank@SG@1}:4_5/3_5(the[DET]:3_4/3_4)),for[PREP]:5_6/5_8(
  loan[N]{loan@SG@1}:7_8/6_8(a[DET]:6_7/6_7)),.[.]:8_9/8_9)
```

For reasons of readability, we represent a sense-tagged SSTC in the following manner when its tree structure is unimportant:

He [PRON] *applied* [V]#3 *to* [PREP] *the* [DET] *bank* [N]#1 *for* [PREP] *a* [DET] *loan* [N]#1 . [.]

A selection of English SSTCs from the BKB was sense-tagged using the Java `SenseTagger` class (see §6.7) and manually checked. These are listed in Appendix D, together with their Malay translations.

6.6 Extending the BKB to Include CVs

As explained in §5.3.4, each sub-S-SSTC in the BKB is to be associated with a profile CV computed from *WN-GT* and the S-SSTCs containing it. A new field is therefore required in the *MySQL* table to store the computed CVs.

Each CV consists of 2,705 values of datatype `float`, and storing it as a comma-separated string of float values was found to consume around 33KB of disk space per CV. In order to reduce disk usage, each comma-separated string is compressed before storage, using the GZIP capabilities provided by the Java API. Each compressed CV now takes up 6KB on disk, and is decompressed upon retrieval during runtime.

A new relational database table is also created to store *WN-GT* entries, using the same compression scheme for storing the CV of each entry. This new table with the CVs can be used as alternative machine-tractable semantic lexicon, $WN\text{-}GT_{cv}$.

6.7 Java Classes for CV Manipulation on SSTCs

The existing Java classes were modified to accommodate the new encoding scheme for the tree in SSTCs described in §6.5. New Java classes were also implemented to perform the sense-tagging and CV manipulations on SSTCs, the most important class being `SenseTagger`. The UML class diagram of `SenseTagger` and related Java classes is shown in Figure 6.4 on page 92. Note that this class diagram shows only the more important classes, methods and properties for the sake of clarity.

The `SenseTagger` class associates each `TreeNode` in an SSTC `Tree` with its `SemanticInfo`, which contains `SenseEntry` objects as retrieved from *WN-GT*. Each `SenseEntry` encapsulates a sense entry s from *WN-GT*, and a `ConceptualVector` that represents $V(s)$ (§5.3.2). The main responsibilities of `SenseTagger` include:

- given an SSTC `Tree` **without** annotations of sense numbers, tag its `TreeNode`s with all possible senses listed in *WN-GT* (as `SenseEntry` objects) of its lexical item, based on the lemma and POS;
- given an SSTC `Tree` **with** sense numbers, tag its `TreeNode`s with a `SenseEntry` object corresponding to each node's lemma, POS and sense number;
- performing upward and downward propagation of `Conceptual Vector`s on `Tree`s (§5.3.3 and §5.3.4);
- performing WSD on a `Tree`, by assigning the most thematically similar `SenseEntry` to each `TreeNode` after `ConceptualVector` propagations (§5.3.3), and updating the sense numbers in the `TreeNode` accordingly.

The `SenseTagger` class was used to sense-tag English SSTCs in the BKB, as well as

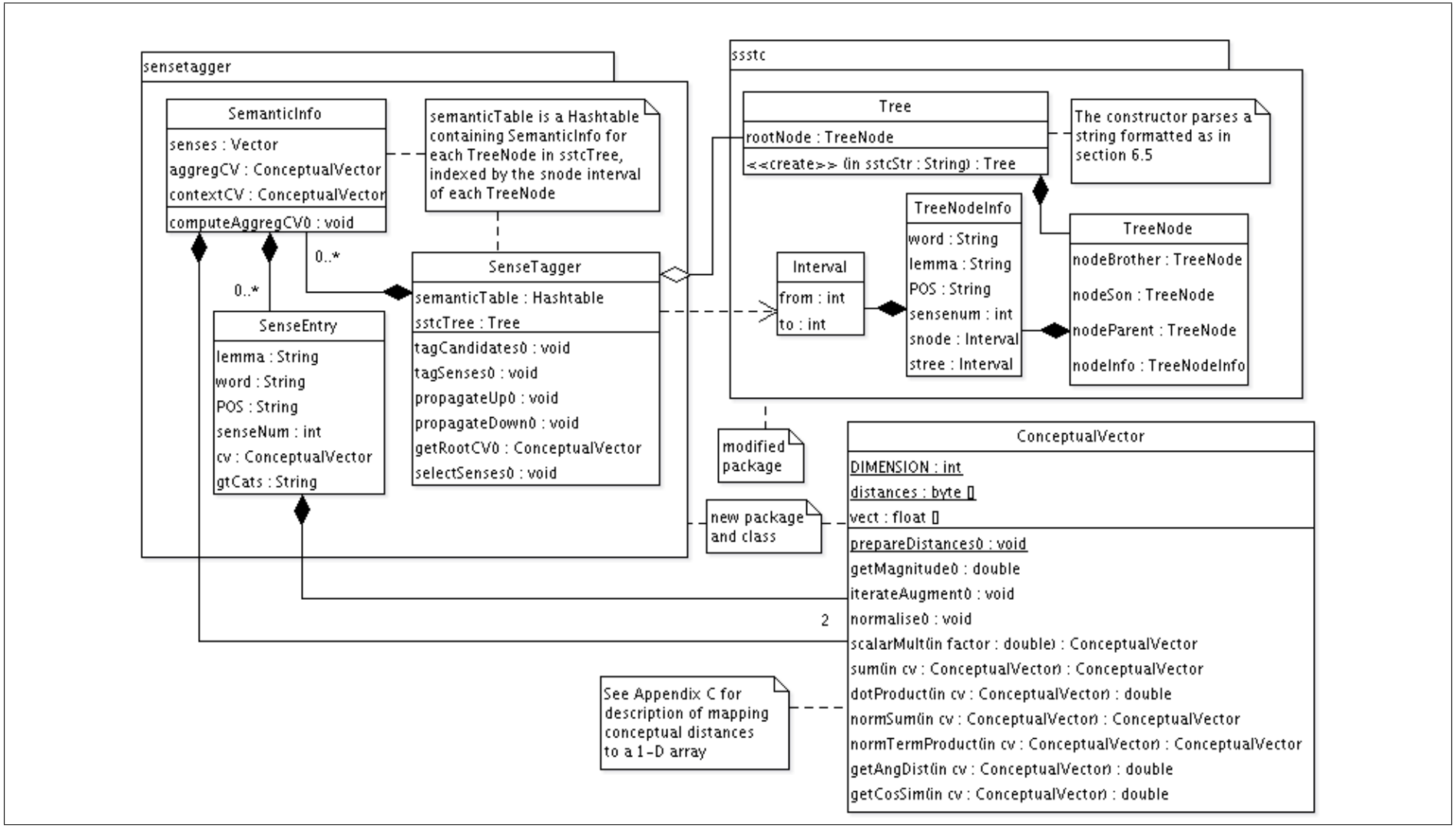


Figure 6.4: UML Class Diagram for `SenseTagger` and Other Important Java Classes

computing profile CVs for sub-S-SSTCs, as described in §5.3.3 and §5.3.4 respectively. Appendix D shows the `SenseTagger` outputs (see also §7.1.1, while Appendix E contains the processed sub-S-SSTCs[5], and the sense-tagged examples in which they appear.

6.8 Sub-S-SSTC Selection During EBMT

The existing Java classes implementing the EBMT system, described in Chapter 3, were modified to implement the sub-S-SSTC selection algorithm in Figure 5.10: the modification required was fairly straightforward. Although the S-SSTC framework allows for matching of TL sub-SSTCs for discontinuous SL substrings (and incomplete SL trees), only the matching of continuous SL substrings is implemented in this research.

6.9 Summary

We have described the activities undertaken to implement the methodologies outlined in Chapter 5 using modified or new Java classes, the most important being the `SenseTagger` class. Some implementation issues encountered were highlighted, as well as the decisions taken to overcome these. A brief account of the two external linguistic resources used, *WordNet* and *GoiTaikei*, was also included. The implementation activities also produced *WN-GT* and *WN-GT*$_{cv}$, two lexicons with additional semantic information, in the form of concept labels and CVs, respectively.

[5] whose English SSTC is made up of a single node and word

CHAPTER 7

RESULTS AND DISCUSSION

This chapter presents and discusses test results from the modified EBMT system, which uses a BKB enriched with semantic information, and an updated sub-S-SSTC matching algorithm (Chapters 5 and 6).

7.1 Experiments and Results

As the data preparation was done manually, it was only feasible to perform testing on a few selected ambiguous words. Two experiments were carried out.

In the *sense-tagging experiment*, BKB examples (S-SSTCs) containing the selected test words were first identified. *WordNet* sense entries[1] for all content words in these examples were tagged with *GT* concepts and entered into *WN-GT* (§5.3.1), and their CVs constructed accordingly (§5.3.2). Next, the chosen examples were sense-tagged (§5.3.3) using the `SenseTagger` tool (§6.7) with respect to the senses listed in *WN-GT*; the results were manually checked and corrected. Profile CVs for the corresponding sub-S-SSTCs were then computed (§5.3.4) to conclude the data preparation phase. The prepared material (except the profile CVs) can be found in Appendix E.

The other experiment, known as the *translation experiment*, involved using the modified EBMT system (referred to as $EBMT_{cv}$) to translate English sentences containing the ambiguous words. The output Malay translations were compared with those produced by

[1] in some cases, not all *WordNet* senses for a word were tagged as there were simply too many senses, and the distinction between them is often too subtle.

the old system (referred to as $EBMT_o$).

The next two sections present the results from the sense-tagging experiment and the translation experiment respectively.

7.1.1 Sense-Tagging Experiment Results

Thirty five English SSTCs were sense-tagged with `SenseTagger` and the outcomes were checked against those produced by a human tagger. In view of the large amount of data preparation work required, *WordNet* senses to be processed as *WN-GT* entries were chosen such that most SSTCs contained only one ambiguous word occurrence.

These SSTCs included 43 occurrences of 11 ambiguous words, where a word is considered to be ambiguous if *WN-GT* contained multiple sense entries of its lemma. For comparison, the baseline strategy always chooses the first sense listed in *WN-GT* — the sense with the highest frequency of occurrence. An instance of an ambiguous word is deemed to be tagged correctly if the assigned sense number agrees with that of the human tagger's.

The number of ambiguous word instances tagged correctly by `SenseTagger` and the baseline strategy is summarised below in Table 7.1.

Table 7.1: Sense-Tagging Experiment Results (43 ambiguous instances)

Strategy	Correct Instances	Accuracy (%)
`SenseTagger`	32	74.42
Baseline	15	34.84

See Appendix D for a breakdown of the results by each SSTC tagged.

7.1.2 Translation Experiment Results

$EBMT_{cv}$ was given 10 sentences to translate from English into Malay. Since the main aim in this research is the selection of TL words, the accuracy metric is the number of ambiguous word instances that were translated correctly. In this experiment, a word is considered ambiguous if it has multiple translations in the BKB, while "translated correctly" means the chosen Malay words are acceptable to a human reader, and there is no other sub-S-SSTCs in the BKB that would give a more satisfactory translation. (In other words, we are interested in translation ambiguity in this experiment.) The results for the same input from $EBMT_o$ were used as the baseline, where sub-S-SSTCs of the longest length and highest frequency of occurrence in the BKB are selected.

Table 7.2 summarises the number of correctly translated ambiguous words by both $EBMT_{cv}$ and $EBMT_o$. For comparison, the experiment was also repeated for two variations of $EBMT_{cv}$: $EBMT_{cv\text{-d}}$ selects translation units based on the CSim value between the "clue" CV of the input and only the lexical definition CV of the translation unit; while $EBMT_{cv\text{-c}}$ uses the CSim value between the "clue" CV of the input and only the context CV of the translation unit.

For more detailed results, including the English inputs and their translations produced by the systems, see Appendix F.

Table 7.2: Translation Experiment Results

System	**Test Word (No. of Instances in Inputs)**									
	bank (6)		circulation (3)		deposit (3)		stock (1)		**All (13)**	
	No.	%	No.	%	No.	%	No.	%	**No.**	%
$EBMT_{cv}$	6	100.00	3	100.00	3	100.00	1	100.00	13	100.00
$EBMT_{cv\text{-d}}$	6	100.00	1	33.33	1	33.33	1	100.00	9	69.23
$EBMT_{cv\text{-c}}$	5	83.33	2	66.67	3	100.00	0	0.00	10	76.92
$EBMT_o$	2	33.33	1	33.33	1	33.33	0	0.00	4	30.77

7.2 Discussion

The overall results from both experiments were favourable, thereby confirming that the CV model (Chapter 4) is indeed useful for WSD and translation selection purposes. In particular, $EBMT_{cv}$ was able to produce Malay translations that better convey the meaning of the input text compared to $EBMT_o$, which always selected more frequently-occurring translation words. The results also indicate that the use of V_{profile} produces more satisfactory translations compared to using either $V_{\text{lex\_def}}$ or V_{context} in isolation. In other words, the translation selection in $EBMT_{cv}$ has indeed improved because of the use of parallel corpus (in the form of the BKB) in addition to dictionary definitions.

7.2.1 Concept-based Matching vs Word-based Matching

By using concepts instead of words as the base of CVs, words of similar meanings are generalised under common groupings. This gives better coverage for both `SenseTagger` and $EBMT_{cv}$. For instance, given the following translation example and English inputs:

E: *to **deposit** one's wages in the bank.*

M: ***menyimpan wang** gaji seseorang di bank.*

Input A: *I went to the bank to **deposit** my wages.*

Input B: *I went to the bank to **deposit** my salary.*

$EBMT_o$, using only word-based matching, would correctly translate *deposit* in Input A as *menyimpan wang*, but fail to do so for Input B. This is because the BKB has seen only examples containing both "wages" and "bank" in the same example, but not "salary" and "bank". In contrast, $EBMT_{cv}$ selects *menyimpan wang* for both inputs, as both *wages* and *salary* share the same concepts MONEY and REMUNERATION.

An even more revealing test result is the the following translation by $EBMT_{cv}$:

> Input: *He drowned near the* ***bank****.*
>
> Output: *Dia mati lemas dekat* ***tebing****.*

Even though there is no translation example containing both the words *bank* and *drown*, but because both share WATER as a related concept, $EBMT_{cv}$ chooses *bank* $\leftrightarrow$ *tebing* over *bank* $\leftrightarrow$ *bank*. This would not have been possible with a simple word-based matching strategy.

However, this approach of reducing words to concepts also means that only content words contribute towards context information. As function words, which are much used in word-as-feature approaches (especially those using machine learning) do not have associated concepts, they are ignored in our approach, even though such words are the most helpful in some cases.

On another note, the approach adopted in this research is not strictly one using "bags" of concepts: the profile CVs (V_{profile} and V_{clue}) are constructed via propagation of CVs up a tree, where the sums of CVs of children nodes are normalised at each non-terminal node. Nevertheless, CVs of all chidren nodes of an internal node are treated as having equal weights. In view of research findings that words of different POS are best disambiguated using different classes of context words (Audibert, 2004), it may be interesting to see if and how varying weights, depending on the POS of individual lexical items, can be assigned to CVs to improve the accuracy of both `SenseTagger` and $EBMT_{cv}$.

7.2.2 V_{profile} vs $V_{\text{lex\_def}}$

Although much can be gleaned from dictionary definitions or glosses, it is obvious that CVs constructed from *WN-GT* entries ($V_{\text{lex\_def}}$) contain less features compared to V_{profile}, which incorporate concepts of the context surrounding a word occurrence.

To illustrate, a dictionary definition of *bank*#1 (p. 101) is unlikely to mention *jewellery*. Hence, `SenseTagger` may well wrongly assign *bank*#2 to the instance in *"The jewellery is kept in the **bank**"*. Conversely, if a similar example existed in the BKB, $V_{\text{profile}}(bank\#1)$ would capture themes related to *jewellery*. $EBMT_{cv}$ would then be better-equipped to reject the Malay translation *tebing* for the same input.

Therefore, the use of a corpus does indeed enhance the "profile" of a lexical item. It is expected that there will be an increase in `SenseTagger`'s accuracy rate if V_{profile} of a word sense is used in place of $V_{\text{lex\_def}}$. This was not done in the sense-tagging experiment because the data prepared was not sufficient to construct V_{profile} for more sense entries.

7.2.3 Multiple Ambiguous Words

Most of the test inputs, both for the sense-tagging and translation examples, contained a single ambiguous word. Nevertheless, when faced with inputs containing multiple ambiguous words, both `SenseTagger` and $EBMT_{cv}$ were able to select a sense and translation for all ambiguous instances, as the CV model causes related concepts across all lexical items to "resonate" with one another. Tables 7.3 and 7.4 show two such test inputs. The sense and sub-S-SSTC selected by `SenseTagger` and $EBMT_{cv}$ are highlighted in grey, while those marked with † indicate that they are acceptable to a human reader.

As mentioned in §4.5 (p. 51), `SenseTagger` — using a CV propagation approach to

Table 7.3: Sense-tagging an SSTC with 2 ambiguous words:
He [PRON] *claimed* [V] ***derivation*** **[n]** *from* [PREP] *French* [ADJ] ***stock*** **[n]**.

derivation #***n***	CSim
1	0.7295
†5	†0.8774

stock #***n***	CSim
1	0.5686
3	0.5464
†6	†0.7292

Table 7.4: Translating an input with 2 ambiguous words:
I [PRON] *went* [V] *to* [PREP] *my* [DET] ***bank*** **[n]** *to* [PREP] ***deposit*** **[v]** *my* [PRON] *salary* [N].

English Substring	Candidate Malay Substring	CSim
† bank	† bank	0.8510
bank	tebing	0.3400
deposit	membayar wang muka	0.6615
† deposit	† menyimpan wang	0.8790
deposited	meletakkan	0.1685
deposited	melonggokkan	0.2635
deposited	terlonggok	0.2837

WSD — disambiguates all test words in the input in two passes: once up the SSTC tree, another down. Its disambiguation window is the entire input sentence, in contrast to to other WSD approaches that uses n-grams as the context: $EBMT_{cv}$ tries contexts of different sizes to decide on the semantically closest sub-S-SSTC.

7.2.4 Homonymy vs Polysemy

Recall that there are two types of lexical sense ambiguity: homonymy and polysemy. *Homonyms* are words spelt the same way, but have unrelated meanings, while *polysemes* have multiple related meanings. `SenseTagger` performed better at disambiguating homonyms compared to polysemes, as is the fact with most WSD approaches. This is mainly due to the fact that homonyms involve very different concepts, and are easily distinguished

using CVs. A good example is *bank*:

bank 1 07909067 n 374, 428, 1170, 1190, 1910 [a financial institution that accepts deposits and channels the money into lending activities]

bank 2 08639924 n 490, 495, 748, 2667 [sloping land (especially the slope beside a body of water)]

As the concepts for these two senses are in different parts of the concept hierarchy, both `SenseTagger` and $EBMT_{cv}$ are able to choose the correct sense and translation for *bank* with *money* and *river* as the contexts respectively.

Polysemes, on the other hand, have related meanings, and are therefore often associated with very similar sets of concepts and themes. The distinctions between senses are at times so subtle, that it is difficult even for human taggers to settle on[2] the most suitable sense (Kilgariff, 2001). *Circulation* is one such potentially confusing word:

circulation 1 05871897 n 1115, 920, 2151 [the dissemination of copies of periodicals (as newspapers or magazines)]

circulation 4 12826961 n 1900, 1115, 2588 [number of copies of a newspaper or magazine that are sold]

Given the word *newspaper* as context (as in *"The newspaper has a wide circulation"*), both senses seem equally feasible under the CV model.

From another point of view, although the `SenseTagger`'s accuracy score of 74.42% is not perfect, especially when compared to some *SENSEVAL* systems scoring over 90%, this might not be such a serious issue as initially thought. Since the final aim is to work with *translations* rather than *sense numbers*, such sense-tagging errors may be ignored, particularly when the incorrectly assigned and actual correct senses translate to the same

[2]even harder for a group of taggers to *agree* on

TL words. However, the same problem is still very valid for $EBMT_{cv}$.

7.3 Reasons of WSD and Translation Error

Some reasons for the errors in the disambiguation and translation outputs are summarised below:

- **Lack of use of linguistic information.** The current approach uses only POS information and word lemmas in the (English) input sentence. However, in some cases of ambiguity, especially for polysemes, linguistic information (such as syntactic roles) may be more effective in disambiguation than concepts of contexts.
- **Loss of word-feature details.** As only the related concepts of content words are harvested, surface-level word features (e.g. morphological inflections) might be lost and function words ignored. Again, both may provide powerful clues for some cases of ambiguities, such as those involving multi-word verbs.
- **Data sparseness.** A word might be incorrectly translated because of insufficient knowledge, i.e. there is no (or not enough) example for a certain usage of a sense or translation pair.
- **Overlapping concepts.** This occurs when the V_{profile} of different sub-S-SSTCs have "peaks" in overlapping regions, especially for polysemes, as illustrated by the example in §7.2.2.
- **Errors in POS-tagging and lemmatising.** As these are done *before* the sub-S-SSTC selection phase, any errors would cause the selection to be done on a wrong candidate set of translations.

7.4 Summary

Two experiments were carried out to test how well `SenseTagger` and $EBMT_{cv}$ perform at WSD and translation selection respectively. Both showed satisfactory results, proving the validity of the approach adopting the CV model, as well as the use of both lexicon and corpus. In particular, $EBMT_{cv}$ produced better translations than the old system, which used only substring length and frequency to select translation units from the BKB. Several factors were identified as the cause of errors, and the next chapter will propose some measures to overcome them as future work.

CHAPTER 8

SUMMARY AND FUTURE WORK

This final chapter summarises the work undertaken in this research, before concluding by outlining future research directions where the methodology can be improved or used in other NLP research.

8.1 Research Summary

In order to obtain translation outputs of good quality from MT systems, words in the SL input text with multiple possible translations in the TL must be translated correctly, to reflect the meaning of the original text. This research aims to add such capabilities to an EBMT system, where the translation examples in its BKB are annotated with the S-SSTC schema (Al-Adhaileh and Tang, 1999; Al-Adhaileh *et al.*, 2002).

During translation run-time, disambiguation was performed between *translations* instead of *sense numbers*, as a sense number is not enough to determine the correct translation for an ambiguous word. All senses of an SL word may have the same translation in a TL; at other times one SL word sense may have different translations when used in different situations. Hence the decision to bypass sense-number tagging during the translation process.

The problem, therefore, is to select the best sub-S-SSTCs in the BKB, given an input SL text (usually a sentence) containing substrings that have multiple possible translation (with different meanings) in the TL. The approach used is a hybrid of knowledge- and

corpus-based approaches, using information from dictionary definitions, concept hierarchies and a corpus.

Concepts, rather than words, were used as the basis for matching and measuring semantic similarity. Word senses from a lexicon are first tagged with concept labels from a semantic concept hierarchy, which may originate from a thesaurus or ontology. To guide this tagging process, the notions of *primary* and *secondary concepts* for nouns, verbs, adjectives and adverbs were proposed. The conceptual themes of each word sense was then encoded as a conceptual vector ($V_{\text{lex\_def}}$) (Lafourcade, 2001; Lafourcade and Boitet, 2002). After the translation examples were sense-tagged using the CV propagation method, "profile" CVs (V_{profile}) are then compiled for translation pairs (sub-S-SSTCs from the BKB). This was built from $V_{\text{lex\_def}}$ of the source word in question and those of the context words surrounding it as found in the BKB. During the EBMT process, the matching algorithm computes "clue" CVs (V_{clue}) from the input text based on lexicon senses, and selects sub-S-SSTCs from the BKB that are of greater lengths and have a higher cosine similarity value between V_{clue} and $V_{\text{lex\_def}}$ of the sub-S-SSTCs.

The outcomes of this research include the following:

- *WN-GT*, a lexicon in which a subset of *WordNet* sense entries are tagged with concept labels from the *GoiTaikei* common noun hierarchy;
- a suite of tools written in the Java language for the construction and manipulation of CVs, including `SenseTagger`, a WSD tool which implements Lafourcade and Boitet's (2002) CV propagation algorithm;
- a BKB annotated with the S-SSTC schema, enriched with semantic lexical information in the form of sense-tagged SSTCs and conceptual vectors for (substring)

translation pairs;

- an improved EBMT system, $EBMT_{cv}$, the matching algorithm of which uses the newly-enriched BKB to select sub-S-SSTCs that would produce translations whose meaning is closer to that of the input sentence.

Table 8.1 sums up the contributions, advantages and weaknesses of this research.

8.2 Future Work

This section presents some possible ways of improving on the work described in this thesis, as well as suggestions of future directions of investigation. Most of these proposals involve further study of the conceptual vector model and its applications.

8.2.1 Automatic Construction of *WN-GT*

The biggest drawback of the approach described in this work is that it requires manual data preparation, especially the assignment of concept labels that constitute *WN-GT* entries. However, this process can be automated by bootstrapping from a set of manually prepared entries, and applying the primary and secondary concept guidelines.

The following subsections describe a few interesting issues that should be considered in such a bootstrapping process.

8.2.1(a) Sense-tagging of Definition Text

Since the definition texts or glosses are themselves written in natural language, WSD needs to be performed first. The `SenseTagger` tool can be of use here. Therefore, it is desirable to improve the accuracy of the tool (§8.2.4), which in turn needs more *WN-GT* entries.

Table 8.1: Contributions, Advantages and Weaknesses

Contributions	**Advantages**	**Weaknesses**
• Adaptation of a WSD approach for the specific aim of translation selection. • Proposal of primary and secondary concepts as concept-tagging guidelines. • Application of a concept hierarchy to the semantic tagging of nouns, verbs, adjectives and adverbs, instead of only nouns, thereby enabling all such words to be disambiguated (both sense-number and translation ambiguity) in the same manner. • Production of lexical semantic knowledge on two different levels, i.e. sense number level (for the SL words) and the SL $\leftrightarrow$ TL translation pair level.	• Able to handle many-to-many source word $\rightarrow$ translation mappings. • Disambiguates and translates content words of all POS. • Allows for bi-directional translation, even though SSTCs in only one language was sense-tagged. • A choice of lexical semantic repository on two different levels that are reusable by other NLP applications, according to their needs. • New usages and translations can be added in the form of new BKB examples. • Avoids combinatorial effect of multiple ambiguities.	• Requires manual data preparation. • Does not handle the translations of function words e.g. prepositions. • Suitability of BKB examples, which may be too sparse or too noisy. • Current implementation does not cater for detection of new senses.

This indicate an iterative process, where the concept-tagging of sense entries and sense-tagging of definition texts are repeated, if the data acquisition is to be fully automated without hand-checking.

8.2.1(b) Guidelines of Primary and Secondary Concepts

The notions of primary and secondary concepts might be refined further, so that they are better suited to be used by computer systems. For example, given an upper ontology which is entirely linked to a lexicon (see §8.2.1(d)), the secondary concepts of a sense s can be defined as those of the ontological properties of s. Domain codes, provided in some dictionaries, can also be included, although they may need to be mapped into the chosen concept hierarchy first.

8.2.1(c) "Noise" Words in Definition Text

Care should be taken to weed out "noise" words in definition texts to eliminate trivial concepts. For example, the definition of *drown* reads

> die from being submerged in water, getting water into the lungs, and asphyxiating

The verbs *being* and *getting*, despite being content words, do not contribute any significant concepts to the meaning of *drown*. Therefore, concepts of such words should not be included as secondary concepts. Rather than having a list of "blacklisted' words, though, further heuristics might be needed to determine if a word in the definition is really trivial or otherwise.

8.2.1(d) Alternative Concept Hierarchies

It is unclear if the choice of concept hierarchy has any significant effects on the results. Future research may therefore undertake such a survey. Possible alternative concept hierarchies include:

- the noun `is-a` hierarchies in *WordNet*;
- the *Suggested Upper Merged Ontology (SUMO)* (Niles and Pease, 2001; Pease, 2005), which has been linked to *WordNet* in the *KSMSA* project (Ševčenko, 2003, 2004);
- the *OpenCyc* project (Reed and Lenat, 2002; OpenCyc, 2005).

Note that the number of concepts and classes in some of the above resources are very huge. This can either be overcome with more efficient computing resources, excluding some top level concepts, or collapsing the lower levels.

These ontologies may also provide an alternative to the automatic acquisition of more *WN-GT* entries. For instance, aligning *GT* with *SUMO* (which may require less effort than annotating individual *WordNet* entries with *GT* concepts) will, at the very minimum, yield the primary concepts for the *WordNet* nouns.

8.2.2 Improving Quality of CVs

The quality of the CVs (both $V_{\mathcal{L}}$ for a sense and V_{profile} for a sub-S-SSTC) needs to be improved to achieve higher accuracy rates for both `SenseTagger` and $EBMT_{cv}$. Some possible ways to do this are suggested below.

8.2.2(a) CVs of *WN-GT* Sense Entries ($V_{\mathcal{L}}$)

During the iterative augmentation of CVs from a list of concepts for each lexicon sense entry, all concepts were considered as having equal weights. It would be interesting to see if some concepts should be given a bigger weight than others for the resulting CV to better reflect the sense, and on what basis. This might include syntactic roles resulting from a syntactic analysis, and the number of times a concept crops up among the primary and secondary concepts. In addition, the CV computation process (§5.3.2) may also be modified to eradicate the "peak-shift" phenomenon (Figure 6.3).

8.2.2(b) CVs of Sub-S-SSTCs ($V_{\mathbf{profile}}$)

As mentioned in §5.6, the choice of translation examples in the BKB will affect the "noise" level of the profile CVs. To overcome this, a mechanism for detecting thematically "noisy" examples – such as

> *he fainted just after coming out of the assembly hall*

where *faint* is remotely related to *assembly* and *hall* – is needed. The mechanism will need to discern between useful and trivial concepts *within* an example, with respect to the word for which the profile is to be built. For instance, in

> *Only one parachutist succeeded in landing on the near bank of the river.*

parachutist and *landing* are good profile candidates for each other, as are *bank* and *river*; but it might not be appropriate for *parachutist* to participate in *bank*'s profile.

However, over-zealous exclusion of whole examples for CV construction would lead to

the inability to detect new senses and translations: see §8.2.3.

8.2.3 Detection of New Senses and Translations

Another line of research is to have the system automatically "flag" possible new word senses, usages and translations as it processes new translation examples, especially those harvested from web pages and general articles.

To illustrate, before junk e-mails became widespread, the word *spam* was associated only with FOOD. It has now developed a new sense that is associated with MESSAGE, COMPUTER, and more. Such new usages are often reflected in corpora, before they are included officially in any dictionary. Automatic processing of translation examples with CVs may detect new senses and translations.

8.2.4 A Better `SenseTagger`

The `SenseTagger` tool is a useful one, which may be used in other NLP applications that require disambiguation at a sense-number level. It is therefore important to improve its accuracy. One way of doing this is to use V_{profile} instead of $V_{\mathcal{L}}$ for each sense entry, as the former contains more thematic information than the latter. However, this is only possible with a larger set of data (see §7.2.2).

A further strategy is to allocate different weights to tree nodes containing lexical items of different POS, during calculation of V_{context}, depending on the POS of the sub-SSTC's root for which the profile is to be compiled. This is in line with Yarowsky's (1993) and Audibert's (2004) findings that different types of information offer more disambiguation clues for different classes of ambiguous words.

8.2.5 Multilingual Semantic Translation Dictionary

The list of sub-S-SSTC that was produced resembles a bilingual translation dictionary, for both word- and phrase-levels. With the addition of translation examples in new languages, this could be the beginning of the construction of a multilingual translation dictionary, the CV contents of which making it suitable for use in NLP applications. The resulting data and techniques developed from such future work can either be used on their own, or contribute towards other multilingual dictionary projects, such as the *Papillon* initiative (Mangeot and Sérasset, 2005).

8.2.6 Matching Algorithm in $EBMT_{cv}$

The overall sub-S-SSTC matching and selection algorithm in $EBMT_{cv}$ can be refined and improved, by integrating the CSim measure with other strategies. In this respect, future research can consider phrase patterns, selectional restrictions, multiword-verbs, implementation of discontinuous substring matching, and the use of the TL corpus. One would need to give attention to how all these different factors can be integrated to optimise the final matching score.

8.2.7 Translation Dictionary as Backup

There may be times when the relevant senses are present in the lexicon, but the BKB contains no usage of a certain sense (data sparseness). The EBMT system is only able to choose from translations occurring in the BKB, even if it conveys the wrong meaning. It may be possible to have a "confidence" level for the CSim value, such that if the CSim of the selected sub-S-SSTC is not satisfactory enough, the EBMT system will then turn to a conventional translation dictionary (containing all word senses) and perform WSD to select a new translation.

REFERENCES

(2004) *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*. Pennsylvania: Association for Computational Linguistics.

Agirre, E. and Martinez, D. (2001) Knowledge Sources for Word Sense Disambiguation. In: *Proceedings of the 4th International Conference on Text, Speech and Dialogue (TSD '01)*. London, UK: Springer-Verlag, pp. 1–10.

Agirre, E. and Rigau, G. (1996) Word Sense Disambiguation Using Conceptual Density. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING 96)*, Copenhagen, Denmark.

Al-Adhaileh, M. H. and Tang, E. K. (1999) Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema. In: *Proceedings of Machine Translation Summit VII*, Singapore. pp. 244–249.

Al-Adhaileh, M. H. and Tang, E. K. (2002) Synchronous Structured String-Tree Correspondence (S-SSTC). In: *Proceedings of the 20th IASTED International Conference on Applied Informatics (AI 2002)*, Innsbruck, Austria. pp. 270–275.

Al-Adhaileh, M. H., Tang, E. K. and Zaharin, Y. (2002) A Synchronization Structure of SSTC and Its Applications in Machine Translation. In: *Proceedings of COLING 2002 Post-Conference Workshop on Machine Translation in Asia*. Taipei, Taiwan.

Allmuallim, I., Akiba, Y., Yamazaki, T., Yokoo, A. and Kaneda, S. (1994) Two Methods for Learning ALT-J/E Translation Rules from Examples and a Semantic Hierarchy. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan. Association for Computational Linguistics, pp. 57–63.

Asanoma, N. (2001) Alignment of Ontologies: WordNet and Goi-Taikei. In: *NAACL WordNet and Other Lexical Resources*. Pittsburgh, pp. 89–94.

Audibert, L. (2004) Word Sense Disambiguation Criteria: A Systematic Study. In: (col, 2004), pp. 910–916.

Baldwin, T., Okazaki, A., Tokunaga, T. and Tanaka, H. (2001) The Japanese Translation Task: Lexical and Structural Perspectives. In: *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. pp. 55–58.

Basili, R., Rocca, M. D. and Pazienza, M. T. (1997) Towards a Bootstrapping Framework for Corpus Semantic Tagging. In: *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., USA. Association for Computational Linguistisc, pp. 66–73.

Boitet, C. and Zaharin, Y. (1988) Representation Trees and String-Tree Correspondences. In: *Proceedings of the 12th International Conference on Computational Linguistics (COLING88)*, Budapest, Hungary. Budapest, Hungary: Association for Computational Linguistics, pp. 59–64.

Bond, F. (2001) *Determiners and Number in English, contrasted with Japanese, as exemplified in Machine Translation.* Ph.D. thesis, The Centre for Language Teaching and Research, University of Queensland, Brisbane, Australia.

Bond, F., Nichols, E., Fujita, S. and Tanaka, T. (2004) Acquiring an Ontology for a Fundamental Vocabulary. In: (col, 2004), pp. 1319–1325.

Bond, F., Sulong, R. B., Yamazaki, T. and Ogura, K. (2001) Design and Construction of a Machine-Tractable Japanese-Malay Dictionary. In: B. Maegaard, (ed.) *MT Summit VIII.* Santiago de Compostela, Spain, pp. 53–58.

Breen, J. (2003) *XJDIC*. [Software application]. Download available from World Wide Web: `http://www.csse.monash.edu.au/~jwb/xjdic/`.

Breen, J. (2004) JMdict: a Japanese-Multilingual Dictionary. In: (col, 2004), pp. 65–72.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D. and Lai, J. C. (1992) Class-based n-gram Models of Natural Language. *Computational Linguistics*, **18**(4), pp. 467–479.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D. and Mercer, R. L. (1991) Word-sense disambiguation using statistical methods. In: *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, Berkeley, California. Morristown, NJ, USA: Association for Computational Linguistics, pp. 264–270.

Buitelaar, P. (2001) The SENSEVAL-2 Panel on Domains, Topics and Senses. In: *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. pp. 49–52.

Cabezas, C. and Resnik, P. (2005) Using WSD Techniques for Lexical Selection in Statistical Machine Translation. *Tech. Rep. CS-TR-4736/LAMP-TR-124/UMIACS-TR-2005-42*, Institute for Advanced Computer Studies, Language and Media Processing Laboratory, Computational Linguistics and Information Processing Laboratory, University of Maryland, College Park, Maryland, USA.

Cabezas, C., Resnik, P. and Stevens, J. (2001) Supervised Sense Tagging using Support Vector Machines. In: J. Preiss and D. Yarowsky, (eds.) *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. Association for Computational Linguistics, pp. 59–62.

Chao, G. and Dyer, M. G. (2001) Probabilistic Network Models for Word Sense Disambiguation. In: J. Preiss and D. Yarowsky, (eds.) *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. Association for Computational Linguistics, pp. 63–66.

Ciaramita, M. and Johnson, M. (2000) Explaining Away Ambiguity: Learning Verb Selectional Preference with Bayesian Networks. In: *Proceedings of the 18th Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany. Morristown, NJ, USA: Association for Computational Linguistics, pp. 187–193.

Cowie, J., Guthrie, J. and Guthrie, L. (1992) Lexical Disambiguation Using Simulated Annealing. In: *Proceedings of the 14th International Conference on Computational Linguistics (COLING92)*, Nantes, France. Morristown, NJ, USA: Association for Computational Linguistics, pp. 359–365.

Dagan, I. and Itai, A. (1994) Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, **20**(4), pp. 563–596.

Fung, P. and Lo, Y. Y. (1998) An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL 98)*, Montreal, Quebec, Canada. Association for Computational Linguistics, pp. 414–420.

Gale, W. A., Church, K. W. and Yarowsky, D. (1992) A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, **26**, pp. 415–439.

Gaume, B., Hathout, N. and Muller, P. (2004) Word Sense Disambiguation using a Dictionary for Sense Similarity Measure. In: (col, 2004), pp. 1194–1200.

Hearst, M. A. (1991) Noun homograph disambiguation using local context in large corpora. In: *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary*. Oxford, UK, pp. 1–22.

Hirst, G. and St-Onge, D. (1998) Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In: C. Fellbaum, (ed.) *WordNet: An Electronic Lexical Database*, pp. 305–332. MIT Press.

Hutchins, W. J. (1986) *Machine Translation: Past, Present and Future*. Chichester: Ellis Horwood.

Hutchins, W. J. and Somers, H. L. (1992) *An Introduction to Machine Translation*. London: Academic Press.

Ide, N. (1999) Parallel Translations as Sense Discriminators. In: *Proceedings of SIGLEX99: Standardizing Lexical Resources (ACL99 Workshop)*, College Park, Maryland, USA. Association for Computational Linguistics, pp. 52–61.

Ide, N., Erjavec, T. and Tufiş, D. (2002) Sense Discrimination with Parallel Corpora. In: *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, USA. Association for Computational Linguistics, pp. 54–60.

Ide, N. and Véronis, J. (1998) Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, **24**(1), pp. 1–41.

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y. and Hayashi, Y. (1999) *GoiTaikei – A Japanese Lexicon CDROM*. Tokyo, Japan: Iwanami Shoten. [CDROM].

Ikehara, S., Shirai, S., Yokoo, A. and Nakaiwa, H. (1991) Toward an MT system without pre-editing – effects of new methods in **ALT-J/E**. In: *Proceedings of 3rd Machine Translation Summit (MT Summit III)*, Washington DC. pp. 101–106.

Jiang, J. J. and Conrath, D. W. (1997) Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan. pp. 19–33.

Kaji, H. and Morimoto, Y. (2002) Unsupervised Word Sense Disambiguation Using Bilingual Comparable Corpora. In: *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan. Morristown, NJ, USA: Association for Computational Linguistics, pp. 1–7.

Kawahara, D. and Kurohashi, S. (2004) Improving Japanese Zero Pronoun Resolution by Global Word Sense Disambiguation. In: (col, 2004), pp. 343–349.

Kilgariff, A. (2001) English Lexical Sample Task Description. In: *Proceedings of the 2nd International Wordshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. New Brunswick, NJ, USA: The Association for Computational Linguistics, pp. 17–20.

Kim, Y.-S., Chang, J.-H. and Zhang, B.-T. (2002) A Comparative Evaluation of Data-driven Models in Translation Selection of Machine Translation. In: *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan. Morristown, NJ, USA: Association for Computational Linguistics, pp. 1–7.

Lafourcade, M. (2001) Lexical Sorting and Lexical Transfer by Conceptual Vectors. In: *Proceedings of the 1st International Workshop on Multimedia Annotation (MMA2001)*. Tokyo, Japan.

Lafourcade, M. (2002) Automatically Populating Acception Lexical Database through Bilingual Dictionaries and Conceptual Vectors. In: *Proceedings of PAPILLON 2002 Workshop*. Tokyo, Japan.

Lafourcade, M. (2003) Conceptual Vectors and Fuzzy Templates for Discriminating Hyperonymy (`is-a`) and Meronymy (`part-of`) Relations. In: *Proceedings of 2nd International Workshop on Managing Specialization/Generalization HIerarchies (MASPEGHI 2003)*. Montréal, Québec, Canada, pp. 19–29.

Lafourcade, M. (2004) Natural Language Processing: Semantics, Conceptual Vectors, Learning, and Ant Algorithms. Oral presentation at Computer Aided Translation Unit, School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.

Lafourcade, M. (2005) Conceptual Vectors Functions and Operations. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.lirmm.fr/~lafourcade/SERVICES/semvec-docs/semvec-docs.html`.

Lafourcade, M. and Boitet, C. (2002) UNL Lexical Selection with Conceptual Vectors. In: *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*. Las Palmas, Canary Island, Spain.

Lafourcade, M., Rodrigo, F. and Schwab, D. (2004) Low Cost Automated Conceptual Vector Generation from Mono and Bilingual Ressources. In: *Proceedings of PAPILLON 2004 Workshop*, Grenoble, France. GETA-CLIPS-IMAG Study Group for Machine Translation, Grenoble, France: GETA-CLIPS-IMAG Study Group for Machine Translation.

LDOCE (2003) *Longman Dictionary of Contemporary English*. Longman ESL.

Leacock, C. and Chodorow, M. (1998) Combining Local Context and WordNet Similarity for Word Sense Identification. In: C. Felbaum, (ed.) *WordNet: An Electronic Lexical Database*, pp. 265–283. Cambridge, Massachusetts, USA: MIT Press.

Leacock, C., Miller, G. A. and Chodorow, M. (1998) Using corpus statistics and WordNet relations for sense identification. *Comput. Linguist.*, **24**(1), pp. 147–165.

Leacock, C., Towell, G. and Voorhees, E. (1993) Corpus-based Statistical Sense Resolution. In: *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, New Jersey, USA. San Francisco: Morgan Kauffman Publishers, pp. 260–265.

Lee, H. A. and Kim, G. C. (2002) Translation Selection through Source Word Sense Disambiguation and Target Word Selection. In: *Proceedings of the 17th International Conference on Computational Linguistics (COLING 2002)*. Taipei, Taiwan.

Lee, Y. K., Ng, H. T. and Chia, T. K. (2004) Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. In: R. Mihalcea and P. Edmonds, (eds.) *Proceedings of 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*. Barcelona, Spain: Association for Computational Linguistics, pp. 137–140.

Legrand, S., Tyrväinen, P. and Saarikoski, H. (2003) Bridging the Word Disambiguation Gap with the Help of OWL and Semantic Web Ontologies. In: *Proceedings of Workshop on Ontologies and Information Extraction, EROLAN 2003: the Semantic Web and Language Technology*, Bucharest, Romania.

Lesk, M. (1986) Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In: *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC'86)*, Toronto, Ontario, Canada. New York, NY, USA: ACM Press, pp. 24–26.

Levow, G.-A. (1997) Corpus-based Techniques for Word Sense Disambiguation. *Tech. Rep. AIM-1637*, MIT AI Lab, Cambridge, Massachussetts, USA.

Li, H. and Li, C. (2004) Word Translation Disambiguation Using Bilingual Bootstrapping. *Computational Linguistics*, **30**(1), pp. 1–22.

Lim, B. T. (2003) *Semantic-Primitive-Based Lexical Consultation System.* Master's thesis, School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.

Lim, B. T., Tang, E. K. and Guo, C. M. (2002) Building a Semantic-Primitive-Based Lexical Consultation System. In: *Proceedings of Pre-COLING 2002 Seminar on Linguistic Meaning Representation and Their Applications over the World Wide Web.* Penang, Malaysia.

Lin, D. (1998) An Information-Theoretic Definition of Similarity. In: *Proceedings of the International Conference on Machine Learning*, Madison.

Magnini, B. and Cavaglià, G. (2000) Integrating Subject Field Codes into WordNet. In: M. Gavrilidou, G. Crayannis, S. Markantonatu, S. Piperidis and G. Stainhaouer, (eds.) *Proceedings of the 2nd International Conference on Langauge Resources and Evaluation (LREC-2000)*, Athens, Greece. pp. 1413–1418.

Magnini, B., Strapparava, C., Pezzulo, G. and Gliozzo, A. (2001) Using Domain Information for Word Sense Disambiguation. In: J. Preiss and D. Yarowsky, (eds.) *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. The Association for Computational Linguistics, pp. 111–114.

Magnini, B., Strapparava, C., Pezzulo, G. and Gliozzo, A. (2002) Comparing Ontology-Based and Corpus-Based Domain Annotation in WordNet. In: *Proceedings of First International WordNet Conference*, Mysore, India. pp. 146–154.

Mangeot, M. and Sérasset, G. (2005) The *Papillon* Project. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.papillon-dictionary.org/`.

McCarthy, D., Caroll, J. and Preiss, J. (2001) Disambiguating Noun and Verb Senses Using Automatically Acquired Selectional Preferences. In: J. Press and D. Yarowsky, (eds.) *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2)*, Toulouse, France. Association for Computational Linguistics, pp. 119–122.

McRoy, S. W. (1992) Using Multiple Knowledge Sources for Word Sense Disambiguation. *Computational Linguistics*, **18**(1), pp. 1–30.

McTait, K. (2003) *Translation Patterns, Linguistic Knowledge and Complexity in an Approach to EBMT*, vol. 21 of *Text, Speech and Language Technology Series*, pp. 307–338. Dordrecht, Amsterdam, Netherlands: Kluwer Academic Publishers.

Meyer, C. B. (2000) An Introduction to Offender Profiling. *The Basel University Law Review*, **1**, pp. 15–20. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.criminalprofiling.ch/introduction.html`.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. (1990) Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography (special issue)*, **3**(4), pp. 235–312.

Nagao, M. (1984) A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In: A. Elithorn and R. Banerji, (eds.) *Artificial and Human Intelligence*, pp. 173–180. Amsterdam, Netherlands: Elsevier Science Publishers.

Navigli, R. and Velardi, P. (2005) Structural Semantic Interconnections: A Knowledge-based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(7), pp. 1075–1086.

Ng, H. T. and Lee, H. B. (1996) Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach. In: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, Santa Cruz, California. Morristown, NJ, USA: Association for Computational Linguistics, pp. 40–47.

Ng, H. T., Wang, B. and Chan, Y. S. (2003) Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. In: E. Hinrichs and D. Roth, (eds.) *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. pp. 455–462.

Niles, I. and Pease, A. (2001) Towards a Standard Upper Ontology. In: C. Welty and B. Smith, (eds.) *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine.

Ogura, K., Bond, F. and Ooyama, Y. (1999) **ALT-J/M**: A prototype Japanese-to-Malay Translation System. In: *Proceedings of Machine Translation Summit VII (MT Summit VII)*, Singapore. pp. 444–448.

O'Hara, T., Bruce, R., Donner, J. and Wiebe, J. (2004) Class-based Collocations for Word Sense Disambiguation. In: R. Mihalcea and P. Edmonds, (eds.) *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3)*. Barcelona, Spain: Association for Computational Linguistics, pp. 199–202.

Okuda, M., Okuda, D. and Mirek, D. (1999) *The Star Trek Encyclopedia*. New York: Pocket Books.

OpenCyc (2005) OpenCyc. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.opencyc.org/`.

Pease, A. (2005) Suggested Upper Merged Ontology (SUMO). [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.ontologyportal.org/`.

Pedersen, T., Banerjee, S. and Patwardhan, S. (2005) Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. *Tech. Rep. UMSI 2005/25*, University of Minnesota Supercomputing Institute.

Pereira, F., Tishby, N. and Lee, L. (1993) Distributional Clustering of English Words. In: *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, Columbus, Ohio. Morristown, NJ, USA: Association for Computational Linguistics, pp. 183–190.

Prince, V. and Lafourcade, M. (2003) Mixing Semantic Networks and Conceptual Vectors: the Case of Hyperonymy. In: *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI-2003)*. London, United Kingdom, pp. 121–128.

Quah, C. K., Bond, F. and Yamazaki, T. (2001) Design and Construction of a Machine-Tractable Malay-English Lexicon. In: *Proceedings of the 2001 Asian Association for Lexicography (ASIALEX) Biennial Conference*. Seoul,Korea, pp. 200–205.

Rada, R., Mili, H., Bicknell, E. and Blettner, M. (1989) Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, **19**(1), pp. 17–30.

Reed, S. L. and Lenat, D. B. (2002) Mapping Ontologies into Cyc. In: *Proceedings of AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, Edmonton, Canada.

Resnik, P. (1995a) Disambiguating Noun Groupings with Respect to WordNet Senses. In: D. Yarowsky and D. Church, (eds.) *Proceedings of 3rd Workshop on Very Large Corpora*, Cambridge, Massachusetts, USA. Association for Computational Linguistics, pp. 54–68.

Resnik, P. (1995b) Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada. pp. 448–453.

Resnik, P. (1997) Selectional Preference and Sense Disambiguation. In: *Proceedings of the Workshop on Taggin Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., USA.

Resnik, P. (1999) Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, **11**, pp. 95–130.

Resnik, P. and Diab, M. (2000) Measuring Verb Similarity. In: *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society (CogSci2000)*, Philadelphia, Pennsylvania, USA.

Resnik, P. and Yarowsky, D. (1997) A Perspective on Word Sense Disambiguation Methods and Their Evaluation. In: *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, Washington, D.C., USA. Association for Computational Linguistisc, pp. 79–86.

Roget, P. (2002) *Roget's Thesaurus of English Words and Phrases*. London, United Kingdom: Penguin Books.

Sadler, V. and Vendelmans, R. (1990) Pilot Implementation of a Bilingual Knowledge Bank. In: *Proceedings of the 13th conference on Computational Linguistics (COLING 1990)*. Helsinki, Finland: Association for Computational Linguistics, pp. 449–451.

Salton, G., Wong, A. and Yang, C. S. (1975) A Vector Space Model for Automatic Indexing. *Commununications of the ACM*, **18**(11), pp. 613–620.

Sanderson, M. (2000) Retrieving with Good Sense. *Information Retrieval*, **2**(1), pp. 49–69.

Sato, S. and Nagao, M. (1990) Toward Memory-based Translation. In: *Proceedings of the 13th Conference on Computational Linguistics (COLING 1990)*, Helsinki, Finland. Morristown, NJ, USA: Association for Computational Linguistics, pp. 247–252.

Schütze, H. (1992) Dimensions of Meaning. In: *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Minneapolis, Minnesota, USA. Los Alamitos, California, USA: IEEE Computer Society Press, pp. 787–796.

Schütze, H. (1998) Automatic Word Sense Discrimination. *Computational Linguistics*, **24**(1), pp. 97–123.

Schwab, D. and Lafourcade, M. (2003) Hardening of Acception Links Through Vectorized Lexical Functions. In: *Proceedings of Papillon 2003 Workshop*.

SENSEVAL (2005) SENSEVAL: Evaluation exercises for Word Sense Disambiguation. ACL-SIGLEX. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://www.senseval.org/`.

Sérasset, G. and Boitet, C. (2000) On UNL as the Future "html of the linguistic content" and the Reuse of Existing NLP Components in UNL-Related Applications with the Example of a UNL-French Deconverter. In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING 2000)*. Saarbr ueken.

Ševčenko, M. (2003) Online Presentation of an Upper Ontology. In: *Proceedings of Znalosti 2003*, Ostrava, Czech Republic.

Ševčenko, M. (2004) The KSMSA Project. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://virtual.cvut.cz:8080/ksmsaWeb/browser/title`.

Shirai, K. and Yagi, T. (2004) Learning a Robust Word Sense Disambiguation Model using Hypernyms in Definition Sentences. In: (col, 2004), pp. 917–923.

Somers, H. L. (1999) Review Article: Example-based Machine Translation. *Machine Translation*, **14**(2), pp. 113–157.

Stevenson, M. and Wilks, Y. (2001) The Interaction of Knowledge Sources in Word Sense Disambiguation. *Comput. Linguist.*, **27**(3), pp. 321–349.

Sumita, E. and Iida, H. (1991) Experiments and Prospects of Example-based Machine Translation. In: *Proceedings of the 29th Conference on Association for Computational Linguistics*, Berkeley, California. pp. 185–192.

Sussna, M. (1993) Word Sense Disambiguation for Free-Text Indexing using a Massive Semantic Network. In: *Proceedings of the 2nd International Conference on Information and Knowledge Management*, Washington, D.C., USA. New York, NY, USA: ACM Press, pp. 67–74.

Tang, E. K. and Al-Adhaileh, M. H. (2001) Converting a Bilingual Dictionary into a Bilingual Knowledge Bank Based on the Synchronous SSTC Annotation Schema. In: *Proceedings of the 8th Machine Translation Summit (MT SUMMIT VIII)*, Santiago, Spain.

Tang, E. K. and Zaharin, Y. (1995) Handling Crossed Dependencies with the STCG. In: *Proceedings of NLPRS'95*, Seoul, Korea.

Tufiş, D., Ion, R. and Ide, N. (2004) Fine-Grained Word Sense Disambiguation based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets. In: (col, 2004), pp. 1312–1318.

Turvey, B. E. (2001) *Criminal Profiling: An Introduction to Behavioral Evidence Analysis*. Elsevier Academic Press, 2nd edn.

Vauquois, B. (1968) A Survey of Formal Grammars ad Algorithms for Recognition and Transformation in Machine Translation. In: C. Boitet, (ed.) *Bernard Vauquois et la TAO: Vingt-cinq Ans de Traduction Automatique – Analectes*, pp. 201–213. Grenoble, France: Association Champollion. (Collection printed in 1988).

Viegas, E., Mahesh, K., Nirenburg, S. and Beale, S. (1999) Predicative Forms in Natural Language and in Lexical Knowledge Bases. In: P. Saint-Dizier, (ed.) *Predicative Forms in Natural Language and Lexical Knowledge Bases*, Text, Speec and Language Technology Series, pp. 171–203. Dordrecht, the Netherlands: Kluwer Academic Publishers.

Wilks, Y., Fass, D., Guo, C.-M., McDonald, J., Plate, T. and Slator, B. (1993) Providing Machine Tractable Dictionary Tools. In: J. Pustejovsky, (ed.) *Semantics and the Lexicon*, pp. 341–401. Dordrecht, the Netherlands: Kluwer Academic Publishers.

Wilks, Y. and Stevenson, M. (1997) Sense Tagging: Semantic Tagging with a Lexicon. In: *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How? (SIGLEX97)*, Washington DC, USA. pp. 47–51.

Wilks, Y. and Stevenson, M. (1998) Word Sense Disambiguation using Optimised Combinations of Knowledges Sources. In: *Proceedings of the 17th International Conference on Computational Linguistics*, Montreal, Canada. pp. 1398–1402.

Wong, F., Hu, D. C., Mao, Y. H. and Dong, M. C. (2004) A Flexible Example Annotation Schema: Translation Corresponding Tree Representation. In: (col, 2004), pp. 1079–1085.

WordNet (2005) *WordNet*: A Lexical Database for the English Language. Cognitive Science Laboratory, Princeton University, NJ, USA. [Online]. [Accessed 1st June 2007]. Available from World Wide Web: `http://wordnet.princeton.edu/`.

Wu, Z. and Palmer, M. (1994) Verbs Semantics and Lexical Selection. In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics (COLING-94)*, Las Cruces, New Mexico. Morristown, NJ, USA: Association for Computational Linguistics, pp. 133–138.

Yarowsky, D. (1992) Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In: *Proceedings of the 14th Conference on Computational Linguistics (COLING 1992)*, Nantes, France. Morristown, NJ, USA: Association for Computational Linguistics, pp. 454–460.

Yarowsky, D. (1993) One Sense per Collocation. In: *Proceedings of ARPA Human Language Technology Workshop*, Plainsboro, New Jersey, USA. San Francisco, California, USA: Morgan Kauffman Publishers, pp. 266–271.

Yarowsky, D. (1995) Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: *Proceedings of the 33rd Annual Meeting of the Association for Computational*

Linguistics, Cambridge, Massachussetts, USA. Association for Computational Linguistics, pp. 189–196.

Ye, H. H. (forthcoming) *Indexing a Bilingual Knowledge Bank for Example-Based Machine Translation (EBMT) based on the Synchronous SSTC Annotation Schema*. Master's thesis, School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.

Zhou, M., Ding, Y. and Huang, C. (2001) Improviging Translation Selection with a New Translation Model Trained by Independednt Monolingual Corpora. *Computational Linguistics and Chinese language Processing*, **6**(1), pp. 1–26.

PUBLICATION LIST

Lim, L. T. and Tang, E. K. (2004) Building an Ontology-Based Multilingual Lexicon for Word Sense Disambiguation in Machine Translation. In: *Proceedings of the 5th Workshop on Multilingual Lexical Databases (Papillon 2004)*, Grenoble, France. Grenoble, France: GETA-CLIPS-IMAG.

APPENDICES

APPENDIX A

MAPPING OF ARRAY INDICES FOR DISTANCES BETWEEN CONCEPT PAIRS

It is desirable to have a fast method for retrieving the distance between pairs of concepts in the chosen concept hierarchy. The method chosen in this research is to store pre-computed conceptual distances in a text file, the contents of which will be read into memory as a single-dimensional array at runtime.

To further minimise the running time and amount of memory required, the array is indexed such that the distance between two concepts is the element whose subscript in the (0-based[1]) array is a function of the integer IDs of the concepts. For this purpose, the D concepts are re-enumerated starting from 0. We use the example concept hierarchy with five concepts, shown in Figure A.1, for demonstration.

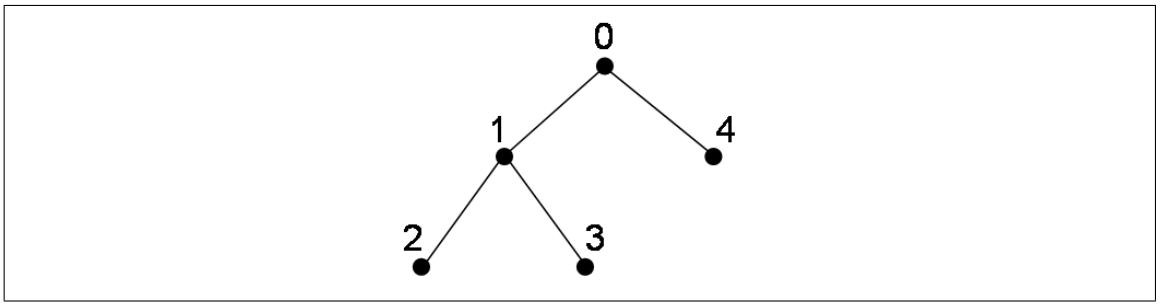


Figure A.1: Small Concept Hierarchy with Five Concepts

Given D concepts in a concept hierarchy C, there are a total of D^2 concept pairs whose distances need to be taken into account.

[1] i.e. numbered from 0

However, since $dist(c_i, c_j) = dist(c_j, c_i)$ and $\text{dist}(c_i, c_i) = 0$, only $\frac{D}{2}(D-1)$ elements need to be stored in the memory during runtime. In the case of the example hierarchy shown in Figure A.1, where $D = 5$, 10 elements are needed, as indicated by the white cells in Table A.1.

The text file `gt_distance` contains the distances between pairs of concepts, the contents of which is shown below for our example hierarchy:

```
-1  keyX  keyY  distance
 0   0     1     1
 1   0     2     2
 2   0     3     2
 3   0     4     1
 4   1     2     1
 5   1     3     1
 6   1     4     2
 7   2     3     2
 8   2     4     3
 9   3     4     3
```

The first and second column on each row indicate the integer ID of each pair of concepts, while the third column contains the distance (i.e. path length) between them. Only values in the third column will be read into a 0-based array $\mathcal{A}$ of length $\frac{1}{2}D(D-1)$ at run-time. Given the integer IDs of two concepts, x and y, the distance between these two concepts can then be determined from x, y and $\mathcal{A}$ as given by:

$$\text{dist}(x, y) = \left\{ \begin{array}{ll} 0 & \text{if } x = y \\ \mathcal{A}_{\text{key}(x,y)} & \text{if } x < y \\ \mathcal{A}_{\text{key}(y,x)} & \text{otherwise} \end{array} \right\} \quad \begin{array}{l} x, y, \text{key}(x, y) \in \mathbb{Z}^+, \\ 0 \leqslant x, y < D, \\ 0 \leqslant \text{key}(x, y) < \frac{1}{2}D(D-1) \end{array} \tag{A.1}$$

where $\mathcal{A}_i$ denotes the ith value in $\mathcal{A}$ and $\text{key}(x, y)$ is a mapping function described below.

Table A.1: Mapping Conceptual Distances to a Single-Dimensional Array

$y \rightarrow$ $x \downarrow$	**0**	**1**	**2**	**3**	**4**
0	0	1 $_0$	2 $_1$	2 $_2$	1 $_3$
1	1	0	1 $_4$	1 $_5$	2 $_6$
2	2	1	0	2 $_7$	3 $_8$
3	2	1	2	0	3 $_9$
4	1	2	3	3	0

Table A.1 shows the distance between two concepts with IDs x and y respectively as an intersection between row x and column y, where $0 \leqslant x < y < D$. The subscript in each white box is $\text{key}(x, y)$, which evaluates to that particular element's position in $\mathcal{A}$. For example, $\text{dist}(1, 4) = \mathcal{A}_{\text{key}(1,4)} = \mathcal{A}_6 = 2$.

With the aid of Table A.1, we can derive $\text{key}(x, y)$ as follows:

$$\begin{aligned} \text{key}(x, y) = {} & \text{total number of elements in previous rows up till row } (x-1) \quad (= N_1) \\ & + \text{total number of elements in row } x \text{ up till column } (y-1) \quad (= N_2) \end{aligned} \tag{A.2}$$

By considering N_1 as the sum of an arithmetic progression with x terms, where the first term is the number of elements on row 0 $(= D - 1)$ and common difference -1, we have

$$\begin{aligned} N_1 &= \frac{1}{2}x \times [2(D-1) + (-1)(x-1)] \\ &= \frac{x}{2}(2D - x - 1) \end{aligned} \tag{A.3}$$

From Table A.1, N_2 is easily seen to be

$$N_2 = y - x - 1 \tag{A.4}$$

Finally, from (A.2), (A.3) and (A.4), we have

$$\begin{aligned} \text{key}(x, y) &= \frac{x}{2}(2D - x - 1) + (y - x - 1) \\ &= \frac{x}{2}(2D - x - 3) + y - 1 \end{aligned} \tag{A.5}$$

As an example, to get the distance between concepts 2 and 4 in the example hierarchy, we have

$$\begin{aligned} \text{key}(2, 4) &= \frac{1}{2} \times 2 \times (2 \times 5 - 2 - 3) + 4 + 1 \qquad (D = 5) \\ &= 8 \\ \text{dist}(2, 4) &= \mathcal{A}_{\text{key}(2,4)} \\ &= \mathcal{A}_8 \\ &= 3 \end{aligned}$$

Using this mapping procedure for the 2,705 concepts in GT, only 3,657,160 values need to be stored in memory, instead of 7,312,025($= 2705^2$) values. This cuts down on both running time and memory requirements greatly.

APPENDIX B

GT COMMON NOUN HIERARCHY

The following (unofficial) rough English translations from the original Japanese *GoiTaikei – A Japanese Lexicon CDROM* (Ikehara *et al.*, 1999) are meant as reference for non-speakers of the Japanese language. In cases of confusion and doubt, (Ikehara *et al.*, 1999) should always be the definitive guide.

Note that concepts 995 to 999 are undefined. Therefore, the total number of concepts in the *GT* common noun hierarchy is 2,705.

Top Level Concepts

1: Noun

- 2: Concrete
  - 3: Agent
    - 4: Person
    - 362: Organization
  - 388: Place
    - 389: Facility
    - 458: Region
    - 468: Nature
  - 533: Object
    - 534: Animate
    - 706: Inanimate
- 1000: Abstract
  - 1001: Abstract thing
    - 1002: Mental State
    - 1154: Action
  - 1235: Event
    - 1236: Human Activity
    - 2054: Phenomenon
    - 2304: Natural phenomenon
  - 2422: Abstract Relationship
    - 2423: Existence
    - 2432: Categorisation System
    - 2443: Relation
    - 2483: Characteristic
    - 2507: State
    - 2564: Form
    - 2585: Numerical
    - 2610: Location
    - 2670: Time

Lower Level Concepts

4: Person
5: Human
6: Form of address
7: First-person
8: First-person Singular
9: First-person Singular Male
10: First-person Singular Female
11: First-person Plural
12: First-person Plural Male
13: First-person Plural Female
14: Second-person
15: Second-person Singular
16: Second-person Singular Male
17: Second-person Singular Female
18: Second-person Plural
19: Second-person Plural Male
20: Second-person Plural Female
21: Third-person
22: Third-person Singular
23: Third-person Singular Male
24: Third-person Singular Female
25: Third-person Plural
26: Third-person Plural Male
27: Third-person Plural Female
28: Indefinite-person
29: Indefinite Singular
30: Indefinite Plural

31: Oneself and others
32: Oneself
33: Others
34: Each
35: Public and Private
36: Public
37: Private
38: Affix human
39: Affix human Singular
40: Affix human Singular Male
41: Affix human Singular Female
42: Affix human Plural
43: Honorific title
44: Honorific title Male
45: Honorific title Female
46: Biological feature
47: Man Woman
48: Man
49: Woman
50: Old Young
51: Child
52: Boy
53: Girl
54: Juvenile
55: Boy
56: Girl
57: Youth
58: Youth Male
59: Youth Female
60: Adult
61: Adult male
62: Adult female
63: Old person
64: Old people male
65: Old people female
66: Body circumstance
67: Figure
68: Sick Injured
69: Sick person
70: Injured person
71: Deceased
72: Kinship relation
73: Family
74: Married couple
75: Husband
76: Wife
77: Parent Grandparent Ancestral
78: Parent
79: Father
80: Mother
81: Grandparent
82: Grandfather
83: Grandmother
84: Ancestral
85: Child Grandchild Descendant
86: Child
87: Son
88: Daughter
89: Grandchild
90: Grandson
91: Granddaughter
92: Descendant
93: Sibling
94: Sibling senior
95: Older brother
96: Older sister
97: Sibling junior
98: Younger brother
99: Younger sister
100: Kindred
101: Uncle Aunt
102: Uncle
103: Aunt
104: Nephew niece
105: Nephew
106: Niece
107: Cousin
108: Cousin Male
109: Cousin Female
110: Kindred
111: Personal relation
112: Society relation
113: Companion Member Partner
114: Companion Member
115: Companion
116: Comrade
117: Follower
118: Companion
119: Member
120: Outcast Nuisance person
121: Partner
122: Partner
123: Enemy Ally
124: Friend Intimate person
125: Friend
126: Acquaintance
127: Lover
128: Lover male
129: Lover female
130: Host Guest
131: Host
132: Host male
133: Host female
134: Guest
135: Visitor
136: Customer
137: Relative position
138: Teacher Student
139: Teacher
140: Student
141: Superior Subordinate
142: Superior
143: Subordinate
144: Senior Junior
145: Senior
146: Junior
147: Master Servant
148: Master
149: Servant
150: Social group
151: Race Ethnic group
152: Race
153: Ethnic group
154: National
155: Fellow countryman
156: Foreigner
157: Residence circumstance
158: Inhabitant
159: Immigrant Refugee
160: Traveler
161: Social class
162: Ruler Retainer
163: Monarch
164: Lord
165: Retainer
166: Authority Populace
167: Authority
168: Populace
169: Status
170: Nobility
171: Warrior
172: Commoner
173: Menial
174: Capital and labor
175: Capitalist
176: Labourer
177: Wealth and poverty
178: Wealthy
179: Poor
180: Ability inclination
181: Ability
182: Great men ordinary person
183: Great man
184: Ordinary person
185: Founder Follower
186: Founder
187: Follower
188: Wise Fool
189: Wise man
190: Fool
191: Expert Beginner
192: Expert
193: Beginner
194: Hero Coward
195: Hero
196: Coward
197: Inclination
198: Hardworker Lazy person
199: Hard worker
200: Lazy person
201: Eccentric Ordinary
202: Eccentric
203: Hobbyist
204: Enthusiast
205: Gourmet
206: Lustful person
207: Eccentric
208: Eccentric
209: Ideologist
210: Lunatic
211: Ordinary man
212: Good man Bad man
213: Good man
214: Bad guy
215: Bad guy
216: Ruffian
217: Criminal
218: Prisoner
219: Quasi-human

220: Deity Buddha
221: Spirit
222: Demon Monster
223: Occupation position role
224: Occupation
225: Professional occupation
226: Professional job
227: Professional medical care
228: Doctor
229: Pharmacist
230: Nurse
231: Professional technology
232: Professional
233: Attorney
234: Guide
235: Consultant
236: Teacher Student
237: Teacher
238: Student
239: Scholar Researcher
240: Artist
241: Writer Poet
242: Artist Calligrapher
243: Musician
244: Artist
245: Journalist
246: Talent contestant
247: Musician
248: Singer
249: Performer
250: Entertainer
251: Contestant
252: Talent Contestant
253: Religion person
254: Monk
255: Priest
256: Clergyman
257: Pilgrim
258: Fortuneteller
259: Administrative Managerial
260: Politician
261: Government official
262: Administrator
263: Judicial official
264: Judge
265: Public prosecutor
266: Diplomat
267: Government official
268: Entreprenuer
269: Administrative position
270: Others
271: Office work job
272: Sales job
273: Merchant
274: Salesmen
275: Agriculture Forestry Fishery
276: Farmer
277: Fisherman
278: Hunter
279: Woodcutter
280: Breeder
281: Operator
282: Operator factory
283: Operator construction mine
284: Operator
285: Technical skill job
286: Craftsman
287: Haircut teacher hairdresser
288: Transport communication
289: Transport industry
290: Carrier
291: Crew
292: Driver
293: Crew
294: Transport industry
295: Communication industry
296: Postal industry
297: Telecomunications industry
298: Security job
299: Military
300: Police
301: Fireman
302: Guard
303: Service job
304: Servant
305: Servant male
306: Servant Female
307: Customer Relations
308: Boy
309: Waitress
310: Geisha
311: Free outlaw occupation
312: Beggar
313: Thief
314: Mafia
315: Prostitute
316: Spy
317: Free outlaw occupation
318: Position
319: King Lord
320: King
321: Nobility
322: Cabinet minister
323: Chief
324: Assistant director
325: Board of director
326: Staff
327: Officer rank
328: Officer
329: Private soldier
330: Contestant class
331: Designation
332: Position
333: Role
334: Leader
335: Management
336: Leader
337: Person in charge
338: Relation
339: Official
340: Assistant
341: Interested party
342: Attorney
343: Plaintiff defendant
344: Human
345: Messenger detective
346: Messenger
347: Detective
348: Ownership related
349: Owner
350: Ownership related
351: Protagonist
352: Reading writing
353: Writer
354: Reader
355: Reviewer
356: Performer Audience
357: Performer
358: Audience
359: Player Referee
360: Player
361: Referee

362: Organization

363: Institution
364: Administrative
365: Judicial
366: Legislature
367: Public institution
368: Armed force team
369: Armed force
370: Team
371: International agency
372: Association Party and Group
373: Association
374: Enterprise
375: Association
376: Alliance
377: Body
378: Club
379: Party and group
380: Political party
381: Class
382: Faction
383: Group
384: Society
385: Country
386: Domain
387: Family

389: Facility

390: Public facility
391: Government Office
392: Administration office
393: Judicial office
394: Government offices
395: Service facility
396: Post office
397: Communication facility
398: Broadcasting station
399: Electrical facility
400: Water and sewer facility
401: Gas facility
402: Research Observatory
403: Service facilities
404: Hospital
405: School
406: Cultural facility
407: Museum

408: Library
409: Theater
410: Public hall
411: Gymnasium
412: Cultural facilities
413: Getting on and off place
414: Station
415: Port
416: Airport
417: Lines of transport
418: Road
419: Railroad
420: Bridge
421: Tunnel
422: Area facility
423: Garden
424: Zoo Botanical garden
425: Recreational area
426: Stadium
427: Public facilities
428: Work place
429: Office
430: Store
431: Branch office
432: Market
433: Department store
434: Store
435: Restaurant
436: Barber bathhouse
437: Accommodation
438: Store
439: Factory
440: Laboratory
441: Studio
442: Mine
443: Farm
444: Work place
445: Residence facility
446: Palace
447: Residence facility
448: Military establishment
449: Fort
450: Camp
451: Military establishment
452: Religion facility
453: Shrine
454: Temple
455: Church
456: Graveyard
457: Facility

458: Region

459: Region range
460: Region human activity
461: Land
462: Realm
463: Territory
464: Administrative district
465: City
466: Village
467: Hometown

468: Nature

469: Topography
470: Terrain
471: Land
472: Mountain
473: Mountain main
474: Mountain part
475: Mountaintop
476: Mountain side
477: Base
478: Pass
479: Valley
480: Cliff
481: Cave
482: Level ground
483: Plain
484: Basin
485: Plateau
486: Island Cape
487: Island
488: Sandbank
489: Promontory Peninsula
490: Shore
491: Coast
492: River bank
493: Lakeshore
494: Embankment
495: River
496: River
497: Waterway
498: Waterfall
499: Lake Marsh
500: Lake
501: Swamp
502: Pond
503: Puddle
504: Spring Well
505: Spring
506: Well
507: Sea
508: Open sea
509: Inland sea
510: Scenery
511: Physiographic
512: Physiographic natural
513: Forest
514: Grassland
515: Uncultivated land
516: Physiographic artificial
517: Arable land Ranch
518: Rice field
519: Field
520: Ranch
521: Forested land
522: Vacant land
523: Site
524: Physiographic artificial
525: Scene
526: Space
527: Celestial
528: Terrestrial
529: Moon celestial body
530: Solar
531: Star
532: Sky

534: Animate

535: Animal
536: Animal Individual
537: Animal
538: Bird
539: Reptile Amphibian
540: Reptile
541: Amphibian
542: Fish and Shellfish
543: Fish
544: Fish and Shellfish
545: Shellfish
546: Crab Octopus Prawn
547: Sea urchin Jellyfish
548: Insect
549: Insect
550: Insect
551: Animal Individual
552: Animal Part
553: Head
554: Head
555: Face
556: Facial
557: Forehead
558: Cheek
559: Chin
560: Eye
561: Eye Main
562: Eye Part
563: Nose
564: Nose Main
565: Nose Part
566: Mouth
567: Mouth Main
568: Lip
569: Tongue
570: Beak
571: Ear
572: Ear Main
573: Ear Part
574: Neck
575: Throat
576: Nape
577: Body
578: Chest
579: Stomach
580: Waist
581: Shoulder
582: Back
583: Side
584: Rear end
585: Tail
586: Bodies
587: Breast
588: Navel
589: Genital
590: Hand Foot
591: Hand Upper limb
592: Arm
593: Elbow
594: Hand

595: Wrist
596: Palm
597: Back of hand
598: Finger
599: Leg Lower limb
600: Hip Thigh
601: Knee
602: Leg Shin
603: Foot
604: Ankle
605: Sole
606: Top of foot
607: Toe
608: Finger
609: Joint
610: Wing Fin Webbed
611: Wing
612: Fin
613: Webbed etc
614: Internal organ
615: Internal organ
616: Respiratory organ
617: Digestive organ
618: Circulatory organ
619: Urinary organ
620: Genital organ
621: Nervous system
622: Gland
623: Membrane
624: Muscle
625: Skin hair
626: Skin
627: Skin
628: Mole Wart
629: Shell
630: Shell
631: Shell
632: Scale
633: Hair
634: Hair
635: Eyebrow Eyelash
636: Facial hair
637: Body hair
638: Feather
639: Bone tooth nail etc
640: Bone
641: Tooth Gum
642: Tooth
643: Tooth Gum
644: Nail Horn Fang
645: Nail
646: Horn
647: Fang
648: Blood secretion excreta
649: Blood
650: Secretion
651: Sweat
652: Tear
653: Milk
654: Digestive juice Hormone
655: Digestive juice
656: Hormone
657: Gonad liquid
658: Secretion misc
659: Nasal mucus
660: Saliva
661: Eye discharge
662: Pus
663: Dandruff
664: Excreta
665: Excretory thing
666: Feces
667: Urine
668: Fart
669: Vomit
670: Egg
671: Plant
672: Plant Individual
673: Tree and Shrub
674: Fruit tree
675: Tree and Shrub
676: Grass
677: Crop
678: Flowering plant Fieldgrass
679: Water plant
680: Plant Individual
681: Moss Fern
682: Fungus
683: Mold
684: Mushroom
685: Bacteria
686: Plant Part
687: Sprout Seedling
688: Sprout
689: Seedling
690: Root
691: Stalk Stump
692: Stalk
693: Stump
694: Branch leaf
695: Branch
696: Leaf
697: Flower
698: Flower Main
699: Flower Part
700: Fruit Seed ear
701: Fruit
702: Seed
703: Ear
704: Bark Fruit peeling
705: Cell

706: Inanimate

707: Natural thing
708: Matter Part
709: Component
710: Element
711: Atom
712: Matter Main
713: Solid
714: Metal
715: Precious metal
716: Base metal
717: Iron
718: Base metal
719: Acid Alkaline Salt
720: Acid solid
721: Alkaline
722: Salt
723: Mineral
724: Ore
725: Jewel
726: Coal
727: Rock
728: Stone Sand
729: Stone
730: Sand
731: Earth
732: High molecular compound
733: Nutrient
734: Cellulose
735: High molecular compound
736: Dust
737: Dust
738: Smoke
739: Ash
740: Rust
741: Dreg
742: Dirt
743: Poison
744: Ice
745: Solid matter
746: Liquid
747: Water Bubble
748: Water
749: Hot water
750: Waterdrop
751: Bubble
752: Acid liquid
753: Petroleum
754: Mercury
755: Liquid
756: Vapour
757: Air
758: Vapour
759: Vapour
760: Artificial thing
761: Goods
762: Personal effect
763: Commodity
764: Gift
765: Offering
766: Treasure
767: Product
768: Cargo
769: Material
770: Paper
771: Wood
772: Timber
773: Plank
774: Stone
775: Stone
776: Cement
777: Glass
778: Ceramic ware
779: Metal Foil
780: Metal
781: Foil Leaf
782: Fiber Leather

783: Fiber
784: Leather
785: Plastics
786: Plastic
787: Rubber
788: Plastics
789: Oil
790: Oil and Fat
791: Mineral oils
792: Fuel Fertilizer Fodder
793: Fuel
794: Solid fuel
795: Liquid fuel
796: Gaseous fuel
797: Fertilizer
798: Fodder
799: Rubbish
800: Chemical
801: Chemical medical
802: Chemical non medical
803: Agricultural chemical
804: Cosmetics
805: Cosmetics
806: Soap
807: Toothpaste
808: Perfume
809: Paint
810: Dye
811: Adhesive
812: Gun powder
813: Clothing
814: Thread Cloth
815: Thread
816: Cloth
817: Clothing
818: Clothing Main
819: Clothing Main upper body
820: Clothing Main lower body
821: Clothing part
822: Clothing attachment
823: Clothing accessory
824: Glove
825: Sock
826: Band
827: Scarf
828: Necktie
829: Headband
830: Mask
831: Clothing accessory
832: Hat
833: Footwear
834: Ornament
835: Rainwear Bedcloth
836: Rainwear
837: Bedcloth
838: Food
839: Foodstuff
840: Grain
841: Vegetable
842: Fish
843: Meat Egg
844: Dried fish Pickle
845: Tofu Agar
846: Seasoning
847: Cooked Food
848: Rice
849: Noodle
850: Bread
851: Soup
852: Cooked Food
853: Snackfood
854: Fruit
855: Confectionary
856: Beverage Tobacco
857: Beverage
858: Tea
859: Coffee juice
860: Milk
861: Liquor
862: Tobacco
863: Building
864: House
865: House Main
866: House Part
867: House Part place
868: Room
869: Corridor
870: Stairway
871: Veranda
872: Balcony
873: House part place addition
874: Houses part element
875: Roof
876: Ceiling
877: Pillars beam
878: Wall
879: Window
880: Floor
881: Base
882: Houses part element
883: House Attachment
884: Fitting
885: Curtain
886: Sunshade
887: Rug
888: Shelf Stand Platform
889: Garden
890: Gate Fence
891: Gate
892: Fence
893: Tool
894: Household effects Light
895: Furniture
896: Desk
897: Chair
898: Rack
899: Cupboard
900: Cooking stove
901: Air-conditioner
902: Furniture
903: Light
904: Container
905: Bottle Pot Tray
906: Bucket Can Barrel
907: Box
908: Sack
909: Basket Straw bag
910: Table ware
911: Bowl Cup Plate
912: Pot Kettle
913: Chopstick spoon
914: Tableware
915: Home equipment
916: Stationery Toy
917: Stationery
918: Document Publication
919: Document
920: Publication
921: Sports equipment
922: Instrument sound producing
923: Musical instrument
924: Bell Whistle
925: Marker Symbol
926: Sign
927: Badge
928: Monument
929: Flag
930: Token Ticket
931: Token
932: Ticket
933: Lottery
934: Money
935: Seal
936: Pointer
937: Ornament
938: Image Book Picture
939: Ornament
940: Job tool
941: Job tool use
942: Tool
943: Edged tool
944: Farming tool Harness
945: Farming tool
946: Harness
947: Fishing Hunting tool
948: Job tool action
949: Job tool opening and closing
950: Job tool join
951: Job tool action
952: Job tool form
953: Wheel
954: Car
955: Rod
956: Tube framework
957: Wire
958: Rope chain
959: Net
960: Job tool form
961: Weapon
962: Machine
963: Machine general
964: Motor
965: Job machine
966: Transmission machine
967: Machine part
968: Electrical machine
969: Electrical equipment
970: Communication equipment
971: Computer

- 972: Applied electronic equipment
- 973: Electric parts
- 974: Optical instrument
- 975: Camera
- 976: Telescope Magnifier
- 977: Eyeglasses
- 978: Optical part
- 979: Mirror
- 980: Lense
- 981: Optical part
- 982: Meter
- 983: Meter degree speed
- 984: Clock
- 985: Meter weight and measure
- 986: Vehicle
- 987: Vehicle Main
- 988: Vehicle Main Land
- 989: Vehicle Main Water
- 990: Vehicle Main Air
- 991: Vehicle Part
- 992: Vehicle Part Land
- 993: Vehicle Part Water
- 994: Vehicle Part Air

1002: Mental state

- 1003: Intellectual product
- 1004: Studies Subject
- 1005: Study general
- 1006: Academic field Subject
- 1007: Knowledge Opinion
- 1008: Knowledge Intelligence
- 1009: Opinion
- 1010: Editorial
- 1011: Theory
- 1012: Theory
- 1013: Outlines detailed exposition
- 1014: Doctrine
- 1015: Thought
- 1016: Impression
- 1017: Idea
- 1018: Enlightenment
- 1019: Logic Meaning
- 1020: Logic
- 1021: Principle
- 1022: Affair
- 1023: Authenticity
- 1024: True
- 1025: False
- 1026: Substantial
- 1027: Conceptual
- 1028: Semantic
- 1029: Motive
- 1030: Main point
- 1031: Summary
- 1032: Theme
- 1033: Problem
- 1034: Secret principle
- 1035: Methodological
- 1036: Plan
- 1037: Artistic creation
- 1038: Literary creation
- 1039: Poetry
- 1040: Poem
- 1041: Japanese ode
- 1042: Haiku
- 1043: Narrative
- 1044: Novel
- 1045: Drama
- 1046: Historical record
- 1047: Physical creation
- 1048: Picture
- 1049: Sculpture
- 1050: Handicraft
- 1051: Calligraphy
- 1052: Photograph Portrait
- 1053: Creation Sound
- 1054: Music
- 1055: Musical composition
- 1056: Song
- 1057: Created
- 1058: Play
- 1059: Opera
- 1060: Dance
- 1061: Movie
- 1062: Language
- 1063: Language Content
- 1064: Speech
- 1065: Name
- 1066: Person name
- 1067: Title
- 1068: Name
- 1069: Number
- 1070: Talk Content
- 1071: Topic
- 1072: Episode
- 1073: Greeting
- 1074: Phrase
- 1075: Idiomatic phrase
- 1076: Proverb
- 1077: Joke
- 1078: Literary style
- 1079: Rhetoric
- 1080: Language Type
- 1081: Sentence Phrase Word
- 1082: Sentence
- 1083: Phrase
- 1084: Word
- 1085: Grammatical
- 1086: Phonological
- 1087: Phonetic
- 1088: Pronunciation
- 1089: Tone
- 1090: Kanji pronunciation
- 1091: Sign language
- 1092: Letter of Alphabet
- 1093: Alphabet type
- 1094: Font
- 1095: Calligraphic style
- 1096: Label
- 1097: Handwriting
- 1098: Brush stroke
- 1099: Mark Symbol
- 1100: Mark
- 1101: Mark type
- 1102: Coat of Arms
- 1103: Diagram Schema
- 1104: Diagram
- 1105: Figure
- 1106: Table
- 1107: Score
- 1108: Equation
- 1109: Document
- 1110: Article
- 1111: Article Main
- 1112: Article Part
- 1113: Document type
- 1114: Letter Mail
- 1115: Publication
- 1116: Information
- 1117: Newspaper
- 1118: Magazine
- 1119: Book content
- 1120: Catalog
- 1121: Calendar
- 1122: Ethic Religion
- 1123: Ethics
- 1124: Humane
- 1125: Righteous path
- 1126: Evil course
- 1127: Moral
- 1128: Virtue
- 1129: Vice
- 1130: Integrity
- 1131: Loyalty
- 1132: Fidelity Infidelity
- 1133: Fidelity
- 1134: Infidelity
- 1135: Treachery
- 1136: Debt of Favour
- 1137: Trust unfaithful
- 1138: Trust
- 1139: Unfaithful
- 1140: Good and Evil
- 1141: Good
- 1142: Evil
- 1143: Sin
- 1144: Religion
- 1145: Tradition Information Reputation
- 1146: Tradition History
- 1147: Information News
- 1148: Reputation
- 1149: Rumor
- 1150: Popularity
- 1151: Fame Dishonour
- 1152: Fame
- 1153: Dishonour

1154: Action

- 1155: System
- 1156: Political system
- 1157: Government form State affairs
- 1158: Government form
- 1159: State affairs
- 1160: Military
- 1161: Law
- 1162: Regulation
- 1163: Legal
- 1164: Treaty
- 1165: Right Obligation

1166: Right
1167: Obligation
1168: Economic system
1169: Economy
1170: Public economy
1171: Household economy
1172: Income Expenditure
1173: Income
1174: Expenditure
1175: Supply Demand
1176: Demand
1177: Supply
1178: Profit Loss
1179: Profit
1180: Loss
1181: Share Debt
1182: Share
1183: Debt
1184: Price Cost
1185: Price
1186: Cost
1187: Capital Money
1188: Capital Fund
1189: Property Asset
1190: Money
1191: Stock
1192: Tax
1193: Wage Fee Interest
1194: Allowance
1195: Remuneration
1196: Annuity
1197: Compensation
1198: Interest
1199: Fee
1200: Security
1201: Quotation
1202: Social system
1203: Register
1204: Postal
1205: Social system
1206: Success Failure Performance
1207: Success Failure
1208: Success
1209: Failure
1210: Performance
1211: Result
1212: Merit Demerit
1213: Merit
1214: Demerit
1215: Manner Custom
1216: Custom Fad
1217: Custom
1218: Fad
1219: Habit
1220: Habit
1221: Convention
1222: Fashion
1223: Ceremony Event
1224: Ceremony
1225: Celebration
1226: Wedding
1227: Funeral
1228: Religious ceremonial
1229: Festival
1230: Event
1231: Gathering
1232: Gathering
1233: Exhibition Entertainment
1234: Banquet

1236: Human activity

1237: Mind
1238: Spirit
1239: Sensation
1240: Feeling
1241: Perception
1242: Pain
1243: Madness
1244: Drunkenness
1245: Starvation Full
1246: Starvation Thirst
1247: Full
1248: Fatigue
1249: Sleep Awake
1250: Sleep
1251: Awake
1252: Dream
1253: Emotion
1254: Emotion
1255: Excitement Calm
1256: Excitement
1257: Calm
1258: Pleasure Suffering
1259: Suffering
1260: Pleasure
1261: Joy Sorrow
1262: Sorrow
1263: Joy
1264: Anger
1265: Surprise
1266: Fear
1267: Feeling towards self
1268: Relief Worry
1269: Relief
1270: Worry
1271: Satisfaction Dissatisfaction
1272: Satisfaction
1273: Dissatisfaction
1274: Repentance
1275: Introspection
1276: Fretfulness Relax
1277: Fretfulness
1278: Relax
1279: Confusion Composure
1280: Confusion
1281: Composure
1282: Self-respect
1283: Pride Humility
1284: Pride
1285: Humility
1286: Self-confidence Self-abandonment
1287: Self-confidence
1288: Self-abandonment
1289: Vanity
1290: Arrogance Modesty
1291: Arrogance
1292: Modesty
1293: Honour Shame
1294: Honour
1295: Shame
1296: Ego feeling
1297: Feeling towards others
1298: Empathy
1299: Like Dislike
1300: Like
1301: Dislike
1302: Love Hatred
1303: Love
1304: Hatred
1305: Degree of Intimacy
1306: Familiarity
1307: Alienation
1308: Good intention Malice
1309: Good intention
1310: Malice
1311: Sympathy Envy
1312: Sympathy
1313: Envy
1314: Gratitude Grudge
1315: Gratitude
1316: Grudge
1317: Respect Scorn
1318: Respect
1319: Scorn
1320: Regard Disregard
1321: Regard
1322: Value
1323: Disregard
1324: Appreciation bias
1325: Appreciation
1326: Bias
1327: Credence Discredit
1328: Credence
1329: Discredit
1330: Service Selfishness
1331: Loyalty Filial piety
1332: Chivalry
1333: Public spirit Selfishness
1334: Public spirit
1335: Selfishness
1336: Feeling towards others
1337: Sentiment
1338: Impressed
1339: Praise
1340: Aspiration
1341: Sentiments
1342: Mood
1343: Expression
1344: Facial
1345: Look
1346: Cry
1347: Laugh
1348: Wonder
1349: Vocal
1350: Shiver
1351: Intention
1352: Will
1353: Determination Hesitation
1354: Determination

1355: Hesitation
1356: Desire Ego Selflessness
1357: Desires
1358: Ego Selflessness
1359: Ego
1360: Selflessness
1361: Wish Disappointment
1362: Request
1363: Wish
1364: Expectation
1365: Disappointment
1366: Roused Discouraged
1367: Spurt
1368: Courage
1369: Discouragement
1370: Tenacity Resignation
1371: Tenacity
1372: Enthusiasm
1373: Obstination
1374: Giving up
1375: Weariness
1376: Change of mind
1377: Diligence Indolence
1378: Diligence
1379: Effort
1380: Pain Hard work
1381: Indolence
1382: Fortitude
1383: Patience
1384: Endurance
1385: Attitude
1386: Religious faith
1387: Learning Memory
1388: Learning
1389: Practice
1390: Imitation
1391: Memory
1392: Memorised
1393: Forgetfulness
1394: Recollection
1395: Thinking
1396: Contemplation
1397: Belief Doubt
1398: Doubt
1399: Conviction
1400: Perplexity
1401: Attention Negligence
1402: Caution
1403: Negligence
1404: Concern
1405: Recognition Understanding
1406: Recognition
1407: Understanding
1408: Identification
1409: Comparison Collation
1410: Comparison
1411: Collation
1412: Identification Confusion
1413: Classification
1414: Distinction
1415: Confusion
1416: Selection Rejection
1417: Selection
1418: Approval Rejection
1419: Calculation Measurement
1420: Calculation
1421: Measurement
1422: Investigation Research
1423: Research
1424: Enquiry
1425: Academic investigation
1426: Experiment
1427: Search
1428: Investigation
1429: Investigation
1430: Changing
1431: Inspection
1432: Observation
1433: Guess
1434: Imaginations
1435: Guess
1436: Estimation
1437: Fortune-telling
1438: Judgement
1439: Judgements
1440: Conclusion
1441: Decision
1442: Decision
1443: Verdict
1444: Resolution
1445: Appraisal
1446: Solutions undecided
1447: Solutions
1448: Pendency
1449: Affirmation Denial
1450: Affirmation
1451: Denial
1452: Error correction
1453: Error
1454: Correction
1455: Proof Distortion
1456: Proof
1457: Distortion
1458: Preparation
1459: Plan
1460: Project plan
1461: Strategy
1462: Observation Reading Writing
1463: Observation
1464: Seeing
1465: Eye-witness
1466: Gaze
1467: First sight
1468: Glance
1469: Looking on
1470: Seeing direction
1471: Distant view
1472: Viewing
1473: Overlook
1474: Appearing and disappearing
1475: Patrol
1476: Vigilance
1477: Suggestion
1478: Revelation
1479: Explicit or Hint
1480: Notice
1481: Designated
1482: Enumeration
1483: Hearing
1484: Reading writing
1485: Reading
1486: Writing
1487: Writing general
1488: Authoring
1489: Mention in writing
1490: Record activity
1491: Inscribing ownership
1492: Signature
1493: Seal
1494: Drawing
1495: Speech and Conduct
1496: Statement
1497: Expressions
1498: Call by name
1499: Expression
1500: Description
1501: Translation
1502: Utterance Silence
1503: Speech
1504: Silence
1505: Signal
1506: Talk
1507: Conversation
1508: Dialogue
1509: Lecture
1510: Question Answer
1511: Question
1512: Answer
1513: Consultation
1514: Discussion
1515: Meeting
1516: Proposition
1517: Argument Quarrel
1518: Argument
1519: Quarrel
1520: Criticism Defense
1521: Criticism
1522: Blame
1523: Defense
1524: Speech
1525: Address
1526: Advocacy
1527: Exposition
1528: Explanation
1529: Annotation
1530: Reports advice
1531: Report
1532: Declaration
1533: Advice
1534: Advice
1535: Pronouncement
1536: Confession
1537: Appeal
1538: Testimony
1539: Contribution article
1540: Reporting
1541: Announcements declaration
1542: Announcement
1543: Declaration

1544: Communication
1545: Communication
1546: Correspondence
1547: Telegraphic
1548: Telephone
1549: Transmissions information
1550: Transmission
1551: Notification
1552: Broadcast
1553: Creation
1554: Invention
1555: Creation Language
1556: Lierary writing
1557: Creation form
1558: Creation sound
1559: Creation
1560: Conduct
1561: Physical movement
1562: Whole body
1563: Gesture
1564: Sexual act
1565: Sitting Standing
1566: Carrying
1567: Standing
1568: Sit
1569: Looking up down
1570: Looking up
1571: Looking down
1572: Lying down
1573: Crouching
1574: Crawling
1575: Foot movement
1576: Walking
1577: Running
1578: Foot movement
1579: Hand movement
1580: Mouth movement
1581: Daily living
1582: Clothing
1583: Dressing up
1584: Change
1585: Undressing
1586: Wear accessory
1587: Food
1588: Eating and drinking
1589: Drinking
1590: Eating
1591: Meal
1592: Dwelling
1593: Residence
1594: Being at home
1595: Lodging
1596: Confinement
1597: Absent
1598: Moving
1599: Wandering
1600: Sleeping Waking
1601: Going to bed
1602: Rising
1603: Sleepless vigil
1604: Health Hygiene Beauty care
1605: Health
1606: Rest
1607: Leisure
1608: Hygiene Beauty care
1609: Make-up
1610: Haircut
1611: Bath
1612: Face washing
1613: Life
1614: Conduct
1615: Register Deregister
1616: Register
1617: Deregister
1618: Inheritance Branch family
1619: Inheritance
1620: Branch Extinct family
1621: Retirement Comeback
1622: Retirement
1623: Comeback
1624: Eminence Downfall
1625: Eminence
1626: Downfall
1627: Career Move
1628: Assumption of office
1629: Being in office
1630: Change of post
1631: Retirement
1632: Job hunting
1633: Academic Military
1634: Academic
1635: Entrance
1636: Attending
1637: Change of schools
1638: Graduation
1639: Taking an examination
1640: Pass Fail in exam
1641: Pass
1642: Fail
1643: Military
1644: Enlistment
1645: Discharge
1646: Celebration Funeral
1647: Celebration
1648: Funeral
1649: Mourning
1650: Marriage Divorce
1651: Marriage
1652: Divorce
1653: Religious act
1654: Worship
1655: Worship
1656: Prayer
1657: Leisure
1658: Travelling sightseeing
1659: Travelling
1660: Touring
1661: Outing
1662: Strolling Long ride
1663: Strolling
1664: Long ride
1665: Enjoying coolness
1666: Hunting
1667: Fishing
1668: Hunting
1669: Sight-seeing
1670: Performance
1671: Show
1672: Stunt
1673: Singing
1674: Musical performance
1675: Dance Play
1676: Martial arts
1677: Play game
1678: Play
1679: Game
1680: Sport
1681: Social Acquiantance
1682: Social life
1683: Acquaintance
1684: Friendship
1685: Reconciliation
1686: Breaking off relation
1687: Assembly
1688: Opening Adjournment of meeting
1689: Opening of meeting
1690: Adjournment of meeting
1691: Presence Absence
1692: Presence
1693: Absence
1694: Encounter Separation
1695: Interview
1696: Separation
1697: Refusing audience
1698: Visit Leaving
1699: Visit
1700: Taking leave
1701: Summons
1702: Invitation
1703: Guide
1704: Lure
1705: Reception
1706: Welcome and Send off
1707: Welcome
1708: Send-off
1709: Mediation
1710: Introduction
1711: Arbitration alienation
1712: Arbitration
1713: Alienation
1714: Salutation
1715: Negotiation Promise
1716: Promise
1717: Contract
1718: Guarantee
1719: Cancellation
1720: Negotiation
1721: Business meeting
1722: Proposal Withdrawal
1723: Proposal
1724: Withdrawal
1725: Request
1726: Persuasion
1727: For and against
1728: For and against
1729: Approval
1730: Opposition
1731: Consent Refusal
1732: Consent

1733: Refusals
1734: Permission Prohibition
1735: Permission
1736: Prohibition
1737: Confirmation Denial
1738: Confirmation
1739: Denial
1740: Cooperation
1741: Compromise
1742: Flattery
1743: Collaboration Participation
1744: Cooperation
1745: Joining Withdrawing
1746: Joining
1747: Withdrawing
1748: Service
1749: Contribution
1750: Loyal
1751: Filial piety
1752: Conflict
1753: Fight
1754: Dispute
1755: War
1756: Competition
1757: Victory Defeat
1758: Victory
1759: Draw
1760: Defeat
1761: Attack Defense
1762: Attack
1763: Defense
1764: Punitive Defense
1765: Raising army
1766: Invasion
1767: Defense
1768: Conquest Surrender
1769: Conquest
1770: Surrender
1771: Resistance Obedience
1772: Resistance
1773: Obedience
1774: Vengeance
1775: Infringement Prevention
1776: Infringement
1777: Prevention
1778: Control
1779: Management
1780: Rule
1781: Administration of justice
1782: Lawsuit Trial
1783: Lawsuit
1784: Trial
1785: Punishment
1786: Public safety
1787: Arrest
1788: Detention
1789: Release
1790: Establishment Operation
1791: Establishment
1792: Operation
1793: Affairs
1794: Appointment and dismissal
1795: Appointment
1796: Dismissal
1797: Impeachment
1798: Dispatch
1799: Recruitment Employment
1800: Recruitment
1801: Employment
1802: Recommendation
1803: Nomination
1804: Election
1805: Induction
1806: Guidance
1807: Guidance
1808: Education
1809: Civilization
1810: Instruction
1811: Adomonition
1812: Training
1813: Rescue Aid
1814: Rescue
1815: Assistance
1816: Help
1817: Protection
1818: Grace
1819: Care
1820: Request
1821: Request
1822: Urge
1823: Recruitment
1824: Order
1825: Restriction
1826: Regulation
1827: Intervention
1828: Non restriction
1829: Noninterference
1830: Liberation
1831: Exemption
1832: Encouragement
1833: Inducement
1834: Lure Charm
1835: Encouragement
1836: Instigation
1837: Observance Violation
1838: Observance
1839: Violation
1840: Obstruction Surmount
1841: Obstruction
1842: Surmount
1843: Treatment
1844: Treatment
1845: Favorable treatment
1846: Cold treatment
1847: Discrimination
1848: Persecution
1849: Recognition
1850: Expression of Gratitude
1851: Discourtesy
1852: Requital of favor
1853: Retaliation
1854: Apology
1855: Expression of gratitude
1856: Reward Punishment
1857: Reward
1858: Punishment
1859: Praise Censure
1860: Praise
1861: Slander
1862: Ridicule
1863: Threaten
1864: Deception Conciliation
1865: Deception
1866: Conciliation
1867: Transaction
1868: Acquisition
1869: Possession Store
1870: Possession
1871: Retention
1872: Storage
1873: Hand-carry
1874: Money received Payment
1875: Money received
1876: Payment
1877: Investment Consumption
1878: Investment
1879: Consumption
1880: Economising
1881: Extravagance
1882: Replenishment
1883: Collection Payment
1884: Collection
1885: Payment
1886: Price hike Price cut
1887: Price hike
1888: Price cut
1889: Loss Gain
1890: Gain
1891: Loss
1892: Trade
1893: Business transaction
1894: Import and export
1895: Accepting order Placing order
1896: Accepting order
1897: Placing order
1898: Exchange
1899: Sales Purchase
1900: Sales
1901: Purchase
1902: Giving Receipt
1903: Giving
1904: Transfer
1905: Presentation gift
1906: Conferment
1907: Receipt
1908: Loan Deposit
1909: Loan
1910: Lending
1911: Borrowing
1912: Deposit
1913: Deposit
1914: Return
1915: Recompense
1916: Collection Delivery
1917: Collection
1918: Distribution
1919: Delivery
1920: Toil
1921: Open Suspension of Business

1922: Labor Vain effort
1923: Labor
1924: In charge
1925: Vain effort
1926: Being employed
1927: Going to work
1928: Duty
1929: On duty
1930: Suspension of business
1931: Rest
1932: Holiday
1933: Absence
1934: Leaving early
1935: Strike
1936: Work
1937: Affairs
1938: Task
1939: Occupation
1940: Industry
1941: Business
1942: Manufacture
1943: Operation
1944: Production
1945: Construction
1946: Handicraft
1947: Repair
1948: Decorative
1949: Manufacture process
1950: Painting
1951: Polishing
1952: Carving
1953: Stretching
1954: Roasting
1955: Shaving
1956: Manufacture process
1957: Industrial
1958: Agriculture Forestry
1959: Agricultural
1960: Cultivation
1961: Farming
1962: Forestry
1963: Reforestation
1964: Lumbering
1965: Breeding Harvesting
1966: Breeding
1967: Stock-farming
1968: Cultivation
1969: Hunting
1970: Hunting
1971: Fishing
1972: Harvesting
1973: Mining
1974: Harvesting
1975: Manufacturing industry
1976: Printing bookbinding
1977: Printing
1978: Bookbinding
1979: Civil engineering
1980: Construction
1981: Transport
1982: Traffic
1983: Conveyance
1984: Cargo handling
1985: Medical care
1986: Publication entertainment
1987: Publications
1988: Editing
1989: Publication
1990: Issue
1991: Entertainments
1992: Entertainment
1993: Performance
1994: Exhibitions
1995: Housekeeping
1996: Needlework
1997: Laundry
1998: Cooking
1999: Cleaning
2000: Housekeeping
2001: Operation
2002: Manipulation
2003: Control
2004: Manipulation
2005: Use
2006: Treatment
2007: Installation
2008: Packing Stuffing
2009: Packing
2010: Wrapping
2011: Stuffing
2012: Pushing pulling supporting
2013: Pushing
2014: Pulling
2015: Supporting
2016: Piercing perforation
2017: Piercing
2018: Perforation
2019: Striking throwing shooting
2020: Striking
2021: Throwing
2022: Shooting
2023: Charging Recording
2024: Fire fighting
2025: Charging electricity
2026: Sound recording
2027: Photography
2028: Charging Recording
2029: Action
2030: Conduct
2031: Activity
2032: Deed
2033: Good deed
2034: Evil deed
2035: Killing
2036: Murder
2037: Suicide
2038: Assault
2039: Stealing
2040: Fraud
2041: Adultery
2042: Depravity
2043: Evil deed
2044: Deed
2045: Crime Disturbance
2046: Crime
2047: Criminal act
2048: Crime Guilt
2049: Disturbance
2050: Execution
2051: Enforcement
2052: Carried out
2053: Taking place

2054: Phenomenon

2055: Happening
2056: Misfortune
2057: Disaster
2058: Damage
2059: Incident
2060: Fire
2061: Calamity
2062: Panic
2063: Happening
2064: Change
2065: Origin Demise
2066: Origin
2067: Appearance
2068: Occurrence
2069: Ascendancy
2070: Revival
2071: Omen
2072: Exposure
2073: Disclosure
2074: Establishment
2075: Completion
2076: Demise
2077: Hiding
2078: Disappearence
2079: Downfall
2080: Cover
2081: Concealment
2082: Failure
2083: Abolition
2084: Abandonment
2085: Removal
2086: Adjustment
2087: Action
2088: Change Stability
2089: Change
2090: Stability
2091: Conversion
2092: Exchange
2093: Reformation
2094: Reform
2095: Restoration
2096: Correction
2097: Substitution
2098: Start End
2099: Start
2100: End
2101: Suspension
2102: Intermittence
2103: Consecutive
2104: Extinction
2105: Continuation
2106: Repetition
2107: Lapse of time
2108: Movement
2109: Dynamic

2110: Motion
2111: Vibration
2112: Floating
2113: Flutter
2114: Rotation
2115: Shaking
2116: Bounce
2117: Standing
2118: Sitting
2119: Invert Fall
2120: Slanting
2121: Walking
2122: Motion
2123: Static
2124: Standstill
2125: Fixed
2126: Stop
2127: Stagnation
2128: Process
2129: Migration Arrival Departure
2130: Migration
2131: Arrival Departure
2132: Departure
2133: Arrival
2134: Transit
2135: Advance
2136: Land Sea Air travel
2137: Land travel
2138: Sea travel
2139: Air travel
2140: Direction Tendency
2141: Encircling Wandering
2142: Encircling
2143: Wandering
2144: Flow Slide Flight
2145: Flow
2146: Slide
2147: Flight
2148: Passthrough Cutoff
2149: Pass through
2150: Penetration
2151: Conduction
2152: Circulation
2153: Cutoff
2154: Propulsion Escape Chase
2155: Preceding Following
2156: Preceding
2157: Following
2158: Propulsion
2159: Escape Chase
2160: Escape
2161: Evasion
2162: Chase
2163: Advance Retreat
2164: Advance
2165: Retreat
2166: U turn
2167: Retrace
2168: Abreast
2169: Crossing over
2170: Crossing water
2171: Crossing gap
2172: Coming Going
2173: Outgoing Returning
2174: Outgoing
2175: Returning
2176: Leaving Arriving
2177: Leaving
2178: Arriving
2179: Rise Descent
2180: Rise
2181: Descent
2182: Coming in out
2183: Exit Entry
2184: Exit
2185: Entry
2186: Putting out in
2187: Putting out
2188: Putting in
2189: Extraction Insertiong
2190: Extraction
2191: Insertion
2192: Absorption Leak
2193: Absorption
2194: Leak
2195: Pouring Drawing
2196: Pouring
2197: Drawing
2198: Opening Closing
2199: Opening
2200: Closing
2201: Burying Flooding
2202: Burying
2203: Flooding
2204: Encirclement Inclusion
2205: Encirclement
2206: Inclusion
2207: Rise Drop
2208: Ascent Descent
2209: Climbing Descent
2210: Climbing
2211: Descent
2212: Ascent Descent
2213: Rise
2214: Drop
2215: Falling Hanging down
2216: Falling
2217: Hanging down
2218: Lifting Lowering
2219: Lifting
2220: Lowering
2221: Mount Dismount
2222: Mount
2223: Dismount
2224: Floating Sinking
2225: Floating
2226: Sinking
2227: Seperation Combination
2228: Combination
2229: Union
2230: Concentration
2231: Accumulation Pile
2232: Accumulation
2233: Pile
2234: Unification
2235: Combination
2236: Mixture
2237: Blending
2238: Connection
2239: Adhesion
2240: Contact Approach Collision
2241: Contact Approach
2242: Collision
2243: Separation
2244: Detachment
2245: Dispersion
2246: Erosion
2247: Disassembly
2248: Fray
2249: Severance
2250: Flaking
2251: Deformation
2252: Destruction
2253: Bending
2254: Folding
2255: Folding
2256: Turn inside out
2257: Undulation
2258: Bulge
2259: Indent
2260: Arrangement
2261: Deformation
2262: Increase Decrease
2263: Increase Decrease
2264: Increase Decrease
2265: Increase
2266: Decrease
2267: Excess Shortage
2268: Excesses
2269: Shortage
2270: Sufficiency
2271: Sufficiency
2272: Insufficiency
2273: Addition
2274: Replenishment
2275: Increase Decrease
2276: Expansion Contraction
2277: Stretch Contraction
2278: Stretch
2279: Shrinkage
2280: Enlargement Reduction
2281: Enlargement
2282: Reduction
2283: Expansion Contraction
2284: Expansion
2285: Contraction
2286: Lengthening Shortening
2287: Lengthening
2288: Shortening
2289: Postponement Advance
2290: Postponement
2291: Advance
2292: Concentration Dilution
2293: Concentration
2294: Dilution
2295: Inflation Devaluation
2296: Inflation
2297: Devaluation
2298: Vicissitudes

2299: Flourishing
2300: Deterioration
2301: Development Retrogression
2302: Development
2303: Retrogression

2304: Natural phenomenon

2305: Non life phenomenon
2306: Object
2307: Non stimulus
2308: Chemical phenomenon
2309: Reaction
2310: Fire
2311: Ignition
2312: Combustion
2313: Extinguishment
2314: Physical phenomenon
2315: Physical change
2316: State change
2317: Solidification
2318: Melting
2319: Evaporation
2320: Liquefaction
2321: Sublimation
2322: Emulsification
2323: Dryness Moisture
2324: Dryness
2325: Moisture
2326: Clarity
2327: Clear
2328: Milky
2329: Pollution
2330: Physical change
2331: Thermal
2332: Heat
2333: Cold
2334: Wave
2335: Electricity
2336: Force physics
2337: Stimulus
2338: Brightness
2339: Light
2340: Darkness
2341: Shadow
2342: Shade light
2343: Shade of colour
2344: Gloss
2345: Optical
2346: Light ray
2347: Light source
2348: Projection
2349: Luminescence
2350: Flickering
2351: Colour
2352: Colour tint
2353: Colour change
2354: Sound
2355: Smell
2356: Taste
2357: Feel of touch
2358: Meteorological Astronomical
2359: Meteorological
2360: Weather
2361: Temperature
2362: Cold
2363: Warmth
2364: Rain
2365: Snow
2366: Dew Frost
2367: Dew
2368: Frost
2369: Cloud
2370: Fog Haze
2371: Fog
2372: Haze
2373: Wind
2374: Wave
2375: Tide
2376: Natural disaster
2377: Thunder
2378: Wind damage Flood
2379: Drought Cold damage
2380: Earthquake
2381: Landslide Avalanche
2382: Eruption
2383: Natural disaster
2384: Astronomical
2385: Life phenomenon
2386: Life Death
2387: Life
2388: Survival
2389: Living
2390: Birth Germination
2391: Birth
2392: Germination
2393: Growth
2394: Growth
2395: Luxuriant growth
2396: Bloom
2397: Bearing fruit
2398: Aging
2399: Aging animal
2400: Aging plant
2401: Rebirth
2402: Death
2403: Death animal
2404: Death plant
2405: Physiology
2406: Breathing
2407: Blood circulation Pulse
2408: Discharge
2409: Secretion
2410: Reproduction
2411: Fermentation
2412: Physiology
2413: Health Imperfection
2414: Health
2415: Imperfection
2416: Sickness
2417: Onset
2418: Recovery
2419: Disease type
2420: Physical disability
2421: Injury

2423: Existence

2424: Presence
2425: Positive
2426: Negative
2427: Distribution
2428: Intrinsic
2429: Remaining
2430: Conservation
2431: Stay

2432: Categorisation System

2433: Category
2434: Type
2435: Similar type
2436: Item
2437: Example
2438: Primary Secondary
2439: Primary
2440: Secondary
2441: Class
2442: System

2443: Relation

2444: Related
2445: Origin End
2446: Origin Source
2447: End
2448: Cause Effect
2449: Condition
2450: Cause
2451: Result
2452: Effect
2453: Influence
2454: Reason Purpose
2455: Reason
2456: Purpose
2457: Evidence
2458: Similarity
2459: Identical
2460: Similar
2461: Difference
2462: Conformance Nonconformance
2463: Conformance
2464: Nonconformance
2465: Correspondence
2466: Mutual
2467: Connection
2468: Correspondence
2469: Contrast
2470: Opposite
2471: Parallel
2472: Intersection
2473: Independency Dependency
2474: Independency
2475: Dependency
2476: Balance Imbalance
2477: Balance
2478: Imbalance
2479: Superiority Inferiority
2480: Superiority
2481: Equality
2482: Inferiority

2483: Characteristic

2484: Attribute
2485: Attribute Agent
2486: Personality
2487: Disposition
2488: Character
2489: Habit
2490: Attribute Thing
2491: Quality
2492: Physical property
2493: Advantage Disadvantage
2494: Advantage
2495: Disadvantage
2496: Essence
2497: Content
2498: Structure
2499: Power Ability
2500: Power Agent
2501: Force Charisma
2502: Ability
2503: Talent
2504: Culture Education
2505: Talent
2506: Arts

2507: State

2508: Aspectual
2509: Circumstance
2510: Actuality
2511: Circumstance
2512: Normality Abnormality
2513: Business condition
2514: Crop condition
2515: Fishing condition
2516: Situation
2517: World situtation
2518: Situation
2519: Safety
2520: Safe
2521: Dangerous
2522: Confusion
2523: Tension Relaxation
2524: Tension
2525: Relaxation
2526: Mood
2527: Trend
2528: Appearance
2529: Conditions
2530: Favorableness
2531: Unfavourableness
2532: Right Wrong
2533: Right
2534: Wrong
2535: Aspect
2536: Circumstance
2537: Standpoint
2538: Personal circumstance
2539: Personal status
2540: Personal background
2541: Personal history
2542: Nobility
2543: Nobility
2544: Lowliness
2545: Poverty Wealth
2546: Poverty
2547: Wealth
2548: Luck
2549: Good luck
2550: Bad luck
2551: Fortune
2552: Misfortune
2553: Good Fortune
2554: Peace Disturbance
2555: Peace
2556: Emergency
2557: Busyness Idleness
2558: Busyness
2559: Idleness
2560: Manner Appearance Figure
2561: Manner
2562: Appearance
2563: Figure

2564: Form

2565: Shape
2566: Point
2567: Line
2568: Angular
2569: Surface
2570: 2D shape
2571: Unevenness
2572: Wrinkle
2573: Crevice
2574: Hole
2575: Three dimensional
2576: 3D shape
2577: Lump
2578: Piece
2579: Grain
2580: Powder
2581: Bundle
2582: Line
2583: Frame
2584: Pattern

2585: Numerical

2586: Number
2587: Quantity
2588: Quantity Frequency
2589: Age
2590: Value Amount
2591: Weight and Measure
2592: Degree
2593: Speed
2594: Quantity
2595: Unit
2596: Calculated value
2597: Many Few
2598: Whole Part
2599: Whole
2600: Part
2601: Set
2602: Group
2603: Pair
2604: Single Multiple
2605: Single
2606: Multiple
2607: Extent Limit
2608: Extent
2609: Limit

2610: Location

2611: Position
2612: Seat
2613: Trace
2614: Range
2615: Spot
2616: Actual spot
2617: Approximate spot
2618: Boundary Joint
2619: Boundary
2620: Joint
2621: Inside Outside
2622: Inside
2623: Interior
2624: Depths
2625: Bottom
2626: Outside
2627: Opening
2628: Inner Outer surface
2629: Outer surface
2630: Reverse surface
2631: Shade surface
2632: Up Down
2633: Above Under
2634: Above
2635: In
2636: Under
2637: Upper Lower
2638: Upper
2639: Middle
2640: Lower
2641: Top
2642: Left Right
2643: Left
2644: Right
2645: Side
2646: In front Behind
2647: In front
2648: Behind
2649: Direction
2650: Direction
2651: Direction
2652: Compass point
2653: Centre Surrounding
2654: Centre
2655: Circumference
2656: End limit
2657: Corner
2658: Edge Tip
2659: Sharp tip
2660: Space between
2661: Distance
2662: Distance
2663: Proximity
2664: Adjoining
2665: Nearby
2666: Beside
2667: Edge
2668: Along

2669: Route

2670: Time

2671: Calendar day
2672: Season
2673: Spring
2674: Summer
2675: Fall
2676: Winter
2677: Seasonal term
2678: Date
2679: Year
2680: Month
2681: Week
2682: Day
2683: Day Night
2684: Morning
2685: Noon
2686: Evening
2687: Night
2688: Non calendar day
2689: Point in time
2690: Time
2691: Opportunity
2692: Moment hour
2693: Usual
2694: Period phase
2695: Activity
2696: Period Life
2697: Period History
2698: Present Past Future
2699: Present
2700: Past
2701: Future
2702: Order
2703: Prior Later
2704: Prior
2705: Later
2706: Beginning Ending
2707: Beginning
2708: End
2709: Midst
2710: Old New Quick Slow
2711: Before After
2712: Before
2713: After
2714: Present
2715: Schedule

APPENDIX C

WN-GT

WN-GT is a lexicon produced by tagging *WordNet* sense entries with concept labels from *GT* (see Appendix B), using the guidelines outlined in §5.3.1. It currently contains 130 sense entries for 102 English words. Of these, 95 are nouns, 9 adjectives and 26 verbs.

Since *WordNet* groups synonyms together (*synsets*), the number of entries in *WN-GT* is easily higher than those given above if all words in the same synsets were included here.

The format of each *WN-GT* entry is:

‹**word**› ‹**sense number**› ‹***WordNet* synset ID**› ‹**POS**› ‹*GT* concepts›
[‹*WordNet* gloss›]

The contents of *WN-GT* are given below.

---

air 1 13996249 n 756, 757, 2373, 2406 [a mixture of gases (especially oxygen) required for breathing; the stuff that the wind consists of]

accommodate 3 01148301 v 1882 [provide with something desired or needed]

apply 3 00740152 v 1723 [ask (for something)]

area 1 07980485 n 459 [a particular geographical region of indefinite boundary (usually serving some special purpose or distinguished by its people or culture or geography)]

array 1 01432466 v 2260, 2582 [lay out in a line)]

bank 1 07909067 n 374, 428, 1170, 1190, 1910 [a financial institution that accepts deposits and channels the money into lending activities]

bank 2 08639924 n 490, 495, 748, 2667 [sloping land (especially the slope beside a body of water)]

boat 1 02757236 n 989, 2136, 748 [a small vessel for travel on water]

canoe 1 02846484 n 2003, 993, 2158, 989 [small and light boat; pointed at both ends; propelled with a paddle]

check 11 01081605 n 1826, 2500, 1560, 2609, 2268 [the act of restraining power or action or limiting excess]

cheque 1 12623674 n 932, 374, 428, 1170, 1190 [a written order directing a bank to pay money]

circulation 1 05871897 n 1115, 920, 2151 [the dissemination of copies of periodicals (as newspapers or magazines)]

circulation 2 10700390 n 2142, 2407, 649 [movement through a circuit; especially the movement of blood through the heart and blood vessels]

circulation 4 12826961 n 1900, 1115, 2588 [number of copies of a newspaper or magazine that are sold]

circulation 5 06863504 n 2145, 956, 746 [free movement or passage through a series of vessels (as of water through pipes or sap through a plant)]

circulation 6 00351860 n 2151, 1145, 1190, 1550 [the spread or transmission of something (as news or money) to a wider group or area]

company 1 07568361 n 327, 428, 1892 [an institution created to conduct business]

coin 1 12629722 n 934, 1190, 714, 953, 780 [a metal piece (usually a disc) used as money]

drown 3 00349077 v 2403, 2203, 748, 616 [die from being submerged in water, getting water into the lungs, and asphyxiating]

declare 2 00933912 v 1543 [announce publicly or officially]

declare 5 02371123 v 1442, 1738, 1876, 1190 [authorize payments of]

decrease 1 00146081 v 2266, 2585 [decrease in size, extent, or range]

deposit 2 02243919 v 1875, 374, 428, 1170, 1190, 1120, 1872 [put into a bank account]

deposit 3 01532141 v 2220 [put (something somewhere) firmly]

derivation 1 07990476 n 2068, 2446 [the source from which something derives (i.e. comes or issues)]

derivation 5 04674161 n 72, 2151, 2540 [inherited properties shared with others of your bloodline]

dividend 1 12649050 n 1918, 1198, 374, 1172, 349, 1188, 1191 [that part of the earnings of a corporation that is distributed to its shareholders; usually paid quarterly]

dollar 1 12889447 n 1190, 2595 [the basic monetary unit in many countries; equal to 100 cents]

dust 1 13994846 n 737, 2580, 757 [fine powdery material such as dry earth or pollen that can be blown about in the air]

edge 1 08042056 n 2658, 2619, 2569 [the boundary of a surface]

erode 2 00265787 v 2246, 2252, 1776, 727, 731 [remove soil or rock]

exchange 4 01045967 n 1898, 1902 [the act of giving something in return for something received]

exchange 5 02887980 n 397, 970, 1548 [a workplace that serves as a telecommunications facility where lines from telephones can be connected together to permit communication]

exchange 6 03182555 n 432, 1899 [a workplace for buying and selling; open only to members]

exchange 8 01030156 n 1190, 1201, 1898 [reciprocal transfer of equivalent sums of money especially the currencies of different countries]

factory 1 03196165 n 439, 374, 1944 [a plant consisting of buildings with facilities for manufacturing]

fall 1 01914423 v 2211, 2336 [descend in free fall under the influence of gravity]

finance 1 01035703 n 1170, 1188, 1190, 1892 [the commercial activity of providing funds and capital]

financial 1 02716591 a 1170, 1190 [involving financial matters]

flower 1 10924920 n 698, 1960 [a plant cultivated for its blooms or blossoms]

frenzied 1 02307582 a 1243 [affected with or marked by frenzy or mania uncontrolled by reason]

frenzy 1 13577231 n 1243 [state of violent mental agitation]

gossip 2 06779782 n 1149, 1550, 1043, 1310 [a report (often malicious) about the behavior of other people]

government 1 07561622 n 167 [the organization that is the governing authority of a political unit]

house 3 03414475 n 865, 1595 [a building in which something is sheltered or located]

inactive 7 00038010 a 2559, 2513 [lacking activity; lying idle or unused]

increase 2 00147655 v 2265, 2597 [make bigger or more]

interest 4 12561863 n 1190, 1198, 1199, 1911 [a fixed charge for borrowing money; usually a percentage of the amount borrowed]

inundate 1 01482245 v 746, 2058, 2203, 2378 [fill quickly beyond capacity; as with a liquid]

inventory 1 06091895 n 1120, 762, 763, 434 [a detailed list of all the items in stock]

inventory 2 04155446 n 762, 763, 434 [the merchandise that a shop has on hand]

inventory 1 02398411 v 1490, 1120, 919 [make or include in an itemized record or report]

investment 1 01036404 n 374, 1179, 1190, 1878 [the act of investing; lay out money or capital in an enterprise with the expectation of profit]

investment 2 12576508 n 1179, 1190, 1878 [money that is invested with an expectation of profit]

jewellery 1 03464807 n 763, 766, 939, 725, 715 [an adornment (as a bracelet or ring or necklace) made of precious metals and set with gems (or imitation gems)]

lake 1 08746083 n 500, 748 [a body of (usually fresh) water surrounded by land]

land 1 01921967 v 2133, 2211, 2126 [reach or come to rest]

large 1 01335840 a 2591, 2597, 2608 [above average in size or number or quantity or magnitude or extent]

left 1 01967912 a 2643 [being or located on or directed toward the side of the body to the west when facing north]

lend 2 02256982 v 1910 [give temporarily; let have for a limited time]

lending 1 01030735 n 1910, 1190, 1189, 1914 [disposing of money or property with the expectation that the same thing (or an equivalent) will be returned]

lie 1 06341404 n 1025, 1054, 1865 [a statement that deviates from or perverts the truth]

line 29 03856841 n 374, 439, 962, 1942, 2498 [mechanical system in a factory whereby an article is conveyed through sites at which successive operations are performed on it]

loan 1 12640219 n 1190, 1910, 1198 [the temporary provision of money (usually at interest)]

magazine 1 06188114 n 1118, 920 [a periodic paperback publication]

market 2 107582637 n 432, 932, 1170, 1878 [the securities markets in the aggregate]

mechanical 1 01449749 a 962, 2498 [using (or as if using) mechanisms or tools or devices]

mine 1 03627241 n 442, 481, 723, 1973 [excavation in the earth from which ore and minerals are extracted]

mine 2 03627027 n 961, 812, 2282, 123 [explosive device that explodes on contact; designed to destroy vehicles or ships or to kill or maim personnel]

mine 1 01127636 v 442, 481, 723, 1973 [get from the earth by excavation]

mine 2 01090715 v 961, 812, 2282, 123 [lay mines]

mining 1 00867315 n 442, 481, 723, 1973 [the act of extracting ore or coal etc from the earth]

mining 2 00904921 n 961, 812, 2282, 123 [laying explosive mines in concealed places to destroy enemy personnel and equipment]

mintage 1 12629354 n 934, 1190, 714, 953, 780 [coins collectively]

money 1 12626498 n 1190 [the most common medium of exchange; functions as legal tender]

mouth 3 08775879 n 2627, 481, 479 [an opening that resembles a mouth (as of a cave or a gorge)]

mud 1 14105899 n 731, 748 [water soaked soil; soft wet earth]

news 1 06232539 n 1147, 1235 [new information about specific and timely events]

newspaper 1 05885165 n 920, 1117, 1147, 1110 [a daily or weekly publication on folded sheets; contains news and articles and advertisements]

newspaper 4 14110978 n 920, 1117, 770 [cheap paper made from wood pulp and used for printing newspapers]

note 4 12635129 n 934, 1190, 770, 374, 428, 1170 [a piece of paper money (especially one issued by a central bank]

number 2 12816962 n 2587 [a concept of quantity derived from zero and units]

paddle 1 01890957 v 2003, 993, 2158 [propel with a paddle]

panic 2 13590669 n 2062 [sudden mass fear and anxiety over anticipated events]

parachutist 1 09723154 n 246, 299, 990, 757, 2147, 2214 [a person who jumps from aircraft using a parachute]

parcel 1 03724948 n 768 [a wrapped container]

park 1 08089344 n 459, 468, 465 [a large area of land preserved in its natural state as public property]

park 2 08089569 n 423, 1657 [piece of open land for recreational use in an urban area]

path 4 08798671 n 418, 2129, 2134 [a line or route along which something travels or moves]

picnic 1 01132257 v 1663, 1588, 2626 [eat alfresco, in the open air]

pipe 2 03795536 n 956, 746, 756, 714, 786 [a long tube made of metal or plastic that is used to carry water or oil or gas]

place 1 08134869 n 388 [a point located with respect to surface features of some region]

popularity 1 04585843 n 1150, 1340 [the quality of being widely admired or accepted or sought after]

present 4 02197331 v 1904 [hand over formally]

proceed 1 01937463 v 2128, 2610, 2670 [move ahead; travel onward in time or space]

product 1 03608510 n 763, 1900 [commodities offered for sale]

product 2 03856368 n 533, 1553 [an artifact that has been created by someone or some process]

product 3 10677631 n 2451, 2511, 2518 [a consequence of someone's efforts or of a particular set of circumstances]

public 2 00465801 a 168 [affecting the people or community as a whole]

rate 1 12568239 n 2596, 1199 [amount of a charge or payment relative to some basis]

resentment 1 07083591 n 1273, 1264 [a feeling of deep and bitter anger and ill-will]

ringgit 1 12925903 n 1190, 2595 [the basic unit of money in Malaysia; equal to 100 sen]

river 1 08820496 n 496, 748 [a large natural stream of water (larger than a creek)]

rumour 1 06780062 n 1149, 1550, 1043, 1023 [gossip (usually a mixture of truth and untruth) passed around by word of mouth]

salary 1 112523352 n 1195, 1190 [something that remunerate]

sell 1 02177556 v 1900, 1190 [exchange or deliver for money or its equivalent]

sell 3 02180066 v 1581, 1171, 1900 [do business; offer for sale as for one's livelihood]

selling 1 01049567 n 1900, 1190 [the exchange of goods for an agreed sum of money]

shade 1 13188833 n 2342 [relative darkness caused by light rays being intercepted by an opaque body]

shop 1 04042110 n 434, 762, 763, 1900 [a mercantile establishment for the retail sale of goods or services]

shop 2 04424512 n 439, 1944 [small workplace where handcrafts or manufacture are done]

silt 1 14188741 n 731, 728, 496, 500 [mud or clay or small rocks deposited by a river or lake]

slope 2 04793119 n 2592, 2567, 2569 [the property possessed by a line or surface that departs from the horizontal]

spending 1 01058350 n 1879, 1190 [the act of spending or disbursing money]

stock 1 12577104 n 374, 1188, 1191 [the capital raised by a corporation through the issue of shares entitling holders to an ownership interest (equity)]

stock 3 04155446 n 762, 763, 434 [the merchandise that a shop has on hand]

stock 6 07610417 n 72, 2540 [the descendant of one individual]

succeed 1 02449033 v 1208, 2456, 1352 [attain success or reach a desired goal]

table 2 04209815 n 896, 2629, 895 [a piece of furniture having a smooth flat top that is usually supported by one or more vertical legs]

town 1 08135936 n 465 [an urban area with a fixed boundary that is smaller than a city]

troop 1 07774524 n 2595, 370, 299 [a group of soldiers]

troop 2 07774613 n 368, 329, 299 [a cavalry unit corresponding to an infantry company]

water 1 14000512 n 748 [binary compound that occurs at room temperature as a clear colorless odorless tasteless liquid; freezes into ice below 0 degrees centigrade and boils above 100 degrees centigrade; widely used as a solvent]

water 2 08651117 n 495, 499, 507 [the part of the earth's surface covered with water (such as a river or lake or ocean]

wage 1 112523352 n 1195, 1190 [something that remunerate]

wide 5 02473433 a 2591, 2597, 2608 [great in degree]

wild 2 02306038 a 2389, 1960, 1967 [in a natural state; not tamed or domesticated or cultivated]

wind 1 10783344 n 2373, 757 [air moving (sometimes with considerable force) from an area of high pressure to an area of low pressure]

withdraw 4 2245322 v 2087, 1868 [cause to be returned]

year 1 14343019 n 2679, 2595 [a period of time containing 365 (or 366) days]

APPENDIX D

DETAILED RESULTS OF THE SENSE-TAGGING EXPERIMENT

In each (simplified) SSTC below, monosemous content words (with repect to *WN-GT* as found in Appendix C) are tagged with their (only) sense number as listed by *WN-GT*, while ambiguous word occurrences are highlighted in **bold**. Each SSTC is accompanied by a table showing the CSim values assigned to the possible senses of the ambiguous word by `SenseTagger`. The sense selected by `SenseTagger` is highlighted in grey, while the sense selected by a human tagger is marked with †.

1. The [DET] **bank** [N] accommodated [V]#3 him [PRON] with [PREP] a [DET] loan [N]#1 . [.]

bank #*n*	CSim
†1	†0.7907
2	0.5544

2. He [PRON] applied [V]#3 to [PREP] the [DET] **bank** [N] for [PREP] a [DET] loan [N]#1 . [.]

bank #*n*	CSim
†1	†0.8018
2	0.6435

3. The [DET] new [A] **bank** [N] rates [N]#1 was [V] a [DET] check [N]#11 on [PREP] spending [N]#1 . [.]

bank #*n*	CSim
†1	†0.7745
2	0.6200

4. The [DET] **circulation** [N] of [PREP] air [N]#1 through [PREP] the [DET] pipes [N]#2 . [.]

circulation #*n*	CSim
1	0.7481
2	0.6050
4	0.5639
†5	†0.7865
6	0.6664

5. The [DET] **circulation** [N] of [PREP] such [DET] rumours [N]#1 will [AU_V] only [ADV] fan [V] public [A]#2 resentment [N]#1 against [PREP] the [DET] ailing [A] government [N]#1 . [.]

circulation #n	CSim
1	0.7814
2	0.5403
4	0.7164
5	0.4560
†6	†0.8667

6. The [DET] **newspaper** [N] has [V] a [DET] wide [A]#5 **circulation** [N] . [.]

newspaper #n	CSim
†1	†0.8814
4	0.8018
circulation #n	CSim
1	0.8348
2	0.5079
†4	†0.7821
5	0.5197
6	0.7658

7. The [DET] government [N]#1 has [V] no [DET] plans [N] to [AU_INF] increase [V]#2 the [DET] number [N]#2 of [PREP] one [NUM_CARD] ringgit [N]#1 coins [N]#1 in [PREP] **circulation** [N] . [.]

circulation #n	CSim
1	0.7930
2	0.5957
4	0.7191
5	0.6297
†6	†0.7804

8. The [DET] one [NUM_CARD] thousand [NUM_CARD] dollar [N]#1 note [N]#4 might [AU_V] be [AU_V] withdrawn [V_EN]#4 from [PREP] **circulation** [N] . [.]

circulation #n	CSim
1	0.8226
2	0.6755
4	0.7000
5	0.6028
†6	†0.8139

9. The [DET] **bank** [N] **declared** [V] a [DET] 5 % [NUM_CARD] dividend [N]#1 . [.]

declare #n	CSim
2	0.6914
†5	†0.8690
bank #n	CSim
†1	†0.8059
2	0.5533

10. She [PRON] **deposited** [V] the [DET] parcel [N]#1 on [PREP] the [DET] table [N]#2 . [.]

deposit #n	CSim
2	0.6979
†3	†0.6438

11. To [AU_INF] **deposit** [V] o's [ABBR] wages [N]#1 in [PREP] the [DET] **bank** [N] . [.]

deposit #n	CSim
†2	†0.8308
3	0.5574
bank #n	CSim
†1	†0.8509
2	0.6073

12. To [AU_INF] **deposit** [V] ten [NUM_CARD] thousand [NUM_CARD] dollars [N]#1 on [PREP] a [DET] new [A] house [N]#3 . [.]

deposit #*n*	CSim
†2	†0.7979
3	0.5947

13. The [DET] strong [A] winds [N]#1 **deposited** [V] dust [N]#1 over [PREP] a [DET] wide [A]#5 area [N]#1 . [.]

deposit #*n*	CSim
2	0.7417
†3	†0.6251

14. The [DET] river [N]#1 **bank** [N] was [AU_V] completely [ADV] deprived [V_EN] of [PREP] shade [N]#1 . [.]

bank #*n*	CSim
1	0.5640
†2	†0.8451

15. He [PRON] claimed [V] **derivation** [N] from [PREP] French [A] **stock** [N] . [.]

derivation #*n*	CSim
1	0.7295
†5	†0.8774
stock #*n*	CSim
1	0.5686
3	0.5464
†6	†0.7292

16. **Bank** [N] lending [N]#1 rates [N]#1 were [V] high [A] even [ADV] then [ADV] . [.]

bank #*n*	CSim
†1	†0.8006
2	0.5976

17. Wild [A]#2 flowers [N]#1 edge [N]#1 the [DET] river [N]#1 **bank** [N] . [.]

bank #*n*	CSim
1	0.5549
†2	†0.7392

18. Water [N]#1 has [AU_V] eroded [V_EN]#2 the [DET] river [N]#1 **banks** [N] . [.]

water #*n*	CSim
†1	†0.7100
2	0.6866
bank #*n*	CSim
1	0.4797
†2	†0.8120

19. The [DET] river [N]#1 burst [V] its [GEN_PRON] **banks** [N] and [CC] inundated [V]#1 the [DET] town [N]#1 . [.]

bank #*n*	CSim
1	0.6572
†2	†0.7913

20. The [DET] **stock** [N] must [AU_V] be [AU_V] inventoried [V_EN]#1 once [ADV] a [DET] year [N]#1 . [.]

stock #*n*	CSim
1	0.5956
†3	†0.6236
6	0.6169

21. The [DET] **bank** [N] has [V] a [DET] slope [N]#2 of [PREP] 1 [NUM_CARD] in [PREP] 3 [NUM_CARD] . [.]

bank #*n*	CSim
1	0.7149
†2	†0.7226

22. The [DET] **stock** [N] market [N]#2 is [V] inactive [A]#7 today [N] . [.]

stock #*n*	CSim
†1	†0.5955
3	0.5927
6	0.5840

23. The [DET] left [A]#1 **bank** [N] of [PREP] the [DET] river [N]#1 . [.]

bank #*n*	CSim
1	0.5944
†2	†0.8316

24. Finance [N]#1 companies [N]#1 lend [V]#2 money [N]#1 at [PREP] almost [ADV] the [DET] same [A] interest [N]#4 rates [N]#1 as [PREP] **banks** [N] . [.]

bank #*n*	CSim
†1	†0.8306
2	0.5538

25. The [DET] **Stock** [N] **Exchange** [N] is [V] a [DET] madhouse [N] today [N] due to [PREP] panic [N]#2 selling [N]#1 . [.]

exchange #*n*	CSim
4	0.7627
5	0.4854
†6	†0.7969
8	0.7965
stock #*n*	CSim
†1	†0.6715
3	0.5441
6	0.5314

26. The [DET] new [A] mintage [N]#1 is [V] now [ADV] in [PREP] **circulation** [N] . [.]

circulation #*n*	CSim
1	0.8491
2	0.6321
4	0.6276
5	0.7126
†6	†0.7790

27. Only [ADV] one [NUM_CARD] parachutist [N]#1 succeeded [V]#1 in [PREP] landing [ING]#1 on [PREP] the [DET] near [A] **bank** [N] of [PREP] the [DET] river [N]#1 . [.]

bank #*n*	CSim
1	0.5968
†2	†0.8039

28. He [PRON] paddled [V]#1 the [DET] canoe [N]#1 to [PREP] the [DET] **bank** [N] . [.]

bank #*n*	CSim
1	0.7361
†2	†0.6951

29. He [PRON] arrayed [V]#1 his [GEN_PRON] **troops** [N] on [PREP] the [DET] river [N]#1 **bank** [N]#2 . [.]

troop #*n*	CSim
1	0.9204
†2	†0.7350
bank #*n*	CSim
1	0.5953
†2	†0.8010

30. The [DET] jewellery [N]#1 is [AU_V] kept [V_EN] in [PREP] the [DET] **bank** [N] . [.]

bank #*n*	CSim
†1	†0.7379
2	0.7413

31. They [PRON] picnicked [V]#1 on [PREP] the [DET] **bank** [N] of [PREP] the [DET] river [N]#1 . [.]

bank #*n*	CSim
1	0.5886
†2	†0.8307

32. He [PRON] has [AU_V] not [NEG-PART] yet [ADV] presented [V_EN]#4 the [DET] cheque [N]#1 at [PREP] the [DET] **bank** [N] . [.]

bank #*n*	CSim
†1	†0.8414
2	0.6200

33. This [DET] path [N]#4 proceeds [V]#1 through [PREP] the [DET] **park** [N] to [PREP] the [DET] river [N]#1 **bank** [N] . [.]

park #*n*	CSim
1	0.7480
†2	†0.7481
bank #*n*	CSim
1	0.6399
†2	†0.7679

34. Silt [N]#1 **deposited** [V] at [PREP] the [DET] mouth [N]#3 of [PREP] the [DET] river [N]#1 . [.]

deposit #*n*	CSim
2	0.7012
†3	†0.6468

35. The [DET] strong [A] winds [N]#1 **deposited** [V] dust [N]#1 over [PREP] a [DET] wide [A]#5 area [N]#1 . [.]

deposit #*n*	CSim
2	0.7417
†3	†0.6251

APPENDIX E

WORD LEVEL SUB-S-SSTCS AND ORIGINATING EXAMPLES

1. **E:** bank [N] **M:** bank [N] **Frequency:** 8

   (a) **Example 194:**

   E: The [DET] **bank** [N]#1 accommodated [V]#3 him [PRON] with [PREP] a [DET] loan [N]#1 . [.]

   M: **bank** [N] itu [DET] memberinya [V] pinjaman [N] . [.]

   (b) **Example 1635:**

   E: He [PRON] applied [V]#3 to [PREP] the [DET] **bank** [N]#1 for [PREP] a [DET] loan [N]#1 . [.]

   M: dia [PRON] memohon [V] pinjaman [N] daripada [PREP] **bank** [N] itu [DET] . [.]

   (c) **Example 2159:**

   E: The [DET] new [A] **bank** [N]#1 rates [N]#1 was [V] a [DET] check [N]#11 on [PREP] spending [N]#1 . [.]

   M: kadar [N] baru [A] **bank** [N] itu [DET] merupakan [V] satu [DET] kawalan [N] atas [PREP] perbelanjaan [N] . [.]

   (d) **Example 3348:**

   E: The [DET] **bank** [N]#1 declared [V]#5 a [DET] 5 % [NUM_CARD] dividend [N]#1 . [.]

   M: **bank** [N] itu [DET] mengisytiharkan [V] dividen [N] 5 % [DET] . [.]

(e) **Example 3684:**

E: To [AU_INF] deposit [V]#2 o's [ABBR] wages [N]#1 in [PREP] the [DET] **bank** [N]#1 . [.]

M: menyimpan wang [V] gaji [N] sso [ABBR] di [PREP] **bank** [N] . [.]

(f) **Example 4078:**

E: Bank [N]#1 lending [N]#1 rates [N]#1 were [V] high [A] even [ADV] then [ADV] . [.]

M: [] pada masa itu pun [ADV] kadar pemberian pinjaman [N] **bank** [N] tinggi [A] . [.]

(g) **Example 17108:**

E: The [DET] jewellery [N]#1 is [AU_V] kept [V_EN] in [PREP] the [DET] **bank** [N]#1 . [.]

M: barang - barang kemas [N] itu [DET] disimpan [V_EN] di [PREP] **bank** [N] . [.]

(h) **Example 24071:**

E: He [PRON] has [AU_V] not [NEG-PART] yet [ADV] presented [V_EN]#4 the [DET] cheque [N]#1 at [PREP] the [DET] **bank** [N]#1 . [.]

M: dia [PRON] masih belum lagi [AU_V] menyerahkan [V_EN] cek [N] itu [DET] kepada [PREP] **bank** [N] . [.]

2. **E:** bank [N] **M:** tebing [N] **Frequency:** 8

(a) **Example 3700:**

E: The [DET] river [N]#1 **bank** [N]#2 was [AU_V] completely [ADV] deprived [V_EN] of [PREP] shade [N]#1 . [.]

M: **tebing** [N] sungai [N] itu [DET] tidak mendapat [V_EN] naungan [N] langsung [AU_V] . [.]

(b) **Example 4737:**

E: Wild [A]#2 flowers [N]#1 edge [N]#1 the [DET] river [N]#1 **bank** [N]#2 . [.]

M: pokok - pokok bunga [N] liar [A] meminggiri [N] **tebing** [N] sungai [N] itu [DET] . [.]

(c) **Example 9112:**

E: The [DET] **bank** [N]#2 has [V] a [DET] slope [N]#2 of [PREP] 1 [NUM_CARD] in [PREP] 3 [NUM_CARD] . [.]

M: **tebing** [N] itu [DET] mempunyai [V] kecuraman [N] 1 [NUM_CARD] dalam [PREP] 3 [NUM_CARD] . [.]

(d) **Example 9974:**

E: The [DET] left [A]#1 **bank** [N]#2 of [PREP] the [DET] river [N]#1 . [.]

M: **tebing** [N] kiri [A] sungai [N] . [.]

(e) **Example 11662:**

E: Only [ADV] one [NUM_CARD] parachutist [N]#1 succeeded [V]#1 in [PREP] landing [ING]#1 on [PREP] the [DET] near [A] **bank** [N]#2 of [PREP] the [DET] river [N]#1 . [.]

M: hanya [ADV] seorang [NUM_CARD] ahli payung [N] berjaya [V] mendarat [ING] di [PREP] **tebing** [N] sungai [N] sebelah sini [A] . [.]

(f) **Example 12399:**

E: He [PRON] paddled [V]#1 the [DET] canoe [N]#1 to [PREP] the [DET] **bank** [N]#2 . [.]

M: dia [PRON] mendayung [V] kanu [N] ke [PREP] **tebing** [N] . [.]

(g) **Example 18090:**

E: They [PRON] picniced [V]#1 on [PREP] the [DET] **bank** [N]#2 of [PREP] the [DET]

river [N]#1 . [.]

M: mereka [PRON] berkelah [V] di [PREP] **tebing** [N] sungai [N] . [.]

(h) **Example 24531:**

E: This [DET] path [N]#4 proceeds [V]#1 through [PREP] the [DET] park [N]#2 to [PREP] the [DET] river [N]#1 **bank** [N]#2 . [.]

M: lorong [N] ini [DET] menuju [V] ke [PREP] **tebing** [N] sungai [N] melalui [PREP] taman [N] itu [DET] . [.]

3. **E:** circulation [N] **M:** edaran [N] **Frequency:** 2

(a) **Example 2301:**

E: The [DET] newspaper [N]#1 has [V] a [DET] wide [A]#5 **circulation** [N]#4 . [.]

M: akhbar [N] itu [DET] mempunyai [V] **edaran** [N] yang luas [DET] . [.]

(b) **Example 10451:**

E: The [DET] new [A] mintage [N]#1 is [V] now [ADV] in [PREP] **circulation** [N]#6 . [.]

M: [] wang syiling [N] baru [A] kini [ADV] dalam [PREP] **edaran** [N] . [.]

4. **E:** circulation [N] **M:** penyebaran [N] **Frequency:** 1

(a) **Example 2300:**

E: The [DET] **circulation** [N]#6 of [PREP] such [DET] rumours [N]#1 will [AU_V] only [ADV] fan [V] public [A]#2 resentment [N]#1 against [PREP] the [DET] ailing [A] government [N]#1 . [.]

M: **penyebaran** [N] khabar angin [N] spt [DET] itu [DET] akan [AU_V] hanya [ADV] memarakkan [V] rasa marah [N] orang ramai [A] terhadap [PREP] kerajaan [N] yang semakin lemah [A] itu [DET] . [.]

5. **E:** circulation [N] **M:** peredaran [N] **Frequency:** 3

(a) **Example 2299:**

E: The [DET] **circulation** [N]#5 of [PREP] air [N]#1 through [PREP] the [DET] pipes [N]#2 . [.]

M: **peredaran** [N] udara [N] melalui [PREP] paip - paip [N] itu [DET] . [.]

(b) **Example 2306:**

E: The [DET] government [N]#1 has [V] no [DET] plans [N] to [AU_INF] increase [V]#2 the [DET] number [N]#2 of [PREP] one [NUM_CARD] ringgit [N]#1 coins [N]#1 in [PREP] **circulation** [N]#6 . [.]

M: kerajaan [N] tidak mempunyai [V] rancangan [N] untuk [AU_INF] menambah [V] jumlah [N] syiling [N] seringgit [NUM_CARD] dalam [PREP] **peredaran** [N] . [.]

(c) **Example 2309:**

E: The [DET] one [NUM_CARD] thousand [NUM_CARD] dollar [N]#1 note [N]#4 might [AU_V] be [AU_V] withdrawn [V_EN]#4 from [PREP] **circulation** [N]#6 . [.]

M: wang kertas [N] seribu [NUM_CARD] ringgit [N] mungkin [AU_V] ditarik [V_EN] daripada [PREP] **peredaran** [N] . [.]

6. **E:** deposit [V] **M:** membayar wang muka [V] **Frequency:** 1

(a) **Example 3686:**

E: To [AU_INF] **deposit** [V]#2 ten [NUM_CARD] thousand [NUM_CARD] dollars [N]#1 on [PREP] a [DET] new [A] house [N]#3 . [.]

M: **membayar wang muka** [V] sepuluh [NUM_CARD] ribu [NUM_CARD] ringgit [N] untuk [PREP] rumah [N] baru [DET] . [.]

7. **E:** deposit [V] **M:** menyimpan wang [V] **Frequency:** 1

   (a) **Example 3684:**

   E: To [AU_INF] **deposit** [V]#2 o's [ABBR] wages [N]#1 in [PREP] the [DET] bank [N]#1 . [.]

   M: **menyimpan wang** [V] gaji [N] sso [ABBR] di [PREP] bank [N] . [.]

8. **E:** deposit [V] **M:** meletakkan [V] **Frequency:** 1

   (a) **Example 3681:**

   E: She [PRON] **deposited** [V]#3 the [DET] parcel [N]#1 on [PREP] the [DET] table [N]#2 . [.]

   M: dia [PRON] **meletakkan** [V] bungkusan [N] itu [DET] di atas [PREP] meja [N] . [.]

9. **E:** deposit [V] **M:** melonggokkan [V] **Frequency:** 1

   (a) **Example 3687:**

   E: The [DET] strong [A] winds [N]#1 **deposited** [V]#3 dust [N]#1 over [PREP] a [DET] wide [A]#5 area [N]#1 . [.]

   M: angin [N] kencang [A] **melonggokkan** [V] debu [N] kawasan [N] yang luas [DET] . [.]

10. **E:** deposit [V] **M:** yang terlonggok [V] **Frequency:** 1

    (a) **Example 3688:**

    E: Silt [N]#1 **deposited** [V]#3 at [PREP] the [DET] mouth [N]#3 of [PREP] the [DET] river [N]#1 . [.]

    M: kelodak [N] **yang terlonggok** [V] di [PREP] muara [N] sungai [N] . [.]

11. **E:** stock [N] **M:** keturunan [N] **Frequency:** 1

    (a) **Example 3723:**

    E: He [PRON] claimed [V] derivation [N]#5 from [PREP] French [A] **stock** [N]#6 . [.]

    M: dia [PRON] mengakui [V] berasal [N] daripada [PREP] **keturunan** [N] Perancis [A] . [.]

12. **E:** stock [N] **M:** saham [N] **Frequency:** 2

    (a) **Example 9166:**

    E: The [DET] **stock** [N]#1 market [N]#2 is [V] inactive [A]#7 today [N] . [.]

    M: [] pasaran [N] **saham** [N] tidak cergas [A] hari ini [N] . [.]

    (b) **Example 10095:**

    E: The [DET] **Stock** [N]#1 Exchange [N]#6 is [V] a [DET] madhouse [N] today [N] due to [PREP] panic [N]#2 selling [N]#1 . [.]

    M: [] Pasaran [N] **Saham** [N] hiruk - pikuk [N] hari ini [N] kerana [PREP] penjualan saham [N] panik [N] . [.]

13. **E:** stock [N] **M:** stok [N] **Frequency:** 1

    (a) **Example 8422:**

    E: The [DET] **stock** [N]#3 must [AU_V] be [AU_V] inventoried [V_EN]#1 once [ADV] a [DET] year [N]#1 . [.]

    M: **stok** [N] itu [DET] mesti [AU_V] dibuat inventori [V_EN] sekali setahun [N] . [.]

APPENDIX F

DETAILED RESULTS OF THE TRANSLATION EXPERIMENT

The following shows the translations produced by $EBMT_{cv}$ and $EBMT_o$ for 10 English phrases or sentences. Each output is accompanied by a table showing the CSim values assigned by $EBMT_{cv}$ to the candidate sub-S-SSTCs. For comparison, three CSim values are shown:

$$\text{CSim}_\text{p} = \text{CSim}(\ V_\text{clue}(\text{Input}), V_{\textbf{profile}}(\sigma)\ )$$

$$\text{CSim}_\text{d} = \text{CSim}(\ V_\text{clue}(\text{Input}), V_{\textbf{lex\_def}}(\sigma)\ )$$

$$\text{CSim}_\text{c} = \text{CSim}(\ V_\text{clue}(\text{Input}), V_{\textbf{context}}(\sigma)\ )$$

The sub-S-SSTC selected to be used in the final translation (as determined by CSim_p) is highlighted in grey, while those marked with † indicate translations that are deemed acceptable to a human reader.

1. **Input:** We walked along the river **bank**.
   Outputs:
   (a) $EBMT_{cv}$:
   kami berjalan di sepanjang **tebing** sungai itu.
   (b) $EBMT_o$:
   kami berjalan di sepanjang sungai itu **bank**.

English Substring	Candidate Malay Substring	CSim_p	CSim_d	CSim_c
† the river bank	† tebing sungai	0.7538	0.9557	0.1752
† the river bank	† tebing sungai itu	0.8507	0.9557	0.0892
bank	bank	0.0805	0.0904	0.0757
† bank	† tebing	0.6420	0.7643	0.5811

2. **Input:** He fell into the river from the **bank**.
   Outputs:

   (a) $EBMT_{cv}$:
   dia jatuh ke dalam sungai itu dari **tebing** itu.

   (b) $EBMT_{o}$:
   dia jatuh ke dalam sungai itu dari **bank**.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
the bank	bank itu	0.1112	0.1154	0.0973
the bank	bank	0.1731	0.1154	0.1862
† the bank	† tebing itu	0.4306	0.5631	0.1135
† the bank	† tebing	0.5512	0.5631	0.2485

3. **Input:** He drowned near the **bank**.
   Outputs:

   (a) $EBMT_{cv}$:
   dia mati lemas dekat **tebing**.

   (b) $EBMT_{o}$:
   dia mati lemas dekat **bank**.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
the bank	bank itu	0.1631	0.1281	0.1573
the bank	bank	0.2047	0.1281	0.2240
† the bank	† tebing itu	0.2193	0.1686	0.1760
† the bank	† tebing	0.3827	0.1686	0.3949

4. **Input:** I went to my **bank** to **deposit** my salary.
   Outputs:

   (a) $EBMT_{cv}$:
   saya pergi ke **bank** saya untuk **menyimpan wang** gaji saya.

   (b) $EBMT_{o}$:
   saya pergi ke **bank** saya untuk **membayar wang muka** gaji saya.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
† bank	† bank	0.8510	0.7492	0.8325
bank	tebing	0.3400	0.1575	0.3509
deposit	membayar wang muka	0.6615	0.6837	0.4932
† deposit	† menyimpan wang	0.8790	0.6837	0.9532
deposited	meletakkan	0.1685	0.1245	0.1179
deposited	melonggokkan	0.2635	0.1245	0.2675
deposited	terlonggok	0.2837	0.1245	0.2927

5. **Input:** I went to my **bank** to **deposit** my wages.
   Outputs:

   (a) $EBMT_{cv}$:
   saya pergi ke **bank** saya untuk **menyimpan wang** gaji saya.

   (b) $EBMT_o$:
   saya pergi ke **bank** saya untuk **menyimpan wang** gaji saya.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
† bank	† bank	0.8510	0.7492	0.8325
bank	tebing	0.3400	0.1575	0.3209
deposit	membayar wang muka	0.6615	0.6837	0.4932
† deposit	† menyimpan wang	0.8790	0.6837	0.9532
deposited	meletakkan	0.1685	0.1245	0.1179
deposited	melonggokkan	0.2635	0.1245	0.2675
deposited	terlonggok	0.2837	0.1245	0.2827

6. **Input:** The river **deposited** a lot of mud along the **bank**.
   Outputs:

   (a) $EBMT_{cv}$:
   sungai itu **terlonggok** banyak lumpur di sepanjang **tebing** itu.

   (b) $EBMT_o$:
   sungai itu **menyimpan wang** banyak lumpur di sepanjang **bank**.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
the bank	bank itu	0.3168	0.3704	0.2622
the bank	bank	0.4833	0.3704	0.4984
† the bank	† tebing itu	0.4787	0.5943	0.1582
† the bank	† tebing	0.5858	0.5943	0.2682
deposit	membayar wang muka	0.3011	0.2997	0.2361
deposit	menyimpan wang	0.3038	0.2997	0.2361
deposited	meletakkan	0.2102	0.0711	0.2314
† deposited	† melonggokkan	0.3678	0.0711	0.4761
† deposited	† terlonggok	0.6352	0.0711	0.8405

7. **Input:** The **circulation** of the bad news caused panic.
   Outputs:

   (a) $EBMT_{cv}$:
   penyebaran berita buruk itu itu menyebabkan panik.

   (b) $EBMT_o$:
   penyebaran berita buruk itu itu menyebabkan keadaan cemas.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
circulation	edaran	0.5682	0.7074	0.3862
† circulation	† penyebaran	0.5959	0.6540	0.3605
circulation	peredaran	0.5165	0.6081	0.4104

8. **Input:** The **circulation** of our magazine has decreased.
   Outputs:

   (a) $EBMT_{cv}$:
   edaran majalah kami itu telah berkurangan.

   (b) $EBMT_o$:
   penyebaran majalah kami itu telah berkurangan.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
† circulation	† edaran	0.8473	0.7018	0.8043
circulation	penyebaran	0.4828	0.5310	0.2909
circulation	peredaran	0.6783	0.5289	0.6576

9. **Input: circulation** of gossip
   Outputs:

   (a) $EBMT_{cv}$:
   penyebaran gosip

   (b) $EBMT_{o}$:
   edaran gosip

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
circulation	edaran	0.5299	0.6668	0.3557
† circulation	† penyebaran	0.7251	0.6657	0.5686
circulation	peredaran	0.3445	0.4758	0.2428

10. **Input:** The shop has depleted its **stock**.
    Outputs:

    (a) $EBMT_{cv}$:
    kedai itu telah menghabiskan **stok** mereka.

    (b) $EBMT_{o}$:
    kedai itu telah menghabiskan **saham** mereka.

English Substring	Candidate Malay Substring	$CSim_p$	$CSim_d$	$CSim_c$
stock	keturunan	0.2824	0.2726	0.2786
stock	saham	0.5291	0.2057	0.5642
† stock	† stok	0.7217	0.7978	0.3354