Chapter 2

Literature Study

This chapter gives details about the information acquired from studying literatures related to this work. Firstly, a summary about WordNet is described. Then, some studies about Latent Semantic Analysis (LSA) and its mathematical foundation, Singular Value Decomposition (SVD) are followed. In the last section, a brief review about automatic sense disambiguation is given.

2.1 WordNet

WordNet is a free lexical resource widely used in a variety of ways and areas, especially in the areas of linguistics, natural language processing, and artificial intelligence. The following sections describe the history, the design, the contents, the relations, and the applications of WordNet consecutively.

2.1.1 History of WordNet

In the past decades, linguists and psycholinguists have rediscovered the importance of lexicon or dictionary. They discovered lexicon as a highly structured repository of rules and principles. Lexicons are now perceived as an essential component of grammar rather than a wastebasket of insignificant, irregular, and inelegant facts about language. That is, lexicons are considered to contain information of words which specifies and determines much of their syntactic behaviour. (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990)

Traditional alphabetical lexicons or dictionaries organise lexical knowledge by arranging words according to their morphological similarities. Such an arrangement requires someone to trace through some alphabetical list in order to find information about a certain word. Nevertheless, doing this action repetitively can be tedious and time consuming. One solution is by providing online dictionaries, i.e. machine-readable lexical databases, so that computers can do the searching in place of humans. Computers can search much faster than humans and keep the history of what has been done recently.

Primarily, WordNet (Fellbaum, 1998) was conceived as a kind of online dictionary browser, particularly a lexical database to provide online searching dictionaries conceptually rather than merely alphabetically. Since 1978, George A. Miller has put forward ideas culminated in WordNet. Yet, it was not until 1985 that the WordNet project was seriously undertaken by a group of psycholinguists and linguists at Princeton University. At that time, WordNet was built based on the discovery of many properties of a lexicon, i.e. the information a lexicon must contain. As the work proceeded, WordNet was proposed to gain more effective combination of traditional lexicographic information and modern high speed computation.

In the early days, WordNet was not intended to be built as a complete lexicon yet, but as a test bed for a network model of lexical knowledge organisation. In this network model, the nodes represent the word meanings and the arrows represent the relations between the meanings. For example, consider the statement *"poodle is a kind of dog". Poodle* and *dog* are two nodes associated by the semantic relation *is a* (*kind of*) which conspicuously suggests that the meaning of *dog* is a component of the meaning of *poodle*. Being able to be extended for larger vocabulary, this kind of semantic relation turned into one of the basic semantic relations in WordNet.

As specific contents were needed to fulfil the network model tests, WordNet had to be manually constructed. Manual construction is certainly slow, expensive, and complex. Also, hand-crafted lexicons are likely to be too small to be used in natural language processing applications. However, a hand-crafted lexicon gives advantages by allowing someone to create entries with the kinds of contents that are expected to be useful for certain applications. These contents may be richer than the information extracted from standard lexicons. Moreover, the format of entries can be controlled, so that the information extraction will require minimum manipulation.

From time to time, WordNet has evolved into a self-contained lexical database. In June 1991, WordNet version 1.0 was released. Until now, WordNet has been updated to version 3.0.

2.1.2 Design of WordNet

Words express concepts of the world. Those concepts, which can be expressed by words, are called lexicalised concepts. The lexicalised concepts are described specifically by means of word meanings. WordNet attempts to organise lexical knowledge in terms of word meaning rather than the word itself. It makes use of a lexical matrix to organise the word meanings and represent their mapping onto the corresponding words (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990).

The columns of the lexical matrix denote the words and the rows denote word meanings referring to the lexicalised concepts. Each entry in a cell (i, j) therefore signifies that the meaning of row i can be expressed by the word j. The mappings between words and word meanings are many-to-many. That is, a column may be filled by several entries indicating that the word is polysemous. Conversely, a row may also be filled by several entries indicating that the words are synonyms. In other words, some words may have several different meanings and some meanings can be expressed by several different words.

WordNet takes advantage of synonymy relations to distinguish word meanings, i.e. the definition of lexicalised concepts. Having acquired a concept, someone can identify the concept by merely comparing it to a synonym. For example, the word *stone* can be signified as '*material consisting of the aggregate of minerals like those making up the Earth's crust'* or *'as the hard inner (usually woody) layer of the pericarp of some fruits (as peaches or plums or cherries or olives)'*. Given only the word *rock* or *pit* as the synonym of *stone*, someone, who has already known both of the meanings, will find no difficulty to choose the appropriate sense. That is to say the synonym sets {*stone, rock*} and {*stone, pit*} is adequate to show the two different meanings of *stone*. Generally, the synonym sets, or simply *synsets*, can be used to identify the word meaning. Each *synset* is associated with a gloss, which is a textual description about its meaning.

Since words can be taken from texts or discourses and *synsets* can be used to represent the word meanings, a lexical matrix can be represented by a mapping between words and *synsets*. However, occasionally a proper synonym is not available. In the case where a *synset* contains only a single word, its gloss can assist to differentiate the senses.

In the process of constructing the network of words and concepts, it was found necessary to assume concepts that are not lexicalised in English. For example, the phrase *hair care* has hypernym *care* and such hyponyms as *comb*, and *shampoo*. The concept *hair care* connects the concept of *care* with that of *brush*. Similarly, it connects the concept of *care* with that of *shampoo*. Without the phrase *hair care,* these hyponyms would be improperly associated to *nursing*, *maternalism*, and *nourishing*, which are some hyponyms of *care*. However, many of the concepts which are not lexicalised in English are lexicalised in other languages.

Explaining a number of concepts that are not lexicalised in English, WordNet differs from thesauri which only explain lexicalised concepts. Since the lexicalised concepts are paraphrased in the form of short phrases, WordNet also differs from standard dictionaries which only contain words of single form.

Although WordNet is neither a standard dictionary nor a thesaurus, it combines some characteristics of both. Dictionaries provide information about words and facilitate someone to understand the concepts behind unfamiliar words he has encountered. On the other hand, thesauri, which are built around concepts, help users to find the right word of a concept in mind.

WordNet resembles a thesaurus by representing concepts in the form of *synsets*. Since a *synset* in WordNet contains all the words expressing a given concepts, someone with a concept in mind can find other words in the same concept. WordNet lists concepts in the form of *synsets* and connects them through many kinds of semantic relations, such as hyponymy, meronymy, and entailment. Another difference from thesauri is that WordNet explicitly enumerates and labels the relations between concepts and words. Hence, someone can select a relation which subsequently directs him from one concept to another.

WordNet resembles a dictionary by providing definitions and sample sentences for most of its *synsets*. Expressing the meaning of a concept, a definition is valid for all the synonyms in a *synset*. Nevertheless, the sample sentences may not fit all synonyms. Thus, different sample sentences are frequently given for different members of a *synset*. Another similarity is that WordNet also contains information about morphologically related words. For example, the connection between the verb *qualify* and the corresponding noun *qualification* exists only for these two word forms. WordNet does not relate *qualify* to *making*, a synonym of *qualification*.

WordNet differs from a standard dictionary in the way that WordNet divides the lexicon into five syntactic categories, which are nouns, verbs, adjectives, adverbs, and function words. However, the function words are omitted regarding the assumption that they are likely to be stored separately as part of the syntactic component of language. The division into these four categories entails that WordNet contains no information about the syntagmatic properties of words. Also, it causes redundancy by storing the same word in more than one category. But, the advantage is that basic differences in the semantic organisation of these syntactic categories can be seen clearly and systematically exploited.

2.1.3 Contents of WordNet

Generally, lexicons are viewed as the repository of word knowledge and encyclopaedias as the repository of world knowledge. Although the boundaries between them are not crystal clear, understanding the meaning and uses of words certainly requires both kinds of knowledge. Hence, lexicons should contain both lexical knowledge and encyclopaedic knowledge to yield successful applications.

WordNet does not include encyclopaedic knowledge, although the definitions of its synonym sets (*synsets*) provide information about concepts that are not strictly part of their lexical structure. In the beginning, WordNet *synsets* were intended to contain no information but pointers to other *synsets.* Afterwards, it was found that the definitions and illustrative sentences were needed to discern closely related *synsets* whose members were polysemous. For many technical concepts, such as unusual plants and animals, lexical and encyclopaedic knowledge are combined together in the definitions to constitute the knowledge.

WordNet purely assigns the word as the basic lexical unit. It does not either decompose words into smaller meaningful units or contain structural units larger than words, such as scripts or frames which have been proposed as building blocks for lexicons. The words were primarily obtained from Kučera and Francis's *Standard Corpus of Present Day Edited English*, familiarly known as the *Brown Corpus.* In late 1986, a list of words given by Fred Chang at the *Naval Personnel Research and Development Center* was added. In 1993, a list of words in *COMLEX*, a common lexicon built by Ralph Grishman and his colleagues, was also added. Further, WordNet also absorbed other various sources.

The synonyms of WordNet were obtained from various thesauri, including *Basic Book of Synonyms and Antonyms* written by Laurence Urdang, *The Synonym Finder* written by Rodale and revised by Urdang, and *Roget’s International Thesaurus 4th edition* by Robert Chapman. Essentially, the *synsets* of WordNet are divided into five types, which are noun *synsets*, verb *synsets*, adjective *synsets*, adjective satellite *synsets*, and adverb *synsets*. An adjective satellite *synset* represents a concept that is similar in meaning to the concept represented by its head *synset*. Head *synsets* are several *synsets* of an adjective *synset* cluster organizing antonymous pairs or triplets, which have direct antonyms, i.e. pairs of words with associative bond resulting from their frequent co-occurrence.

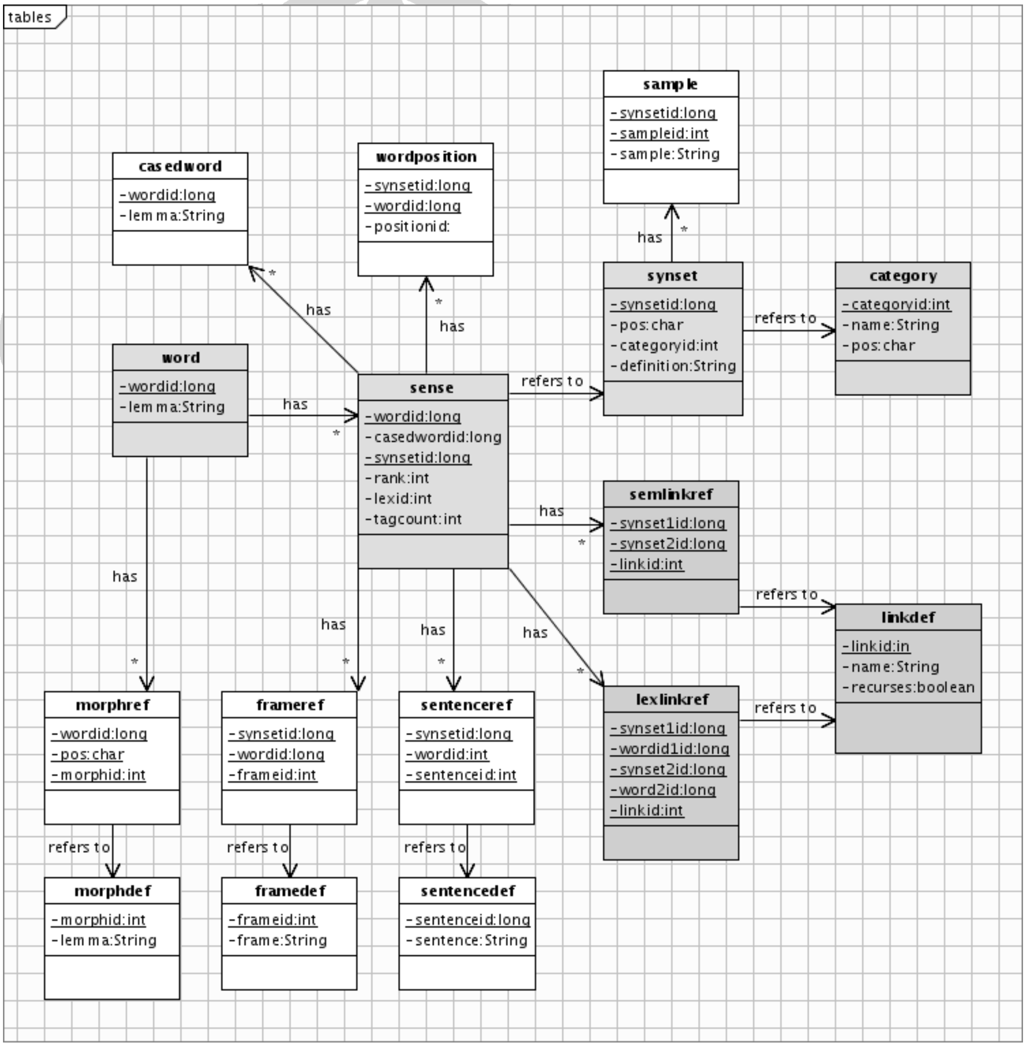

Figure 2.1 WordNet SQL version 3.0 Scheme (Bou, 2007)

The source files of WordNet written by lexicographers are products of lexical semantics analysis constructed by a variety of lexical and semantic relations used to represent the lexical knowledge organisation (Princeton University, 2006). The lexicographers frequently make addition and changes in the lexical source files. Periodically, *Grinder*, a significant program in Wordnet, converts these files into a lexical database.

Originally, the lexical database of WordNet was created in the form of a Prolog database. Another version of the database was generated in SQL form, namely for MySQL database. The Figure 2.1 above depicts the scheme of WordNet SQL version 3.0 consisting of 147306 words and 117659 *synsets*. Specifically, there are 117798 nouns, 11529 verbs, 23492 adjectives (8324 adjectives and 15168 adjective satellites) and 4481 adverbs. Since a word may have more than one syntactic category, the total is greater than 147306. With respect to the type of *synset*, there are 82115 noun *synsets*, 13767 verb *synsets*, 7463 adjective *synsets*, 10693 adjective satellite *synsets* and 3621 adverb *synsets*.

2.1.4 Relations in WordNet

WordNet connects words and concepts through a variety of semantic relations, i.e. relations between meanings, based on similarity and contrast. It focuses on the semantics of words and concepts rather than on semantics of text or discourse. Thus, it contains no relation indicating relationship among words in a topic of text. For example, WordNet does not connect doctor with hospital. This particular situation is referred to as *tennis problem* (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990).

On the other hand, frame semantics represents lexical semantics by describing a particular type of situation, object, or event (Ruppenhofer, Ellsworth, Petruck, Johnson, & Scheffczyk, 2006). For example, the *Education_teaching* frame describes a common situation involving frame elements (concepts) such as *Institution*, *Material*, *Precept*, *Qualification*, *Skill*, *Student*, *Subject*, and *Teacher*, as well as the existing relationships between them. The frame is produced by lexical units, which are pairs of words and its meaning, that may contain different syntactic categories, e.g. *coach* (*verb*), *educate* (*verb*), *education* (*noun*), *educational* (*adjective*), *graduate* (*noun*), *instruct* (*verb*), *instruction* (*noun*), *learn* (*verb*), *lecturer (noun)*, *master* (*verb*), *professor* (*noun*), *pupil* (*noun*), *school* (*verb*), *schoolmaster* (*noun*), *schoolmistresss* (*noun*), *schoolteacher* (*noun*), *student* (*noun*), *study* (*verb*), *teach* (*verb*), *teacher* (*noun*), *train* (*verb*), *training* (*noun*), *tutee* (*noun*), *tutor* (*noun*), *tutor* (*verb*).[1]

WordNet excludes framelike semantics by separating the words according to their syntactic category. Yet, some structure of frame semantics may be captured by the relational semantics in WordNet, namely the relation between words in the same category. For example, *instruct* is connected to *educate* by the hypernymy relation.

WordNet concerns not only the pattern of semantic relations between concepts, but also that of lexical relations between individual words. WordNet clearly separates the conceptual and lexical levels which are reflected in the semantic relations and lexical relations. Since concepts are represented by *synsets*, the semantic relations can be considered as pointers between *synsets*. The relations in WordNet include synonymy, antonymy, hyponymy/hypernymy, meronymy/holonymy, entailment, morphological relations, etc. The following are explanations of several kinds of relations used in WordNet.

Synonymy is a lexical relation between words with the same meaning. It is the most important relation in WordNet regarding the representation of word meanings. Two expressions are said to be synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made. Since true synonyms are rare, WordNet uses the weakened version of synonymy relation that is two expressions are synonymous in a linguistic context if the substitution of one for the other *in the context* does not alter the truth values. The definition of synonymy in terms of substitutability makes the division of words according to their syntactic category in WordNet is essential. That is to say, since the synonyms must be interchangeable, words in different syntactic categories cannot be synonyms, i.e. cannot form *synsets*, due to their noninterchangeability.

Antonymy is a lexical relation which connects words with their semantic oppositions. Note that the antonymy of a word *x* is not always *not-x*. For example, white and black are antonyms, but something which is not white is not necessarily black. Rather than a semantic relation between word meanings, antonymy is a kind of lexical relation between words. For example, the *synsets* {*rise, ascend*} and {*fall, descend*} may be conceptual opposites, but they are not antonyms. *Rise* and *fall* are antonyms and so are *ascend* and

[1] This example was taken from http://framenet.icsi.berkeley.edu.

descend. But, *rise* and *descend* or *ascend* and *fall* do not appear like antonyms. Thus, semantic relations between words and semantic relations between word meanings must be distinguished.

Hyponymy/hypernymy is a semantic relation between word meanings or semantic concepts. The semantic concept represented by the *synset* {x_1,x_2, …x_n} is said to be a hyponym of the semantic concept represented by the *synset* {y_1,y_2, …y_n} when the word *x* is a kind of the word *y*. For example, {*baby, babe, infant*} is a hyponym of {*child, kid*}, i.e. {*child, kid*} is a hypernym of {*baby, babe, infant*}, because *baby, babe* and *infant* are a kind of *child* as well as that of *kid*.

Meronymy/holonymy is also a semantic relation between semantic concepts. The semantic concept represented by the *synset* {x_1,x_2, …x_n} is said to be a meronym of the semantic concept represented by the *synset* {y_1,y_2, …y_n} when *y* has *x* as a part. For example, {*person, individual, someone, somebody, mortal, soul*} has part, i.e. is a meronym of, {*personality*} and is a part of, i.e. is a holonym of, {*people*}.

Morphological relations in WordNet are kinds of semantic relations due to the polysemous characteristic of words. They connect *synsets* in terms of morphology and cover both inflectional and derivational morphology. Earlier, a program to prune suffixes and transform words into their base form, called *Morphy*, was developed. Hence, WordNet, which contains only base forms, is able to recognise words modified by inflectional morphology. For example, WordNet is able to refer input *persons* to *person* with respect to plural suffix rule. Later, the derivational morphology was composed in the morphological relations. To some extent, it is more complex than the inflectional morphology. In WordNet, the derivational morphology relations include connecting deadjectival nouns to their root adjective, deadjectival adverbs to their root adjective, denominal adjectives to their root nouns, deverbal nouns to their verbs, denominal verbs to their nouns, etc (Harabagiu, Miller, & Moldovan, 1999).

2.1.5 Applications of WordNet

WordNet has been applied widely in a variety of ways and areas, especially in the areas of linguistics, natural language processing, and artificial intelligence. Frequently, WordNet has served as a valuable resource for various approaches of word sense disambiguation projects. Using lexical relations of WordNet as a knowledge base, (Leacock, Miller, & Chodorow, 1998) suggested an improvement on corpus-based word sense disambiguation with knowledge-based technique. Taking advantage of noun classification of WordNet, (Agirre & Rigau, 1996) made an attempt to resolve lexical ambiguity of nouns using conceptual density.

In favour of information retrieval and extraction applications, WordNet has been exploited as a linguistic knowledge tool to represent and interpret the meaning of information and subsequently provide access to information efficiently. Also, it is commonly used for query expansion in order to improve the performance, enhance the efficiency and optimise the precision of Internet resource searching. As well as text retrieval, WordNet has been utilised as an aid in image, audio and video retrieval. More applications on WordNet are exposed in (Morato, Marzal, Llorens, & Moreiro, 2004).

Following the outstanding work of WordNet, WordNets for various languages have been built recently. EuroWordNet project (Vossen, 1997) was carried out to deliver WordNets for other European languages including English, Spanish, Dutch, Italian, Czech, Estonian, etc. A later work was BalkaNet project (Sofia, et al., 2002) resembling the EuroWordNet project to build WordNets for Balkan languages including Czech, Turkish, etc.

2.2 Singular Value Decomposition

Singular Value Decomposition (SVD) is matrix decomposition which provides the fundamental numerical analysis for LSA. The following sections describe the properties and the construction of SVD. Also, the SVD special property of providing optimal rank approximation is briefly explained.

2.2.1 Properties of SVD

SVD (Strang, 1993) is a powerful decomposition which enables factorization of any matrix into two orthogonal matrices and a diagonal matrix. Every matrix m by n can be factored into $A = U\Sigma V^T$, where U is an m by m orthogonal matrix, V is an n by n orthogonal matrix, and Σ is an m by n diagonal matrix. The diagonal entries of Σ are nonnegative. The positive ones are called the singular values of A. The columns of U are called left singular vectors of A and the columns of V are called the right singular vectors of A.

$$A_{mxn} = U_{mxm} \, \Sigma_{mxn} \, V_{nxn}^T$$

The rank of A is $r \leq \min(m, n)$. It represents the number of the singular values of A. The first r columns of U are orthonormal bases for the column space of A and the last $m - r$ columns are those for the left nullspace of A. The first r columns of V are orthonormal bases for the row space of A and the last $n - r$ columns are those for the nullspace of A.

If $m > n$ and $r = n$, the SVD of A looks like:

$$A = [u_1 \ldots u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & \end{bmatrix} [v_1 \ldots v_n]^T.$$

If $n > m$ and $r = m$, the SVD of A looks like:

$$A = [u_1 \ldots u_m] \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_m & \end{bmatrix} [v_1 \ldots v_n]^T.$$

2.2.2 Construction of SVD

SVD is closely related to eigenvalue decomposition, a decomposition which diagonalise a matrix by using a basis of linearly independent eigenvectors. Recall from (Strang, 1993) that the equation $Ax = \lambda x$ where x is an eigenvector and λ is an eigenvalue. Suppose A is an n by n matrix and S is a linearly independent eigenvectors matrix of A.

$$AS = A\,[x_1 \ldots x_n] = [\lambda_1 x_1 \ldots \lambda_n x_n]$$

$$\text{where } [\lambda_1 x_1 \ldots \lambda_n x_n] = [x_1 \ldots x_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = S\Lambda.$$

Therefore $AS = S\Lambda$. The equation $AS = S\Lambda$ can also be written $A = S\Lambda S^{-1}$ or $S^{-1}AS = \Lambda$, where Λ is the diagonal eigenvalues matrix of A.

In fact, not all matrices have eigenvalue decomposition. Only n by n matrices with n linearly independent eigenvectors can be diagonalised by eigenvalue decomposition. Remarkably, singular value decomposition is able to diagonalise any matrix whether it is square or rectangular.

To diagonalise an arbitrary matrix, SVD selects two certain bases U and V. The formula $A = U\Sigma V^T$ specifies that U and V must be orthogonal, i.e. $U^T U = I$ and $V^T V = I$. Note that when both of U and V are equal to S, V^{-1} must be equal to V^T. The condition $V^{-1} = V^T$ is satisfied when V is an orthogonal matrix. Since $U = V = S$, both of U and V must be orthogonal matrices.

To get U, multiply A times A^T.

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T$$

Suppose V is an orthogonal matrix, then $V^T V = I$. Thus,

$$AA^T = U\Sigma\Sigma^T U^T$$

$$AA^T = U\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{bmatrix} U^T$$

To get V, multiply A^T times A.

$$A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T$$

Suppose U is an orthogonal matrix, then $U^T U = I$. Thus,

$$A^T A = V\Sigma^T \Sigma V^T$$

$$A^T A = V\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix} V^T$$

Seeing that $AA^T = U\Sigma^2 U^T$ and $A^T A = V\Sigma^2 V^T$, AA^T and $A^T A$ are simply symmetric matrices. According to the spectral theorem, every symmetric matrix $A = A^{\mathrm{T}}$ can always be factorized into $A = Q\Lambda Q^{\mathrm{T}}$ with a real diagonal eigenvalues matrix Λ and an orthogonal matrix Q. Therefore U and V are indeed orthogonal matrices.

An orthogonal matrix consists of an orthonormal set of columns or rows. Since a symmetric matrix has orthogonal eigenvectors and they might be a multiple of their unit eigenvector (eigenvector with length one), the eigenvectors of a symmetric matrix can always be chosen orthonormal. Columns of orthogonal matrix U are orthonormal eigenvectors u_1 ... u_m that correspond to each eigenvalues of AA^T. Likewise, columns of orthogonal matrix V are orthonormal eigenvectors v_1 ... v_n that correspond to each eigenvalues of $A^T A$.

By orthogonality, $U^T = U^{-1}$ and $V^T = V^{-1}$. Therefore the diagonalisation of n by n matrix $A = S\Lambda S^{-1}$ can be written

$$AA^T = U\Lambda U^{\mathrm{T}} \text{ by replacing } S \text{ with } U\text{, and}$$

$$A^T A = V\Lambda \mathrm{V}^{\mathrm{T}} \text{ by replacing } S \text{ with } V.$$

As a result, $\Lambda = \Sigma^2$.

AA^T and $A^T A$ are positive semidefinite matrices which means that AA^T and $A^T A$ has nonnegative eigenvalues. The eigenvalues of AA^T are σ_1^2 ... σ_r^2 ... σ_m^2 and the eigenvalues of $A^T A$ are σ_1^2 ... σ_r^2 ... σ_n^2. Both AA^T and $A^T A$ have r identical positive eigenvalues and otherwise are zeros. Since zero is not considered as a singular value, the singular values of A are merely the square roots of those positive eigenvalues.

As the computation of U, V, and Σ have been introduced, the formula $A = U\Sigma V^T$ can now be constructed. The formula $A = U\Sigma V^T$ always holds. (Strang, 1993) explains it in two steps. The first step is to get the length of Av_i. The second step is to get the equation $Av_i = \sigma_i u_i$. Suppose u_i denotes column i of U, σ_i denotes diagonal element i of Σ, and v_i denotes column i of V.

The first step:

$$A^T Av_i = \sigma_i^2 v_i\text{, multiply each side with } v_i^T$$

$$(v_i^T A^T)Av_i = v_i^T \sigma_i^2 v_i$$

$$(Av_i)^T\, Av_i = v_i^T \sigma_i^2 v_i$$

Length of Av_i can be obtained by calculating its Frobenius or Euclidean norm, which is the square root of the sum of the square of its entries. Multiplication of $(Av_i)^T\, Av_i$ gives the sum of the square of Av_i's entries, thus, $(Av_i)^T\, Av_i = ||Av_i||^2$.

$$||A\ v_i||^2 = \sigma_i^2\text{, so that}$$

$$||Av_i|| = \sigma_i$$

The second step:

$$A^T A\, v_i = \sigma_i^2 v_i \text{ multiply each side with } A$$

$$(AA^T)(Av_i) = \sigma_i^2 Av_i$$

$$u_i = Av_i$$

To get the unit vector u_i, divide u_i by its length.

$$u_i = \frac{Av_i}{\sigma_i}$$

$$Av_i = \sigma_i u_i$$

To illustrate, suppose A is an m by n matrix with $m > n$ and $r = n$.

$$AV = A\,[v_1 \dots v_n] = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} [v_1 \dots v_n]$$

$$U\Sigma = [u_1 \dots u_n \dots u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & \\ & & \end{bmatrix}$$

$$AV = U\Sigma$$

$$A = U\Sigma V^{-1}$$

$$\text{since } V^T = V^{-1},\ A = U\Sigma V^T$$

$$A = [u_1 \dots u_n \dots u_m] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & \\ & & \end{bmatrix} [v_1 \dots v_n]^T$$

Typically, singular values are arranged in decreasing order. The corresponding eigenvectors u_i and v_i are required to follow the decreasing order as well.

2.2.3 Rank Approximation

The rank of a matrix is the number of its linearly independent columns. It is also the number of linearly independent rows. If an m by n matrix is not identically zero, its rank is at least one and at most min (m, n).

Suppose B is a matrix of rank one. If u is the single entry of a basis, then each column of B is a multiple of u. Suppose the v_i is the coefficient of column i of B, then $B = [v_1 u \ldots v_n u] = uv^T$. The SVD of any matrix A with rank r can be represented as a sum of matrices of rank one (Trefethen & Bau, 1997), that is

$$A = \sum \sigma_i u_i v_i^T$$

SVD has the special property of providing a smaller rank matrix with the optimal approximation to the original matrix. Suppose $A = A_k + E_k$ where A_k is the sum of the first k terms and E_k is the sum of the remaining terms. A_k represents a smaller rank matrix of A with rank $k \leq r$ and E_k represents the error matrix $A - A_k$.

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

$$E_k = \sum_{i=k+1}^{r} \sigma_i u_i v_i^T$$

A_k is the optimal rank k approximation to A when E_k has minimal length. (Golub & Kahan, 1965)

$$\|E_k\| = \|A - A_k\| = \|U\Sigma V^T - A_k\|$$

Taking into account the orthogonality of U and V which preserves their norms,

$$\|E_k\| = \|A - A_k\| = \|\Sigma - U^T A_k V\|$$

Let $U^T A_k V = M$, then $A_k = UMV^T$ where M is a diagonal matrix of eigenvalues of A_k.

$$\|E_k\| = \|A - A_k\| = \|\Sigma - M\|$$

$$||E_k|| = \sqrt{\sum_{i=1}^{n} (\sigma_i - m_i)^2}$$

$$||E_k|| = \sqrt{\sum_{i=1}^{r} (\sigma_i - m_i)^2 + \sum_{i=r+1}^{n} (\sigma_i - m_i)^2}$$

Since $\sigma_i = m_i$ and both $\sigma_{r+1} \ldots \sigma_n$ and $m_{k+1} \ldots m_n$ are zeros, then

$$||E_k|| = \sqrt{\sum_{i=k+1}^{r} {\sigma_i}^2}$$

$||E_k||$ is minimised when $k = r$. The approximation quality of the smaller rank matrix is given by the ratio of the error length and the original matrix length (Kalman, 1996).

$$\frac{||E_k||}{||A||} = \sqrt{\frac{\sum_{i=k+1}^{r} {\sigma_i}^2}{\sum_{i=1}^{r} {\sigma_i}^2}}$$

In addition, the relative error for a sum of the first k terms $e(k)$ is defined as

$$e(k) = 1 - \frac{||E_k||}{||A||}$$

The optimal rank approximation property of SVD is very important for LSA computation. It facilitates LSA to extract information about term and document vectors in a semantic space and represent it in a smaller semantic space. More about LSA is explained in the next section.

2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing contextual-usage meaning of words by statistical computation applied to a large corpus of text (Landauer, Foltz, & Laham, 1998). Furthermore, LSA is a convenient method to obtain approximate similarities of word meanings. The approximate similarities depend on powerful mathematical analysis, which is able to infer deep word-word relations, word-document relations, and document-document relations.

2.3.1 LSA as a Theory of Meaning

Every human grows up surrounded by language. Naturally, humans can easily learn to understand utterances of their indigenous language without being explicitly taught about word meanings or language rules. Accordingly, there ought to be a method which allows human minds to learn any language by simply immersing themselves into a language milieu. (Landauer, McNamara, Dennis, & Kintsch, 2007)

LSA is a computational method which attempts to learn language and then use it properly. Its main emphasis is to construct word and document meanings from experiencing language. For many years, philosophers have believed that computers processing only a sample of natural language would not be able to understand verbal meaning as humans do. LSA is a breakthrough that an algorithm with simple concept applied to text can learn language and is capable of matching literate humans in doing tasks related to understanding word and document meanings.

Meaning declares the truth of matters, basic and essential properties of objects and happenings, mental and emotional constructions, etc. It can also be considered as an abstraction of existing concepts or properties regardless of language. In other words, LSA cannot create word meanings. The meaning must exist first and be derived from already meaningful primitives in perception or action.

In view of the fact that strings of words must be able to represent and express information about both human physical world and mental world, much perceptual experience should be able to be represented in linguistic expressions. Conversely, many linguistic expressions should also be able to be represented in perceptual experience. Once the representations have been obtained through the evolution of language, most of the knowledge of meaning is likely to be able to be learnt by exploring the language.

Formerly, knowledge of verbal meanings was interpreted in the form of rules, descriptions, and variables, such as part of speech, grammars, etc. This kind of interpretation could only be applied by human intervention. However, LSA interpretation only needs typical text of natural human language.

LSA derives knowledge of meaning by analysing a large corpus of text in which the words reflect human knowledge and experiences. A corpus may contain collections of texts with related semantic meaning. Each text in a collection is called an episode. LSA assumes its knowledge of meaning of documents as abstractions of episodes. Its knowledge of word meanings is closely linked and mutually dependent with its knowledge of episodes.

LSA's knowledge of the meaning of a word can be considered as a kind of average of the meaning of all the documents in which it appears, and the meaning of a document as a kind of average of the meaning of all the words it contains. The ability of LSA to concurrently derive this knowledge depends on SVD as its mathematical foundation.

According to LSA, verbal meanings are almost entirely created by relations of words and collections of words. Humans often first discover relations of words and documents to each other by connecting the perceptual experience with verbal experiences. For example, someone who reads "Jakarta is the capital city of Indonesia and Batavia is the former name of Jakarta" may conclude that "Batavia is the former name of the capital city of Indonesia". In the same way, LSA infers indirect knowledge by observing the relations among words. LSA assumes that words do not have their own meanings, nevertheless meaning of words can be inferred from their relations to each other.

Notice that LSA derives knowledge from relations of unitary expressions of meaning rather than relations of sequences of words. Not only neighbouring co-occurrences of words does LSA observe, but also detailed patterns of occurrences of words over numerous local meaning-bearing contexts. LSA observes that some words did occur in a particular document and some did not. It captures the relations of different word choice in different document meanings. That information about pattern of words usage makes LSA possible to define the documents and infer relations among words and documents.

Unfortunately, LSA is not a complete model of language due to the fact that it does not take into account the word order. On the other hand, word order may affect the meaning of sentences or the implication of sentences and paragraph order. Therefore, LSA does not cover all aspects of language. LSA often does not adequately represent variability of meanings conveyed by prediction, anaphora, metaphor, modifications, etc.

2.3.2 Similarity Approximation of LSA

Knowledge of word and document meanings derived by LSA resembles a variety of human cognitive phenomena, one of which is the way human representation of meaning reflects human knowledge and experiences. Since this meaning is reflected in word choice, LSA can approximate human judgements of meaning similarity between words and objectively predict the implication of overall word-based similarity between documents.

Co-occurrence of two words in the same document does not determine the similarity between those words, although they might be related to the same topic, supporting to produce the whole meaning of the document. In addition, the number or proportion of words shared between two documents does not determine their similarity as well. Using SVD and dimension reduction, LSA measures the similarity from the effect of the words wherever they occur in meaningful documents. It learns about the meaning of a word from the composition of documents in which the word occurs and the composition of other documents in which the word does not occur.

2.3.3 Mathematical Properties of LSA

LSA is closely related to the concept of vector space model in linear algebra. It represents terms and documents of a corpus in a high dimensional term-document semantic space. The term and document definitions are customisable depending on the application. Terms can be words, phrases, concepts, etc. On the other hand, documents can be sentences, paragraphs, collection of paragraphs, etc.

The following is a small example which demonstrates the computation of LSA. The text documents are taken from a small collection of documents in Table 2.1. The collection of documents consists of five article titles related to Internet and four article titles related to baking. The articles related to internet are labelled I1 – I5 and the articles related to baking are labelled B1 –B4. Suppose the articles only consist of the italicised-bold keywords and each article has at most a single occurrence of a keyword.

Table 2.1 Example of Data: Titles for Topics on Internet and Baking

Label	Titles
I1	***Cookies*** Setting for Mozilla ***Web Browser***
I2	***Internet Browser*** Software Review 2008
I3	***Network*** Services Provided by ***Internet Web Server***
I4	***Internet Network*** Security Issue
I5	***Techniques*** for Building Client ***Server*** Applications
B1	Easy ***Chocolate Cake Recipe***
B2	***Chocolate Baking*** Tips
B3	***Cake*** Decorating ***Techniques*** for Beginners
B4	***Recipe*** Collections for ***Baking Cookies***

At first, a term-document matrix must be constructed. Each row of the matrix denotes a unique term vector and each column denotes a document vector. Then, each cell denotes the occurrence frequency of a term in a particular document. Usually, a weighting function is then applied to each cell in order to express the importance of a term in a particular document. Since containing many more zero entries than nonzero entries, a term-document matrix is typically considered sparse.

Table 2.2 shows the term-document matrix in which the occurrence frequency of each term corresponding to each document has been listed. Assume that the terms have equal importance, so that the weighting function is not necessarily applied.

Table 2.2 Term-Document Matrix with Term Frequencies Corresponding to the Titles in Table 2.1

Term	I1	I2	I3	I4	I5	B1	B2	B3	B4
Baking	0	0	0	0	0	**0**	1	0	1
Browser	1	1	0	0	0	0	0	0	0
Cake	0	0	0	0	0	1	0	1	0
Cookies	1	0	0	0	0	0	0	0	1
Internet	0	1	1	1	0	0	0	0	0
Network	0	0	1	1	0	0	0	0	0
Recipe	0	0	0	0	0	1	0	0	1
Server	0	0	1	0	1	0	0	0	0
Techniques	0	0	0	0	1	0	0	**1**	0
Chocolate	0	0	0	0	0	1	1	0	0
Web	1	0	1	0	0	0	0	0	0

The term-document matrix is transformed into term-document semantic space by singular value decomposition (SVD). SVD decomposes the term-document matrix into matrix U which describes the original term vectors, matrix V which describes the original document, and diagonal matrix Σ of the singular values. Table 2.3 shows the SVD of the term-document matrix in Table 2.2.

Finally, the result of LSA is obtained by reconstructing the term-document matrix with smaller dimension. The reconstruction uses only a number of first columns of the three matrices of SVD. In this example, the reconstruction use the first 2 shaded columns of matrix U, V, and Σ in Table 2.2. The result is a two dimensional reconstructed term-document matrix shown in Table 2.4.

Table 2.3 SVD of Term-Document Matrix Represented in Table 2.2

Matrix U of Term Vectors											
Baking	-0.08	-0.44	0.15	0.23	-0.44	0.47	0.01	-0.29	0.41	-0.19	-0.14
Browser	-0.28	-0.05	0.38	-0.32	0.49	0.38	-0.11	0.22	-0.10	-0.38	-0.29
Cake	-0.05	-0.36	-0.44	-0.16	0.46	-0.24	0.09	-0.26	0.35	0.15	-0.40
Cookies	-0.20	-0.34	0.45	-0.30	-0.17	-0.14	0.25	-0.16	-0.34	0.53	-0.11
Internet	-0.58	0.18	-0.06	0.33	0.25	0.29	0.29	0.10	0.20	0.38	0.29
Network	-0.45	0.16	-0.15	0.33	-0.11	-0.22	0.15	-0.37	-0.44	-0.38	-0.29
Recipe	-0.09	-0.53	-0.02	0.13	-0.01	-0.33	0.36	0.52	-0.01	-0.34	0.25
Server	-0.36	0.09	-0.31	-0.21	-0.46	-0.01	-0.23	0.52	0.08	0.15	-0.40
Techniques	-0.09	-0.09	-0.48	-0.55	-0.12	0.35	0.19	-0.19	-0.23	-0.15	0.40
Chocolate	-0.05	-0.45	-0.19	0.31	0.16	0.15	-0.62	0.03	-0.41	0.19	0.14
Web	-0.43	0.01	0.20	-0.24	-0.05	-0.41	-0.45	-0.23	0.33	-0.15	0.40

Matrix Σ of Singular Values									
	2.52	0	0	0	0	0	0	0	0
	0	2.23	0	0	0	0	0	0	0
	0	0	1.89	0	0	0	0	0	0
	0	0	0	1.63	0	0	0	0	0
	0	0	0	0	1.43	0	0	0	0
	0	0	0	0	0	1.14	0	0	0
	0	0	0	0	0	0	1.12	0	0
	0	0	0	0	0	0	0	0.82	0
	0	0	0	0	0	0	0	0	0.39
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Matrix V of Document Vectors									
I1	-0.36	-0.17	0.55	-0.52	0.18	-0.15	-0.28	-0.21	-0.30
I2	-0.34	0.06	0.17	0.01	0.52	0.58	0.16	0.39	0.26
I3	-0.72	0.20	-0.17	0.13	-0.26	-0.31	-0.22	0.02	0.43
I4	-0.41	0.15	-0.11	0.40	0.10	0.06	0.39	-0.33	-0.60
I5	-0.18	0.00	-0.42	-0.46	-0.41	0.30	-0.04	0.40	-0.40
B1	-0.08	-0.60	-0.35	0.18	0.42	-0.37	-0.15	0.34	-0.16
B2	-0.05	-0.40	-0.02	0.33	-0.20	0.54	-0.54	-0.32	0.00
B3	-0.06	-0.20	-0.49	-0.43	0.23	0.10	0.25	-0.56	0.30
B4	-0.14	-0.59	0.31	0.04	-0.43	-0.01	0.56	0.08	0.16

LSA takes advantage of SVD's special property of giving an optimal reduced rank approximation matrix. By reducing dimensions of the term-document matrix, irrelevant information and variability of the term choice associated with the document, referred to as noise, is removed. Polysemy and synonymy in documents are instances of such noise. By reducing dimensionality, LSA combines surface information into deeper abstraction which captures mutual implicit relation of terms and documents.

The optimal dimensionality gives correct induction of implicit relations. However, finding the optimal dimensionality which matches the human term and document meanings is still an empirical issue. With too few dimension, estimated similarity of different meaning of terms or documents might be too high. Conversely, with too many dimensions, similar terms or documents can be considered as different.

Table 2.4 Two-Dimensional Reconstructed Term-Document Matrix Represented in Table 2.2 based on Two Shaded columns of SVD matrices

Term	I1	I2	I3	I4	I5	B1	B2	B3	B4
Baking	0.24	0.01	-0.05	-0.07	0.04	**0.61**	0.40	0.21	0.61
Browser	0.27	0.23	0.49	0.27	0.13	0.12	0.08	0.06	0.17
Cake	0.18	0.00	-0.06	-0.07	0.03	0.49	0.33	0.17	0.49
Cookies	0.31	0.13	0.22	0.09	0.09	0.49	0.33	0.18	0.52
Internet	0.46	0.53	1.14	0.67	0.26	-0.13	-0.09	0.00	-0.03
Network	0.35	0.41	0.88	0.52	0.20	-0.12	-0.08	0.00	-0.04
Recipe	0.28	0.00	-0.07	-0.09	0.04	0.73	0.49	0.25	0.73
Server	0.29	0.32	0.69	0.40	0.16	-0.05	-0.03	0.01	0.02
Techniques	0.12	0.07	0.13	0.07	0.04	0.14	0.09	**0.06**	0.15
Chocolate	0.21	-0.02	-0.10	-0.10	0.02	0.61	0.40	0.21	0.61
Web	0.38	0.37	0.79	0.45	0.19	0.07	0.04	0.06	0.14

The immense importance of dimension reduction in LSA can be observed by comparing the term-document matrix before and after applying dimension reduction. Before reduction, there were a lot of zero points in the term-document matrix. However, the dimension reduction changes the values of each cell, so that the reconstructed term-document matrix consists of estimated similarity between terms and between documents. By dimension reduction, LSA estimates greater or lesser frequency for terms that did occur in some documents. In addition, the terms that did not occur in some documents might be estimated to occur in those documents.

Look at the cells for *baking* in column B1 and *techniques* in column B3 in both Table 2.2 and Table 2.4. The term *baking* did not occur in B1 of the original term-document matrix, thus its frequency was zero in B1. But, it is replaced with 0.61 in the two-dimensional reconstructed term-document matrix. This indicates that LSA estimates the term *baking* to occur 0.61 times in B1 or any other documents containing the terms *cake*, *recipe*, and *chocolate*. LSA states that the term *baking* describe an associated context with the terms *cake*, *recipe*, and *chocolate*.

On the contrary, the frequency of term *techniques*, which was 1 in B3, has been replaced with 0.06. This indicates that the term *techniques*, which originally occurred in B3, is unexpected in this context. LSA states that the term *techniques* is unimportant to describe the document.

Table 2.5 Correlation Coefficient between Titles in the Original Term-Document Matrix

	I1	I2	I3	I4	I5	B1	B2	B3	B4
I1	1.00	0.24	-0.04	-0.29	-0.29	-0.38	-0.29	-0.29	0.08
I2	0.24	1.00	0.13	0.39	-0.22	-0.29	-0.22	-0.22	-0.29
I3	-0.04	0.13	1.00	0.62	0.13	-0.46	-0.36	-0.36	-0.46
I4	-0.29	0.39	0.62	1.00	-0.22	-0.29	-0.22	-0.22	-0.29
I5	-0.29	**-0.22**	0.13	-0.22	1.00	-0.29	-0.22	0.39	-0.29
B1	-0.38	**-0.29**	-0.46	-0.29	-0.29	1.00	0.24	0.24	0.08
B2	-0.29	-0.22	-0.36	-0.22	-0.22	0.24	1.00	-0.22	0.24
B3	-0.29	-0.22	-0.36	-0.22	0.39	0.24	-0.22	1.00	-0.29
B4	0.08	-0.29	-0.46	-0.29	-0.29	0.08	0.24	-0.29	1.00

The dimension reduction of LSA also changes the relations between documents. Table 2.5 shows that the correlations between the five internet-related documents are generally low. Likewise, the correlations between the four baking-related documents are also low.

Table 2.6 Correlation Coefficient between Titles in the Two-Dimensional Reconstructed Term-Document Matrix

	I1	I2	I3	I4	I5	B1	B2	B3	B4
I1	1.00	0.83	0.81	0.79	0.88	-0.51	-0.51	-0.43	-0.45
I2	0.83	1.00	1.00	1.00	1.00	-0.90	-0.90	-0.86	-0.87
I3	0.81	1.00	1.00	1.00	0.99	-0.91	-0.91	-0.88	-0.89
I4	0.79	1.00	1.00	1.00	0.99	-0.93	-0.93	-0.89	-0.90
I5	0.88	**1.00**	0.99	0.99	1.00	-0.86	-0.86	-0.81	-0.82
B1	-0.51	**-0.90**	-0.91	-0.93	-0.86	1.00	1.00	1.00	1.00
B2	-0.51	-0.90	-0.91	-0.93	-0.86	1.00	1.00	1.00	1.00
B3	-0.43	-0.86	-0.88	-0.89	-0.81	1.00	1.00	1.00	1.00
B4	-0.45	-0.87	-0.89	-0.90	-0.82	1.00	1.00	1.00	1.00

However, in the two-dimensional reconstructed term-document matrix shown in Table 2.6, the correlations between documents with similar topics are much higher than before. In contrast, the correlations between documents with different topics are much lower than before. For example, the correlation coefficient changes between I2 and I5. It increases greatly from -.22 to 1.00. Then, the correlation coefficient change between I2 and B1, it decreases greatly from -.29 to -.90.

The reduced dimensional semantic space is the foundation for the semantic structures. It uncovers the latent semantic structure in the pattern of term usage to define documents. As a result of reconstructing the approximate term-document matrix with reduced dimensional semantic space, LSA is able to capture the important latent semantic structure of terms and documents.

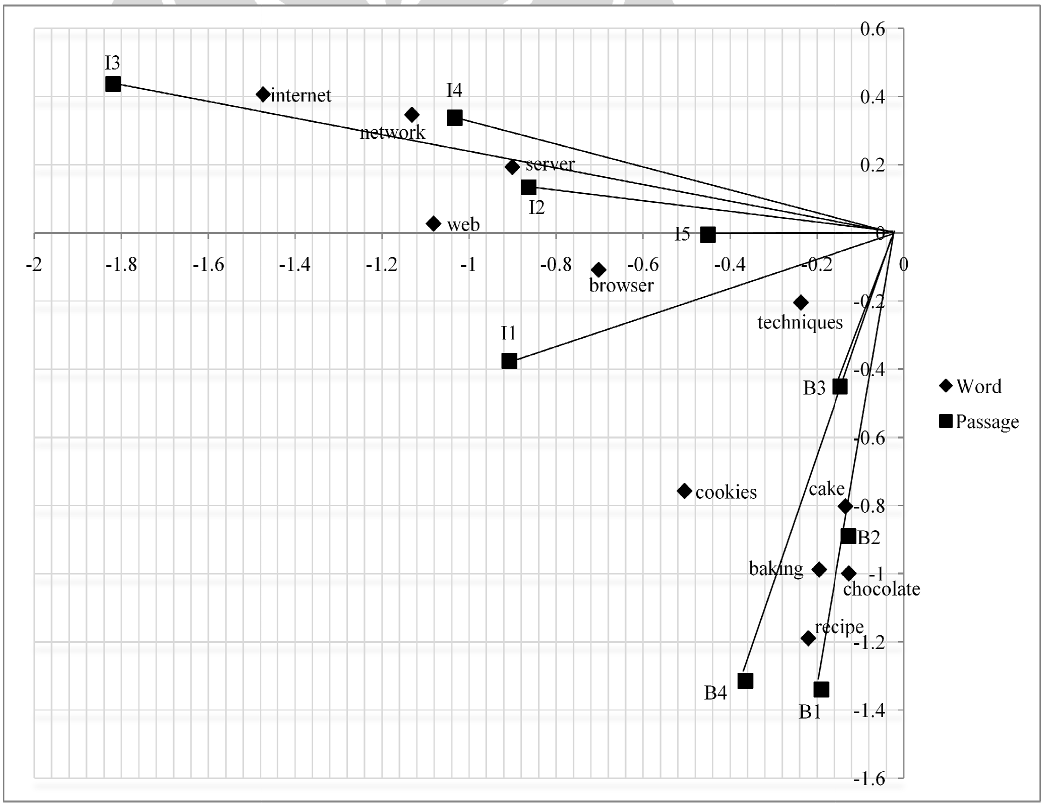

Figure 2.2 Two Dimensional LSA Semantic Space for Internet-Baking Titles Documents Showing Similarity of Meaning of the Documents

Look at Figure 2.2 and Figure 2.3 showing the two dimensional LSA semantic space for internet-baking titles documents. Each point represents a document or a term vector starting from the origin. The documents are denoted by the squares and the terms are denoted by the diamonds. The coordinate of a term point is defined by:

- = the first column (dimension) of matrix multiplied by the first singular value
- = the second column of matrix multiplied by the second singular value

Similarly, the coordinate of a document point is defined by:

- = the first column of matrix multiplied by the first singular value
- = the second column of matrix multiplied by the second singular value

Similarity between documents or terms is determined by the angle between vectors. The smaller the angle between two vectors is, the larger their similarity of meaning is. In Figure 2.2, the document B3 "*Cake Decorating Techniques for Beginners*" is the closest document to the document B4 "*Recipe Collections for Baking Cookies*" despite sharing no terms in general. Similarly in Figure 2.3, the terms *cake*, *baking*, *chocolate*, and *recipe* are very close to each other. The similarity between *cake* and *baking* is noteworthy since they have never occurred concurrently in a same document.

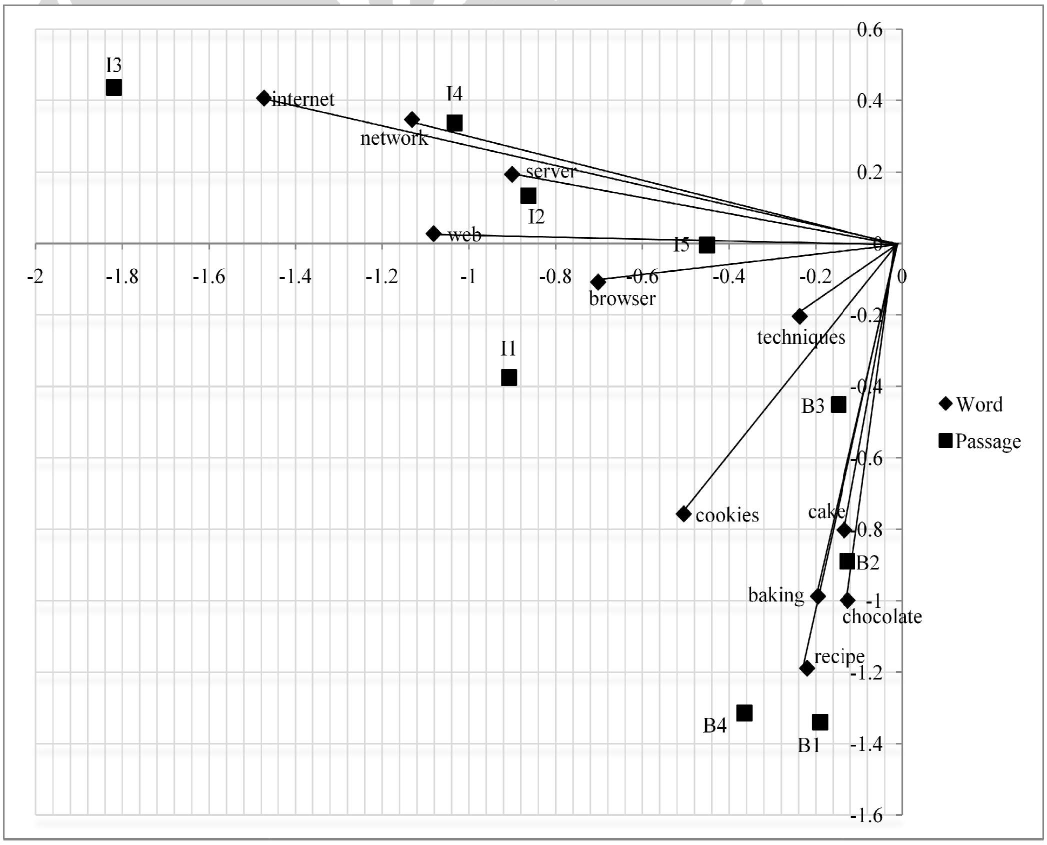

Figure 2.3 Two Dimensional LSA Semantic Space for Internet-Baking Titles Documents Showing Similarity of Meaning of the Terms

The term vector represents the average effects of a term on the meaning of documents in which it occurs. Also, it represents the results from effects of documents in which it does not occur. Thus, the similarity of terms simply depends on the effects they have on the documents in which they occur and do not occur. It does not simply depend on co-occurrence within a single document.

Even if the terms never occur concurrently in a document, term vectors with similar meaning are near each other in the reduced dimensional semantic space. Likewise, documents with similar conceptual meaning are near each other although they share no types in general. This may happen because the similarity of documents does not depend on the terms they contain, but on the semantic content.

2.3.4 Applications of LSA

LSA is a computational model which mimics human language usage. By experiencing language, LSA elucidates how term and document meanings can be constructed. Using a simple structure of verbal meaning constraint and a rough approximation of experience as humans, LSA is able to do meaning-based cognitive tasks as well as humans. Thus, LSA might be considered as a tool for measuring the corresponding human abilities.

For example, after self learning from a large body of representative text, LSA gets a good score on standardised multiple choice vocabulary tests for high school students. LSA has also been used to rate the adequacy of content of expository essays. LSA measured the effects on comprehension of paragraph to paragraph coherence better than human coding. LSA is also able to detect improvements in student knowledge from before to after reading as well as human judges. LSA improves information retrieval up to 30% by matching queries to documents of the similar meaning and reject irrelevant documents. More applications of LSA are reported in (Landauer, Foltz, & Laham, 1998).

2.4 Automatic Sense Disambiguation

Since the earliest days of computerised language processing, automatic disambiguation of word sense has been noticed and endeavoured. Sense disambiguation is necessary to accomplish most natural language processing tasks. It is essential for language understanding applications as well as for machine translation, information retrieval, speech and text processing, etc.

2.4.1 Word Sense Disambiguation

The term *sense* is regarded as *meaning*. In general, word sense disambiguation (WSD) involves the association of a given word in a text or discourse with a definition or meaning which is distinguishable from other meanings potentially attributable to the word (Ide & Veronis, 1998). In other words, sense of a word depends on the context in which the word appears. For example, the word *bank* expresses the sense "*financial institution*" in the context "*deposit some money to the bank*". On the other hands, it expresses the sense "*sloping land beside a body of water*" in the context "*go fishing by the river bank*".

Essentially, the task of WSD requires a list of senses for every word relevant to a given text or discourse. Much work on WSD relies on pre-defined list of senses. Usually, such a list is obtained from dictionaries or a group of features, categories, or associated words, e.g. synonyms in a thesaurus. However, in actual fact, the precise definition of a sense is a matter of considerable debate with respect to the degree of sense granularity and word usage.

The context of the target word, i.e. the word to be disambiguated, provides information used to disambiguate the target word sense. The context is used in two ways, namely the "bag of words" approach and the relational information approach. In the bag of words approach, the context is considered as some window of words surrounding the target word in a text or discourse. The surrounding words are regarded as a group without consideration of their relationship to the target word. Conversely, in the relational information approach, the context is considered in terms of some relation to the target word, including distance from the target words, syntactic relations, selectional restrictions, phrasal collocation, etc.

The assignment of words to senses also relies on external knowledge resources. Like the context, the external knowledge resources provide information to assign each instance of a word to the appropriate sense. The external knowledge resources include lexical and encyclopaedic resources. The lexical resources are generally regarded as the repository of word knowledge and the encyclopaedic resources as the repository of world knowledge.

The problem of WSD has been described as AI-complete, i.e. the problem can be solved after resolving all difficult problems in artificial intelligence (AI), such as the representation of common sense and encyclopaedic knowledge. All work on WSD has been accomplished in two ways, namely *knowledge-driven WSD* and *data-driven WSD*. *Knowledge-driven WSD* matches the context of a target word to the information from an external knowledge resource. On the other hand, *data-driven* or *corpus-based WSD* matches the context to information about contexts of previously disambiguated instances of the target word derived from corpora.

Firstly, WSD was attempted in the area of machine translation. In machine translation, WSD is essential for producing proper translation of words. For example, the English word *bank* can be translated to Indonesian words *bank* or *tepian sungai* depending on the context of the English word. Afterwards, work on WSD continued in AI-based natural language understanding research as well as in the fields of content analysis, stylistic and literary analysis, and information retrieval.

AI-based methods for WSD include symbolic methods and connectionist methods. The symbolic methods make use of semantic networks for representing word senses. On the other hand, the connectionist methods use semantic priming, that is, a process in which the introduction of a certain concept will influence and facilitate the introduction of other semantically related concepts. For example obtained from (Ide & Veronis, 1998), introduction of the concept *throw* will introduce the *"physical object"* sense of *ball*, which in turn would inhibit the introduction of the other senses of *ball* such as *"social event"*.

As large scale lexical resources became widely available, many efforts attempt to automatically extract knowledge from these resources, including machine readable dictionaries and thesauri, and construct large-scale knowledge bases, such as WordNet. In spite of the inconsistency in dictionaries and difficulty of automatic extraction, the machine readable dictionaries provide a ready-made source of information about word senses which has been rapidly used in much WSD research. Differing from dictionaries, thesauri provide information about relationship among words potentially valuable for language processing work. Roget's International Thesaurus, which has been used in a variety of applications including machine translation and information retrieval, suggests that each occurrence of the same word under different categories typically represents different senses of that word. However, Roget's and other thesauri have not been used extensively for WSD.

WordNet, a large-scale hand-built knowledge base, enumerates individual senses of words and defines *synsets* of synonymous words representing a single concept (see

Section 2.1). Due to its availability, WordNet is used largely for WSD as well as for other natural language processing tasks. However, the fine-grained sense distinction of WordNet senses has been cited as a hindrance for WSD (Ide & Veronis, 1998).

Empirical methods are commonly applied on corpora that provide a bank of samples prompting the development of numerical language models. As the study of corpus linguistic increased, larger corpora has been created and utilised by statistical methods. In the area of WSD, many empirical methods as well as corpus-based methods were developed to resolve the problem.

2.4.2 Cross Language Information Retrieval Using LSA

Cross Language Information Retrieval (CLIR) is a task of information retrieval which attempts to retrieve documents in any language in response to a given query in another language. Based on LSA (see Section 2.3), (Dumais, Landauer, & Littman, 1996) proposed cross-language latent semantic indexing (CL-LSI) as a fully automatic method for cross language document retrieval. CL-LSI was firstly experimented on French-English parallel texts. Afterwards, (Rehder, et al., 1997) experimented CL-LSI on larger document collections, much noiser data (no human translations), and more languages (English-French-German).

CL-LSI is applied on a multi-language $m \times n$ word-document matrix of n documents and m unique words. Specifically, in the experiments carried out by (Rehder, et al., 1997), the multi-language word-document matrix M is composed by word-document matrices of three different languages. Let E be the English word-document matrix of n English documents with m^E English words. Let F be the French word-document matrix of n semantically equivalent French documents with m^F French words. Let G be a German word-document matrix of n semantically German documents with m^G German words. The multi-language word-document matrix

$$M = \begin{bmatrix} E \\ F \\ G \end{bmatrix}$$

is an $(m^E + m^F + m^G) \times n$ matrix in which the column i is a vector representing English, French, German words appearing in the union of document i expressed in all three languages. Various weighting schemes can be applied to this matrix. Particularly, (Rehder, et al., 1997) used log-entropy weighting.

According to the LSA methodology, the word-document matrix M is decomposed by SVD into

$$M = \begin{bmatrix} U_k^E(M) \\ U_k^F(M) \\ U_k^G(M) \end{bmatrix} \cdot \Sigma_k(M) \cdot V_k(M)$$

where $U_k^E(M)$, $U_k^F(M)$, $U_k^G(M)$ are k-dimensional vector lexicons for English, French, and German respectively. A vector lexicon is a matrix containing vector representations of each word of the corresponding language. It gives definition, i.e. the result of the numerical analysis, for each word of the corresponding language.

LSA creates a reduced dimension space in which words that occur in similar contexts are near each other. Without necessarily using any external knowledge resources, LSA is able to discover important underlying word associations by means of SVD as its mathematical foundation. The word associations are solely derived from numerical analysis of texts and subsequently used for retrieval.

In the vector space, a query is represented by an $n \times 1$ vector, much like a column of the word-document matrix and with the same kind of weighting. For the retrieval, documents are ranked by their similarity to the query, typically using a cosine measure of similarity. In the vector semantic space, the cosine measure of a query q and a document d is represented as

$$sim(q,d) = \frac{q^T d}{\sqrt{q^T q \cdot d^T d}}$$

Both query and document are represented as a weighted vector sum of the vector representations of its constituent words. Therefore, the similarity between a query q and a document d in k-dimension semantic space can be computed as

$$sim\left(U_k^X(M)q, U_k^Y(M)d\right)$$

where $U_k^X(M)$ denotes the vector lexicon for query in language X and $U_k^Y(M)$ denotes the vector lexicon for document in language Y. For example, the similarity between an English query q_E and a French document d_F can be computed by

$$sim(U_k^E(M)\ \mathrm{q}_E, U_k^F(M)\ \mathrm{d}_F)$$

Since LSA represents both documents and queries as language independent numerical vectors in the same multi-lingual semantic space, queries in any language can retrieve documents in any language without the need of query translation. In other words, the similarity between a query and a document can be computed regardless of language. Moreover, CL-LSI is able to match queries against documents in all languages simultaneously.

Instead of merely matching query words exactly to the words in documents, LSA examines the similarity of the contexts of the words and extracts the similarity usage or meaning of the words. Taking advantage of this characteristic, CL-LSI is able to retrieve relevant documents which share no words of query in common.

(Rehder, et al., 1997) carried out the CL-LSI experiments for TREC-6. They used 80698 French-German parallel documents from Schweizerischen Depeschaenagentur, Swiss news agency, as a training set to create vector lexicons for German and French. In addition, other 3000 French-German parallel documents from the same source were taken as a verification set, which was used to calculate mate-retrieval value using the vector lexicons.

After performing some initial experiments, they chose 40000 French-German parallel documents with the highest similarity values. Subsequently, they extended the French-German retrieval system by including English. Over 40000 German documents, only 39988 English documents were created by machine-translating the German documents. As a consequence, the CL-LSI analysis was carried out on 39988 document collection of German with each French and English mate for each German document. The result of English to French mate retrieval was comparable to the initial German to French performance. This fact is noteworthy, because the statistical relationship between English and French in the system was very indirect.

Another CL-LSI experiment was carried out by (Orengo & Huyck, 2002) on Portuguese-English parallel texts. In that experiment, they calculated the similarities between some English words and their corresponding words in Portuguese. The task of computing the word similarity is akin to that of bilingual term mapping in this thesis (see Section 3.3).

2.4.3 WSD Using LSA

As a WSD task, the work in this thesis aims to automatically map the English senses derived from the Princeton WordNet to Indonesian senses enumerated in an Indonesian machine readable dictionary. Princeton WordNet, as a knowledge resource, specifies each English word sense via the corresponding gloss of the *synsets* containing the word. Particularly, Princeton WordNet assumes that each synonym in a *synset* conveys the same sense. As such, the work in this thesis attempts to build Indonesian *synsets*, i.e. collections of Indonesian words with the same sense.

The automatic assignment of Indonesian senses to English senses is carried out by mapping the previously established English senses to Indonesian senses. In order to do the automatic mapping, LSA technique is applied to English-Indonesian parallel texts. Subsequently, the sense similarity between English sense and Indonesian sense is computed (see Section 3.4). In a previous work, (Clodfelder, 2003) has shown that LSA applied on a small French-English parallel texts has some capability to align terms with similar orthographic form as well as terms with similar meaning.

To some extent, the task of computing the sense similarity is akin to that of computing query-document similarity in CL-LSI. While CL-LSI aims to retrieve relevant or similar documents of any language to a given query, computation of sense similarity aims to retrieve similar Indonesian senses to a given English sense. Specifically, the design of the term-document matrix for WSD using LSA in this thesis (see Section 3.2.1) resembles that of the CL-LSI described above.

On the other hand, bilingual term mapping, described in Section 3.1.1, attempts to map English words to their corresponding Indonesian words, regardless of WSD. Previous work on word mapping or alignment between English and Indonesian has been carried out by (Cathcart & Dale, 2001). They used statistical machine translation technique adopted from the IBM Model 1. On another work, (Deng & Gao, 2007) took advantage of bilingual LSA as a prior knowledge for statistical word alignment training. The word similarity in reduced dimensional LSA space was computed as a prior knowledge and it evidently improved the translation performance.

Chapter 3

Design

This chapter describes two tasks, namely *bilingual term mapping* and *bilingual concept mapping*, which are introduced to achieve the research objectives. Via statistical approach, a model employing Latent Semantic Analysis (LSA) is proposed to accomplish these tasks. The definition of the tasks, the description of how LSA is applied, and the design of the tasks are sequentially detailed in the following sections.

3.1 Task Definitions

The work in this thesis has a purpose to prepare the Indonesian *synsets* needed for building an Indonesian WordNet. By using expand approach (Darma Putra, Arfan, & Manurung, 2008), *synsets* along with their relations are derived from a previously established WordNet. Specifically, this work intends to provide Indonesian *synsets* by automatically mapping English concepts derived from an English WordNet, i.e. Princeton WordNet, to Indonesian concepts derived from an Indonesian machine readable dictionary, i.e. the electronic version of Kamus Besar Bahasa Indonesia (KBBI). Also, this work conducted automatic lexical mapping between English terms and Indonesian terms to verify the validity of the concept mapping, i.e. whether LSA is able to map English terms to their Indonesian analogues.

For the purpose of this work, a WordNet is delineated as a 4-tuple (C, T, χ, ω) as follows:

- A concept $c \in C$ is a semantic entity which represents a specific meaning of a concept of the world. Each concept is associated with a gloss, which is a textual description of its meaning. For example, concept c_1 and c_2 are defined, where c_1 is associated with the gloss *"a financial institution that accepts deposits and*