Chapter 4

Implementation Details

This chapter explains the implementation details of the design described in chapter 3. First of all, the overview of the implementation is described. In the following sections, it explains how to implement a bilingual term-document matrix and how to construct the corresponding LSA matrix. Particularly for bilingual concept mapping, it explains how to build a conceptual semantic matrix. Finally, it explains how to conduct both bilingual term and concept mappings.

4.1 Overview

Based on the design in chapter 3, some implementation is developed to carry out both bilingual term and concept mapping tasks. Before starting the implementation, some requisite data are prepared, including an English lexicon, an Indonesian lexicon, and an English-Indonesian parallel corpus. Specifically, for English and Indonesian lexicon databases, the SQL version of Princeton WordNet and KBBI are used, respectively. The parallel corpora used in this work have already been aligned, e.g. document level, and the document pairs can be identified, e.g. by their file numbers.

The implementation comprises three major components, which are the implementation of LSA described in Section 4.2, the implementation of conceptual semantic matrix described in Section 4.3, and the implementation of similarity matrix described in Section 4.4. All of these implementations are a combination of both PERL and MATLAB programming languages.

The implementation of LSA includes listing unique terms of English and Indonesian, creating parallel document collection, building and weighting bilingual term-document

matrix, applying SVD and finally building bilingual LSA matrix. Both English and Indonesian unique terms and a parallel document collection are used to define the rows and columns of a bilingual term-document matrix, respectively. Afterwards, a variety of weighting schemes can be applied to the bilingual term-document matrix.

Once a bilingual term-document matrix is established, whether weighted or not, SVD is ready to be applied on it. SVD decomposes a bilingual term-document matrix into three matrices, which are matrix U representing the original rows, matrix V representing the columns, and matrix Σ of the singular values. A bilingual LSA matrix is then created by reducing the dimension of the three matrices and then reconstructing them into a single matrix. It is necessary to divide the bilingual LSA matrix into two LSA matrices for each language, so as to carry the subsequent processes out.

The task of bilingual term mapping can be approximated by comparing similarity of term vectors, which represents English and Indonesian terms. These term vectors are obtained from their corresponding LSA matrices. Afterwards, the similarity matrix containing the mapping result is constructed.

On the other hand, bilingual concept mapping can be approximated by comparing similarity of conceptual semantic vectors, which represents English and Indonesian concepts. The implementation of conceptual semantic matrix makes use of the monolingual LSA matrices to provide semantic information in the construction of the conceptual semantic vectors. Then, the conceptual semantic matrices of two languages are used as the inputs of the implementation of similarity matrix.

4.2 Implementing LSA

To attain both bilingual term and concept mappings, some lexical information of the two languages is needed. Essentially, unique terms of the two languages are required to define the rows of a bilingual term-document matrix. For both languages, the list of unique terms, i.e. list of unique words and short phrases, is derived from their lexicons. Explicitly, the synonyms of the *synsets* (including English words and short phrases) and the *sublemmas* (including Indonesian words and short phrases) are listed.

Some other terms from the gloss and the example sentences, which are not already in the list, are added. In the case of English, most of the additional terms are the inflected orthographic forms. For example, the term *enumerate* is already in the list. Then, the

added terms are *enumerated*, *enumerates*, and *enumerating*. Eventually, the list of unique terms yields 169583 English unique terms and 87171 Indonesian unique terms.

Next, a parallel document collection is needed to define the columns of a term-document matrix. The collection can be derived randomly from a parallel corpus with respect to a certain size. Let P be the parallel corpus, then P_x is a document collection, i.e. a subset of P, of x document pairs. The work in this thesis makes use of several document collections with different size where collection of smaller size is specified as a subset of that of larger size.

```
function COLLECTION (Files, InitialFiles, initialSize, collectionSize, RandomArray)
returns a document collection

if initialSize = 0
    for i from 0 to collectionSize-1 do
    j ← RandomArray[i]
    FileList[i] ← Files[j]
else
    for i from 0 to initialSize-1 do
        FileList[i] ← InitialFiles[i]
        delete FileList[i] from Files

    for i from initialSize to collectionSize-1 do
        j ← RandomArray[i]
        FileList[i] ← Files[j]

return FileList
```

Figure 4.1 Pseudocode for Creating a Document Collection

Figure 4.1 above describes the pseudocode for creating a document collection. This pseudocode is implemented in PERL. Function **COLLECTION** has five parameters, which are *Files*, *InitialFiles*, *initialSize*, *collectionSize*, and *RandomArray*. *Files* is an array of file paths of all document in a corpus. *InitialFiles* is also an array of document file paths, but only those which are already included in a smaller collection. For creating the smallest collection, the *InitialFiles* is merely an empty array.

Essentially, a collection of larger size is created based on that of smaller size. The size of the collection of smaller size is carried by the variable *initialSize*, whereas the size of that of larger size is initialised by the variable *collectionSize.* The last parameter is *RandomArray*, which is an array of random numbers used to select document files randomly. Since the parallel documents used in this work have already been aligned via the same document file number, a parallel document collection can be created by using the same *RandomArray.*

```
function TERM-DOCUMENT-MATRIX (FileList, UniqueTerms)
returns a term-document matrix

for i from 0 to length(FileList) -1 do
    CollectionUniqueTerms = COUNTTERM(READFILE(FileList[i]), UniqueTerms)
    UniqueTerms = CollectionUniqueTerms

foreach term (sort keys UniqueTerms) do
    if UniqueTerms{term} < 1)
        delete UniqueTerms{term}
    else
        UniqueTerms{term} ← 0

for j from 0 to length(FileList) -1 do
    TermFrequency = COUNTTERM(READFILE(FileList[i]), UniqueTerms)
    i ← 0
    foreach term (sort keys TermFrequency) do
        TermDocument[i,j] ← TermFrequency{term}
        i++

    TermFrequency ← 0

return TermDocument
```

Figure 4.2 Pseudocode for Building a Term-Document Matrix

Once a parallel document collection and the list of unique terms from the lexicons of both languages have been prepared, a bilingual term-document matrix is ready to be built. The idea is to build two monolingual term-document matrices and then join them appropriately. To build a monolingual term-document matrix, first of all, unique terms of the corresponding language which appears in the collection should be listed. The rows of the term-document matrix then denote all the collection unique terms which are in the list of lexicon unique terms. On the other hand, the columns simply denote the documents. Each cell therefore contains the frequency of a term in a particular document.

Figure 4.2 describes the pseudocode for building a term-document matrix. This pseudocode is implemented in PERL. Function TERM-DOCUMENT-MATRIX has two parameters, which are *FileList* and *UniqueTerms*. *FileList* is an array of file paths of all monolingual documents in a parallel collection. *UniqueTerms* is a hash whose keys are the lexicon unique terms and values are their frequencies.

```
function TFIDF-WEIGHTING (TermDocument) returns a weighted term-document
matrix

rowSize ← row size of TermDocument
columnSize ← column size of TermDocument

for i from 0 to rowSize-1 do
docSize ← number of column of TermDocument[i,:]
    if docSize > 1
        for j from 0 to columnSize-1 do
            Idf[i,j] ← log2(columnSize/docSize)
            WeightedTermDocument[i,j] ← TermDocument[i,j]* Idf[i,j]
    else
        WeightedTermDocument[i,:] ← 0

return WeightedTermDocument
```

```
function LOGENTROPY-WEIGHTING (TermDocument)
returns a weighted term-document matrix

rowSize ← row size of TermDocument
columnSize ← column size of TermDocument
GlobalFreq ← array of sum of all values in a TermDocument row
entropy ← 0

for i from 0 to rowSize-1 do
    for j from 0 to columnSize-1 do
        LocalWeight[i,j] ← log(TermDocument[i,j]+1)
        P[i,j] ← TermDocument[i,j] / GlobalFreq[i]
        entropy ← entropy + ((P[i,j]*log2(P[i,j]))/ log2(columnSize))

    GlobalWeight[i] ← entropy +1
    entropy ← 0

for i from 0 to rowSize-1 do
    for j from 0 to columnSize-1 do
        WeightedTermDocument[i,j] ← LocalWeight[i,j]* GlobalWeight[i]

return WeightedTermDocument
```

Figure 4.3 Pseudocodes for *TF-IDF* and *Log-Entropy* Weighting

The function COUNTTERM counts the frequency of each terms of a given document, where the frequencies of phrases are counted first, before the single words. Formerly, the document is cleansed from the document tags and unnecessary punctuations by the function READFILE. Also, the text of the document is changed to lowercase. All the punctuations are removed, except dots, apostrophes and dashes which are enclosed by alphabets or numbers. The exceptions are due to the characteristic of both English and Indonesian terms. For example, the English terms include *'hood*, and *.22-calibre*.

Various weighting schemes can be applied to a term-document matrix. In this work, two weighting schemes, *TF-IDF* and *Log-Entropy*, are applied. Figure 4.3 describes the pseudocodes for both *TF-IDF* and *Log-Entropy* weighting. These pseudocodes are implemented in MATLAB. See Section 3.2.1 for the weighting formulas.

Finally, two monolingual term-document matrices of a parallel document collection, whether weighted or not, can be concatenated together as a bilingual term-document matrix. In practise, the concatenation is made in the function LSA-MATRIX described in Figure 4.4.

```
function LSA-MATRIX (EngTermDocument,IndTermDocument,k)
returns a bilingual LSA matrix

M ← concatenate EngTermDocument and IndTermDocument
[U,S,V] ← svd(M)

Mk ← U(:,1:k)*S(1:k,1:k)*V(:,1:k)'

return Mk
```

Figure 4.4 Pseudocode for Building an LSA Matrix

The pseudocode above is implemented in MATLAB. Function LSA-MATRIX has three parameters, which are *EngTermDocument*, *IndTermDocument*, and *k*. *EngTermDocument* is an English term-document matrix and *IndTermDocument* is the corresponding Indonesian term-document matrix. Lastly, *k* is the variable which determines the rank approximation of the LSA matrix. This function eventually returns a bilingual LSA matrix. However, to carry out the mapping, it is necessary to split the bilingual LSA matrix back into English and Indonesian LSA matrices.

4.3 Building Conceptual Semantic Matrix

In essence, conceptual semantic matrix consists of conceptual semantic vectors which represent concepts. A conceptual semantic vector for a concept can be approximated by constructing a set of textual context representing that concept. This set of textual context can be derived from some lexical resource or lexicon. (See Section 3.4)

In this work, Princeton WordNet serves as the English lexicon. Specifically, WordNet represents an English concept as a union of a *synsetid*, the *synset* for the *synsetid*, the gloss of the *synset*, and some example sentences. These data are acquired from the SQL version of Princeton WordNet. The result provides adequate textual contexts for the English concepts. Note that a *synsetid* is merely an identity of a textual context set. Thus, a set of textual context only includes a synset, a gloss, and example sentences.

Query Result Structure: synsetid # synonym # gloss # example

100028651#infinite#the unlimited expanse in which everything is located#they tested his ability to locate objects in space#

100028651#infinite#the unlimited expanse in which everything is located#the boundless regions of the infinite#

100028651#space#the unlimited expanse in which everything is located#they tested his ability to locate objects in space#

100028651#space#the unlimited expanse in which everything is located#the boundless regions of the infinite#

Figure 4.5 Sample of Query Result Acquired from Princeton WordNet Database

The result of the query to the Princeton WordNet database, however, has not enumerated the compact sets of the English concept textual-contexts yet. Nevertheless, it specifies its line as a unique union of a *synsetid*, **a synonym** of the *synset*, the gloss, and **an example sentence for the particular synonym**. Figure 4.5 shows a query result sample for a single concept where each component of the lines, i.e. column of the database table, is separated by the symbol #. The query result must be modified so that some results explaining the same concept merge into a single textual context set. In the case of the query result in Figure 4.5, the four rows are merged into a single textual context set covering all information, which is shown in Figure 4.6.

Query Result Structure: synsetid # synonym # gloss # example

100028651#infinite space#the unlimited expanse in which everything is located#they tested his ability to locate objects in space the boundless regions of the infinite#

Figure 4.6 Textual Context Set for Concept in Figure 4.5

On the other hand, KBBI serves as the Indonesian lexicon in which an Indonesian concept is represented as a union of *kebiid*, *lemma* (bare word), *sublemma*, definition, and example sentences. For the purpose of this work, other attributes are just ignored. These data are acquired from the KBBI database and used to provide textual-context sets for Indonesian concept.

Query Result Structure: kebiid # lemma # sublemma # definition # example

k00141#abnormal#abnormal#tidak normal; tidak sesuai dengan keadaan yg biasa; mempunyai kelainan#hidup dl keadaan yg abnormal; sejak kecelakaan itu dia menjadi abnormal#

k00142#abnormal#keabnormalan#keadaan tidak normal#keabnormalan pertumbuhan anak dapat dicegah dng perawatan medis#

Figure 4.7 Sample of Query Result from KBBI Database

Like *synsetid*, *kebiid* is simply an identity for the textual context set or the concept itself. Thus, a textual context set of an Indonesian concept only consists of the *sublemma*, the definition, and the example sentences. Unlike the query result for English, the query result for Indonesian has already enumerated compact sets of the Indonesian concept textual-contexts. The lines of the result are distinguished by unique pairs of *sublemma* and *definition*.

For each set of textual context, a conceptual semantic vector is constructed so as to represent the corresponding concept. Specifically, a conceptual semantic vector is composed by averaging and weighting the textual context term vectors. First, the term vectors are taken from an LSA matrix with respect to its language. Then, some term vectors which form a component or subset of the textual context set are averaged and weighted. Thus, a single component vector will be created. In the case of English textual

context set, the components are the *synset*, the gloss, and the example sentences. On the other hand, components of Indonesian textual context set are the *sublemma*, the definition, and the example sentences.

```
function CONCEPTUAL-SEMANTIC-MATRIX (LSAMatrix, textualContext, delimiter, Weight)
returns a conceptual semantic matrix

rowSize ← row size of LSAMatrix
columnSize ← column size of LSAMatrix
numberOfConcept ← number of textualContext lines

for i from 0 to numberOfConcept do
    line ← a line of textualContext
    [synsetid, setOfTextualContext] ← STRINGTOKEN(line)
    [frontDelimiter, context] ← STRINGTOKEN(setOfTextualContext)

    ConceptVector ← 0
    ComponentVector ← 0

    w ← 0 index of weight
    y ← 0 number of term

    while context
        [contextTerm, context] ← STRINGTOKEN(context)
        if contextTerm
          e ← STRING2NUM(contextTerm)

        if e = delimiter
            if y > 0
                ConceptVector ← ConceptVector + (Weight[w]* ComponentVector)/y
                ComponentVector ← 0
            w ++
            y ← 0
        else
            TermVector ← LSAMatrix(e,:)
            ComponentVector ← ComponentVector + TermVector
            y++

    ConceptualSemanticMatrix [i,:] ← ConceptVector

return ConceptualSemanticMatrix
```

Figure 4.8 Pseudocode for Building Conceptual Semantic Matrix

Those component vectors of a textual context set are summed into a single conceptual semantic vector. All conceptual semantic vectors of a particular language are merged into a conceptual semantic matrix. Notice that a conceptual semantic matrix is monolingual, i.e. of a specific language.

Figure 4.8 describes the pseudocode for building a conceptual semantic matrix. This pseudocode is implemented in MATLAB. Function CONCEPTUAL-SEMANTIC-MATRIX has four parameters, which are *LSAMatrix*, *textualContext*, *delimiter*, and *Weight*. *LSA Matrix* is an LSA matrix of a particular language. Then, *textualContext* is a text file enumerating textual context sets of that language in its lines and *delimiter* is a variable to initialise the delimiter between components of the textual context set. The last parameter, *Weight* is an array containing the weights for the textual context subsets or components, which are ordered according to the order of the components.

4.4 Conducting Bilingual Mapping

Both bilingual term mapping and bilingual concept mapping are approximated by comparing similarity between bilingual semantic vectors. Bilingual term mapping takes advantage of the similarity of the bilingual term vectors, which represent terms in English and Indonesian. The term vectors are obtained from a monolingual LSA matrix with respect to the language of the terms.

According to the bilingual term mapping task, an English term is supposed to be mapped to Indonesian terms which may convey all concepts that can be conveyed by the English term. In the implementation, each English term are mapped to some Indonesian terms with the highest similarity values.

On the other hand, bilingual concept mapping takes advantage of the similarity of bilingual conceptual semantic vectors, which represent English and Indonesian concepts. The conceptual semantic vectors are obtained from a conceptual semantic matrix with respect to the language of the concepts.

According to the bilingual concept mapping task, pairs of English-Indonesian concepts which represent the same concept should be established. In the implementation, the similarity between each English conceptual semantic vector and each Indonesian semantic vector is computed. Then, some conceptual semantic vector pairs with the highest similarity values are designated as the mapping results for the corresponding concept.

Figure 4.9 shows the pseudocode for computing the similarity between bilingual semantic vectors and building a similarity matrix, which contains the result of either bilingual term mapping or bilingual concept mapping. This pseudocode is implemented in MATLAB. For bilingual term mapping, a lexical similarity matrix is created, whereas for bilingual concept mapping, a conceptual similarity matrix is created.

```
function SIMILARITY (EngMatrix, IndMatrix, Selection) returns a similarity matrix

engRowSize ← row size of EngMatrix
indRowSize ← row size of IndMatrix

for i from 1 to engRowSize do

  engvector ← []
  indvector ← []
  simval ← []

    for j from 1 to indRowSize do
        n ← NORM(EngMatrix[i,:]) * NORM(IndMatrix [j,:])
        cosine ← SUM(EngMatrix[i,:] .* IndMatrix [j,:])/n

        simval ← [simval cosine]
        engvector ← [engvector i]
        indvector ← [indvector j]

  R ← [engvector'  indvector' simval']
  R ← sort rows of R in descendent order according to simval
  R ← take only a number of top rows according to Selection

  SimilarityMatrix[i,:] ← R

return SimilarityMatrix
```

Figure 4.9 Pseudocode for Building Similarity Matrix

Function SIMILARITY has three parameters, which are *EngMatrix*, *IndMatrix*, and *Selection*. For bilingual term mapping, *EngMatrix* is a matrix consisting of English lexical semantic vectors, i.e. an English LSA matrix. Conversely, for bilingual concept mapping, it consists of English conceptual semantic vectors, i.e. English conceptual semantic matrix. Similarly, *IndMatrix* is a matrix consisting of Indonesian lexical semantic vectors, i.e. Indonesian LSA matrix, in the case of bilingual term mapping. In the case of bilingual concept mapping, it consists of Indonesian conceptual semantic vectors, i.e. Indonesian conceptual semantic matrix.

The third parameter, *Selection*, is a variable which represents the means used for determining the result of mapping. Several means which can be applied are choosing top n semantic vectors with the highest similarity and choosing all semantic vectors above a minimum threshold. The threshold can be defined in a variety of ways, e.g. the average of the similarity values.

Chapter 5

Result and Discussion

This chapter describes bilingual term and concept mapping experiments carried out in this work. Additionally, some bilingual term mapping experiments were carried out for testing text alignment granularity. First, the existing resources and the variables used are listed. Then, the summaries of mapping results and some discussions are given.

5.1 Bilingual Term Mapping Experiment

Firstly, bilingual term mapping experiments are conducted by defining a collection of document pairs and then building a term-document matrix from it. In other way, a term-document matrix can be built from a collection which excludes the *stopwords*. Then, a weighted term-document matrix can be created by applying *TF-IDF* or *Log-Entropy* weightings on the original term-document matrix (see Section 3.2.1).

LSA procedure is applied to a term-document matrix whether weighted or not, i.e. SVD of the matrix is computed and an LSA matrix is created by reconstructing the matrix with rank-k approximation. Finally, based on the LSA matrix, a set of English terms are mapped to n most similar Indonesian terms by computing the cosine measure of similarity between the English and Indonesian term vectors (see Section 3.2.2).

Two baselines are provided as comparison to bilingual term mapping results. For a very simple baseline, called random baseline, n Indonesian terms appearing in a collection are selected randomly as translations or mapping results for an English term. For a more sophisticated baseline, called frequency baseline, the original term-document matrix consisting of the raw term frequency is used to compute the term similarity.

5.1.1 Existing Resources

Several resources used to perform the bilingual term mapping experiments are:

- **Parallel Corpus 1**

  For parallel corpus 1, 3273 English-Indonesian article pairs were taken from ANTARA news agency. These articles have been paired semi-automatically using a statistical approach developed by Mirna Adriani and Monica Lestari Paramita at the Information Retrieval Laboratory, Faculty of Computer Science, Universitas Indonesia.

  Since articles of parallel corpus 1 were paired semi-automatically, the accuracy of the alignment is rather doubtful. There is a possibility that the articles are not accurately aligned. Moreover, the translations are not of a very high quality. The corpus might be more suitably considered as comparable rather than parallel. Since LSA interpretation relies on the article contexts, these aspects may affect LSA performance. (See Appendix A.1.1 for article pair samples)

- **Parallel Corpus 2**

  For parallel corpus 2, a couple of English and Indonesian bibles were taken from http://bibledatabase.org. For the English bible, the Basic Bible version written in Basic English is used. On the other hand, the new translation version of Indonesian bible is used, i.e. the Indonesian bible released by Lembaga Alkitab Indonesia (Indonesian Bible Society) in 1994.

  In experiments, verses are treated as documents. These verses are much smaller than articles in parallel corpus 1. A verse usually contains no more than two sentences. According to the maximum capability of the hardware to compute an LSA matrix, only the first 3500 verses of each bible were used. This set of verses covers almost three books of the bible, i.e. from Genesis to Leviticus 25:30, and composes 115 chapters.

- **English-Indonesian machine readable dictionary**

  A bilingual English-Indonesian machine readable dictionary was constructed by combining data from various resources, including a handcrafted dictionary by

Hantarto Widjaja[2], a bilingual dictionary created by BPPT, kamus.net, and word translations of Transtool version 1.6. This bilingual dictionary consists of 37678 English unique terms and 60564 Indonesian unique terms.

5.1.2 Variables

Several variables observed in the bilingual term mapping experiments are:

- **Collection Size**

  Three collections of article pairs were created by randomly generating subsets of the parallel corpus 1. Let P_x denote a collection of x article pairs. The first collection P_{100} consists of 100 article pairs, the second collection P_{500} consists of 500 article pairs, and the third collection P_{1000} consists of 1000 article pairs. Each successive collection entirely contains the previous collections, i.e. $P_{100} \subset P_{500} \subset P_{1000}$.

  From parallel corpus 2, four collections of verses were created by choosing x first verses of both English and Indonesian bibles. Three collections are of the same size with the collections of parallel corpus 1 and the fourth collection P_{3500} consists of 3500 verses.

- **Rank Approximation**

  For each collection, LSA reconstructs several term-document matrices with different rank approximations. Generally, the rank approximations are 10%, 25%, and 50% the number of dimensions of the original collection. For instance, the 10, 25, and 50-rank approximations are computed for P_{100}.

- ***Removal of Stopwords***

  Stopwords are words that appear numerously in a text or discourse, thus are assumed as insignificant to represent the specific context of a text or discourse. These words are similar to closed class words or function words. Removal of *stopwords* is a common technique used to improve performance of information retrieval systems. Following the idea, this technique is applied in preprocessing the collections, i.e. removing all instances of the *stopwords* in the collections before applying LSA, is expected to improve the LSA performance.

[2] http://www.geocities.com/CapeCanaveral/1999/

- **Frequency Weighting**

  Various weighting schemes can be applied to the raw frequency of the original term-document matrix so as to adjust the degree of importance of a term in a particular document. In this work, two weighting schemes are applied, namely *TF-IDF* and *Log-Entropy* weightings. (See Section 3.2.1)

- **Source Terms Selection**

  To approximate the bilingual term mapping results, a hundred English terms are randomly selected from P_{100}. Hence, it ensures that the terms also appear in larger collections, i.e. the supersets of P_{100}. Another way of selecting the source terms is used to verify the underlying intuition that terms which appear frequently in the collection should have better semantic context provided by the collection. For this reason, the top 100 English terms are selected according to their cumulative frequency in documents of P_{100}.

- **Target Term Mapping Selection**

  Each English term is mapped to several Indonesian terms, i.e. the target terms, by choosing the top *n* terms with the highest similarity values, i.e. the top 1, 10, 50, and 100 terms are reserved as mapping results or Indonesian translation sets.

5.1.3 Sample of Bilingual Term Mapping Result

A sample of bilingual term mapping results is given in Table 5.1. In this sample, the mapping results of English term *film* present a successful mapping, whereas the mapping results of English term *billion* present an unsuccessful mapping. Both are taken from experiment results using P_{1000} from the parallel corpus 1, which includes the *stopwords*, and LSA matrix of rank 50% approximation. For each English term, top 10 Indonesian terms with highest similarity values were taken as the Indonesian translations, i.e. using mapping selection Top 10.

LSA correctly mapped the English term *film* to its Indonesian translation *film,* which has the highest similarity value among others. Note that, although they share the same orthographic form, they are treated separately as any other terms. In other words, each term has its own term vector regardless of the orthographic form. Additionally, LSA suggests other Indonesian translations which are semantically related. Indeed, it shows the LSA capability of approximating similarity meaning of terms.

Table 5.1 Sample of Bilingual Term Mapping for English terms (a) film and (b) billion

(a) English Term: *film*

Indonesian Translations	Similarity Value
film	0.828
sutradara	0.702
filmnya	0.691
garapan	0.569
perfilman	0.547
penayangan	0.545
kontroversial	0.539
bioskop	0.490
menyabet	0.475
aktor	0.455

(b) English Term: *billion*

Indonesian Translations	Similarity Value
terjadwal	0.973
zero	0.973
yudistira	0.973
silika	0.973
setengahnya	0.973
pengrajin	0.973
memproses	0.973
batuan	0.973
akun	0.973
viskositas	0.973

On the other hand, suggested translations for the English term *billion* are incorrect and have no semantic relatedness with *billion*. More importantly, the correct translation *milyar* is missing. This failure may be due to several factors. Firstly, *billion* does not by itself invoke a particular semantic frame, and thus its term vector might not suggest a specific conceptual domain. Secondly, billion can sometimes be translated numerically instead of lexically. In general, however, the failure is believed simply due to the lack of data: the collection is simply too small to provide useful statistics that represent semantic context. Similar LSA approaches are commonly trained on collections of text numbering in the tens of thousands of articles, e.g. (Rehder, Littman, Dumais, & Landauer, 1997).

Although the translations for *billion* are incorrect, their similarity values are very high. This fact suggests that the similarity values do not accurately reflect the correctness of the term translations. Thus, it is necessary to evaluate the mapping results against a gold standard, which is achieved by comparing their precision and recall against the Indonesian terms returned by the bilingual dictionary, i.e. how isomorphic is the set of LSA-derived term mappings with a human-authored set of term mappings?

In the case of bilingual term mapping, precision represents the number of translation pairs matching the bilingual dictionary pairs, which is divided by the number of all translation pairs taken according to the mapping selection used. On the other hand, recall represents the number of translation pairs matching the bilingual dictionary pairs, which is divided by the number of translation pairs given by bilingual dictionary.

Table 5.2 Comparison of Mapping Results

English Term: *film*			
Bilingual dictionary	**Random Baseline**	**Frequency Baseline**	**LSA**
lapisan tipis selaput ***film*** memfilemkan saput pilem memfilmkan	jenderal mimi darah tko balai liburan berbakti paham generasi sabar	***film*** sutradara filmnya kontroversial penayangan sumbangsih segara menyuntikkan garapan epik	***film*** sutradara filmnya garapan perfilman penayangan kontroversial bioskop menyabet aktor
Precision	0.00	0.10	0.10
Recall	0.00	0.14	0.14

Table 5.2 shows the comparison of mapping results for the English term *film*, given by the bilingual dictionary, random baseline, frequency baseline, and LSA. Random baseline performs the worst, i.e. no terms match with the bilingual dictionary terms. Frequency baseline and LSA share the same precision and recall values. Nevertheless, not all terms given by the frequency baseline are semantically related to the English term *film*, e.g. the terms *sumbangsih*, *segera*, and *menyuntikkan*.

5.1.4 Effect of Source Terms Selection and Frequency Weighting

In this work, two source term sets, namely *random set* and *top score set*, were created using P_{100} of parallel corpus 1. The *random set* contains 100 English terms selected randomly, whereas the *top score set* contains top 100 of English terms sorted in descending order according to their cumulative frequency in P_{100}. Other experiments using larger collections, P_{500} and P_{1000}, use these two sets as well. See the list of these sets in Appendix A.1.2 and A.1.3.

In the beginning, the mappings were performed for the random set and the LSA experiments were carried out using collections P_{100}, P_{500}, and P_{1000} which include the *stopwords* and those collections which exclude the *stopwords*. For each collection, the 10%, 25%, and 50% rank approximations were computed. Then, four sets of Indonesian translations were taken using different mapping selections, which are the top 1, 10, 50, and 100 terms with the highest similarity values (See Appendix B.1.1).

For these experiments, some frequency baselines were also computed. The variable setups were the same as LSA experiments, but the frequency baselines computed the term vector similarity using full term-document matrices only.

Since the mapping results for the random set were very poor, bilingual term mappings using top score set were carried out to verify the underlying intuition that terms which appear frequently in the collections contain better semantic context and thus may yield better results. For LSA experiments and frequency baselines, the mappings used the same variable setups as the mapping for random set, correspondingly.

Since the similarity values cannot accurately reflect the correctness of the mappings, the mapping results were compared to the mappings of bilingual dictionary and their precisions and recalls were computed (see Section 5.1.3). Of the 100 English terms in the random set, only 95 terms exist in bilingual dictionary. For the top score set, only 88 terms exist.

Table 5.3 shows the mapping result summaries of the LSA experiments and the frequency baselines. The **P** column shows the average precision of all experiments carried out for either random set or top score set. Similarly, the **R** column shows the average recall of those experiments.

Table 5.3 Comparison of Source Terms Selection

Source Terms	FREQ		LSA No Weighting	
	P	R	P	R
Random Set	0.0181	0.0623	0.0185	0.0557
Top Score Set	**0.1009**	**0.2285**	0.0757	0.1948

Generally, the average precision and recall of the mapping results are very small. This failure may be due to several factors. Firstly, as mentioned in Section 5.1.1, the parallel corpus 1 may be more suitably considered as comparable rather than parallel. Although containing similar topics, the English articles are not the proper translations of the Indonesian articles. The Indonesian translations of English terms in an English article may not appear in the corresponding Indonesian article. Thus, the translation pairs cannot be obtained.

Secondly, the quality of bilingual dictionary may influence the precision and recall values. The bilingual dictionary used in this work may be too small, e.g. Transtool only

gives one translation per term. Since an English term can be translated into different Indonesian terms with the same meaning, correct mapping results may not exist in the bilingual dictionary.

Even though the precision and recall values are very small, experiments for top score set consistently yield significantly higher values than those for random set. This fact confirms the intuition that top score terms are likely to appear more often than random terms, thus they should have better semantic context provided by the collection.

Average precision and recall values of frequency baseline are higher than LSA, except for the average precision using random set. To improve the LSA performance, two weighting schemes were applied to the term-document matrices. Using the same rank approximations, LSA was applied to the weighted term-document matrices. Then, bilingual term mappings were carried out for the top score set only (See Appendix B.1.2).

Table 5.4 Comparison of Weighting Usage

Weighting Usage	FREQ		LSA	
	P	R	P	R
No Weighting	0.1009	0.2285	0.0757	0.1948
Log-Entropy	**0.1347**	**0.2753**	0.1041	0.2274
TF-IDF	0.1013	0.2319	0.0694	0.1802

Although LSA using TF-IDF seems to perform worse than without using any weighting, LSA using Log Entropy weighting does perform better. It yields better precisions than frequency baseline without weighting, but frequency baselines using the Log Entropy performs even better.

5.1.5 Effect of Collection Size and Rank Approximation

From parallel corpus 1, three article pair collections of size 100, 500, and 1000 were created. Similarly, four verse pair collections of size 100, 500, 1000, and 3500 were created from parallel corpus 2. Since verse size is likely to be much smaller than article size, collections of parallel corpus 2 contain less unique terms than those of parallel corpus 1. Table 5.5 shows the statistics of both English and Indonesian unique terms contained in the collections.

Table 5.5 Unique Term Statistics

Collection Size	With *stopwords*		Without *stopwords*	
	Indonesian Unique Terms	English Unique Terms	Indonesian Unique Terms	English Unique Terms
P_{100}	3943	4477	3702	4404
P_{500}	8192	10224	7926	10149
P_{1000}	10461	13364	10188	13289

(a) Parallel Corpus 1

Collection Size	With *stopwords*		Without *stopwords*	
	Indonesian Unique Terms	English Unique Terms	Indonesian Unique Terms	English Unique Terms
P_{100}	466	403	379	344
P_{500}	1233	932	1091	867
P_{1000}	1897	1243	1735	1177
P_{3500}	3696	2082	3498	2016

(b) Parallel Corpus 2

Table 5.6 presents the mapping result summaries which emphasize the effect of varying the collection size of parallel corpus 1 and 2. The results show that the larger the collection size is, the higher the precision and recall values are. The intuition says that large collection sizes provide more semantic context used for mapping the terms. Moreover, the results show that LSA confidently outperforms the random baselines. However, frequency baseline seems to consistently perform better than LSA.

For parallel corpus 1, the mapping results of LSA were taken from all experiments for top score set described in Section 5.1.4. The mapping results of frequency baseline were only taken from experiments using term-document matrices without weighting (See Appendix B.1.2). The average values of precision and recall were then computed in terms of the collection size. For each collection of parallel corpus 1, some random baselines were also computed. The values in column **P** and **R** represent the average precisions and recalls of their results. (See Appendix B.1.3)

Since the top score set consistently has greater precision and recall values than random set (see Section 5.1.4), all experiments using parallel corpus 2 were conducted for top score set only. Like parallel corpus 1, the top score set was created based on English P_{100} of parallel corpus 2. Of the 100 English terms, only 90 terms exist in the bilingual dictionary.

Moreover, LSA experiments employing parallel corpus 2 only use term-document matrices without weighting. The term-document matrices were built from collections including the *stopwords* as well as from collections excluding the *stopwords*. The same rank approximations and mapping selections as experiments employing parallel corpus 1 were used. Some frequency baselines employing parallel corpus 2 were computed. However, no random baseline was computed. (See Appendix B.1.4)

Table 5.6 Effect of Collection Size

Collection Size	RNDM		FREQ		LSA	
	P	R	P	R	P	R
P_{100}	0.0003	0.0027	**0.0513**	**0.1601**	0.0346	0.1053
P_{500}	0.0002	0.0021	**0.1124**	**0.2535**	0.0974	0.2368
P_{1000}	0.0001	0.0024	**0.1391**	**0.2721**	0.1172	0.2603

(a) Parallel Corpus 1

Collection Size	FREQ		LSA	
	P	R	P	R
P_{100}	**0.0765**	**0.1386**	0.0588	0.1220
P_{500}	**0.0838**	**0.1443**	0.0678	0.1282
P_{1000}	**0.1204**	**0.1969**	0.1093	0.1860
P_{3500}	**0.1277**	**0.2156**	0.1210	0.2137

(b) Parallel Corpus 2

For P_{100}, the results of mappings using parallel corpus 2 are higher than those of mappings using parallel corpus 1. Nonetheless, the results using parallel corpus 2 for larger collections are lower compared to those using parallel corpus 1 for the corresponding collection size.

Essentially, this is because the absolute term size of P_x of parallel corpus 1 is different with that of parallel corpus 2. Remember that articles of parallel corpus 1 are of larger size, i.e. contain more terms, than verses of parallel corpus 2. Thus, collections of parallel corpus 1 should contain more information needed to do the mapping.

It is interesting though that the results of parallel corpus 2 collections, which are much smaller than parallel corpus 1 collections, are comparable to those of parallel corpus 1 collections, especially for the average precision values. P_{3500} comprises about 115 bible chapters, thus if the chapters are assumed to be articles in parallel corpus 1, then the results of P_{3500} can be compared to P_{100} of parallel corpus 1. It is noteworthy that the average precision of P_{3500} is more than twice higher than P_{100} of parallel corpus 1.

Intuitively, the translation of parallel corpus 2 is better than that of parallel corpus 1. Recall that the parallel corpus 1 may be more suitably considered as comparable rather than parallel (see Section 5.1.2). The Indonesian translations of English terms in an English article may not appear in the corresponding Indonesian article, thus the translation pairs cannot be obtained. Parallel corpus 2, on the other hand, is of quite high quality translation (see A.2.1). It may significantly improve the correctness of the mappings.

Moreover, the smaller size of documents in parallel corpus 2 may also be significant for the improvement. The results of the experiments testing the effect of text alignment granularity confirm that bilingual term mapping results are improved by using documents with finer text alignment granularity (see Section 5.1.8).

Yet, the average precision of parallel corpus 2 is still considerably small. This is suspected because the contents of parallel corpus 2 are homogenous. Thus, it may be difficult for LSA to discern the proper context of the terms. It suggests that using a balanced corpus containing more variety of domains probably can improve the results.

For each collection of parallel corpus 1 and 2, LSA was applied on the corresponding term-document matrix using different rank approximations, i.e. 10%, 25%, and 50% the number of dimension of the original collection. On the other hand, frequency baselines were computed using the 100% rank of those term-document matrices.

Table 5.7 Effect of Varying Rank Approximation

Rank Approximation	P	R
10%	0.0680	0.1727
25%	0.0845	0.2070
50%	0.0967	0.2226
100%	**0.1009**	**0.2285**

(a) Parallel Corpus 1

Rank Approximation	P	R
10%	0.0828	0.1554
25%	0.0881	0.1637
50%	0.0967	0.1684
100%	**0.1021**	**0.1739**

(b) Parallel Corpus 2

Table 5.7 shows the effect of using term-document matrix with different rank approximations. For parallel corpus 1, the average precision and recall values were computed using all experiment results for top score set only, which are described in Section 5.1.4. For parallel corpus 2, the values were computed using the experiment results explained above, i.e. varying only the collection size, the rank approximation, the removal of *stopwords*, and the mapping selection (See Appendix B.1.2 and B.1.4).

The results show that experiments using term-document matrix with rank 100% yield the highest results. Subsequently, it suggests that frequency baseline performs better than LSA. This issue is particularly discussed in Section 5.1.9.

5.1.6 Effect of Stopwords

Removal of *stopwords* is a common technique used to improve information retrieval systems. Likewise, this technique is expected to improve LSA performance in bilingual term mapping. Specifically, LSA is applied to a term-document matrix built from a collection in which the stopwords have already been removed. Thus, the directions of term vectors in reduced-rank semantic space are not affected by the *stopwords* vectors.

Table 5.8 shows the comparison between mapping results of experiments using collections containing the *stopwords* and experiments using those which excludes the *stopwords*. The average precision and recall values were computed using experiment results for top score set only (See Appendix B.1.2, B.1.3, and B.1.4). The experiment results, however, surprisingly tend to be worse than not removing the *stopwords*. This issue is further discussed in Section 5.1.9.

Table 5.8 Effect of Stopwords

Stopwords	RNDM		FREQ		LSA	
	P	R	P	R	P	R
Contained	0.0002	0.0023	**0.1009**	**0.2285**	0.0840	0.2051
Removed	0.0002	0.0028	**0.1009**	**0.2285**	0.0822	0.1964

(a) Parallel Corpus 1

Stopwords	FREQ		LSA	
	P	R	P	R
Contained	**0.1058**	**0.1869**	0.0933	0.1734
Removed	**0.0983**	**0.1608**	0.0851	0.1516

(a) Parallel Corpus 2

The mapping results suggest that experiments using collections including the *stopwords* yield better results than using those which exclude the *stopwords*. It is believed that *stopwords* are not bounded by semantic domains, thus do not carry any semantic bias. But, on account of the small size of collections, in coincidence, *stopwords,* which consistently appear in a specific domain, may carry some semantic information about the

domain. Additionally, the results suggest that although LSA comfortably outperforms random baseline, frequency baseline seems to perform better than LSA.

5.1.7 Effect of Target Term Mapping Selection

There are four mapping selections used to determine the mapping results for each experiment, i.e. selecting the top 1, 10, 50, and 100 terms with the highest similarity values. Table 5.9 shows the effect of varying mapping selection to the mapping results for top score set of parallel corpus 1 and 2 are given (See Appendix B.1.2, B.1.3and B.1.4).

Table 5.9 Effect of Varying Target Term Mapping Selection

Mapping Selection	RNDM		FREQ		LSA	
	P	R	P	R	P	R
Top 1	0.0000	0.0000	**0.3333**	**0.1496**	0.2380	0.0987
Top 10	0.0002	0.0002	**0.0477**	**0.2021**	0.0434	0.1733
Top 50	0.0003	0.0025	**0.0143**	**0.2689**	0.0133	0.2338
Top 100	0.0004	0.0069	**0.0083**	**0.2935**	0.0081	0.2732

(a) Parallel Corpus 1

Mapping Selection	FREQ		LSA	
	P	R	P	R
Top 1	**0.3153**	**0.0911**	0.2718	0.0764
Top 10	**0.0656**	**0.1732**	0.0586	0.1498
Top 50	**0.0177**	**0.2097**	0.0169	0.2021
Top 100	**0.0098**	0.2214	0.0095	**0.2218**

(b) Parallel Corpus 2

Precision is the number of mapping results or translation pairs, which match the pairs in bilingual dictionary, divided by the number of mapping selection. Thus, as the number of translation pairs selected increases, the precision value decreases. On the other hand, recall is the number of translation pairs, which match the pairs in bilingual dictionary, divided by the number of pairs given by the bilingual dictionary itself. As the number of translation pairs selected increases, the possibility to find more pairs matching the pairs in bilingual dictionary increases. Thus, the recall value increases as well. These intuitions are confirmed by the results shown in Table 5.9.

5.1.8 Effect of Text Alignment Granularity

Most word alignment systems employ corpora which are aligned down to the sentence level, e.g. (Deng & Gao, 2007). The intuition suggests that by using finer-grained segmentation of the parallel texts, LSA will be able to capture more specific context of term-usage. Thus LSA will perform better than using coarser-grained text segmentations. Some experiments using parallel corpus 2 were carried out to verify this intuition.

From parallel corpus 2, only the largest collection containing 3500 verses was used. These verses comprise 115 bible chapters. Verse is considered as fine-grained text segmentation, whereas chapter is as coarse-grained. A verse usually contains no more than two sentences. On the other hand, a chapter usually contains about 30 verses, i.e. the average number of verses in a chapter is 30.38.

The idea is to compare LSA performance between experiments using the collection of verses and the collection of chapters under the same variable configurations. The experiments were conducted for the top score set of parallel corpus 2. The removal of *stopwords* was applied, but no weighting is applied on the term-document matrices. For each collection, the rank approximations are 10%, 25%, and 50% the number of dimensions of the original collection. Thus, for the collection of verses, the 350, 875, and 1750-rank approximations are computed. For collection of chapters, 12, 29, and 58-rank approximations are computed.

Table 5.10 Sample of LSA Experiment Results using Chapters and Verses

English Term: *name*		
Bilingual dictionary	**LSA using Chapters**	**LSA using Verses**
mencaci seseorg asma memanggil ***menamai*** menamakan ***menyebut*** menyebutkan ***nama***	***menamai*** melahirkan beberapa anak sebab sangat dilahirkan telah ketika ***nama***	***nama*** sembarangan namanya ***menamai*** hormat demi membusukkan ***menyebut*** melayani terlalu
Precision	**0.200**	**0.300**
Recall	**0.250**	**0.375**

Table 5.10 shows comparison of mapping results given by LSA experiments using chapters and verses for English term *name*. Of the eight terms given by the bilingual dictionary, LSA using chapters matches two terms, whereas LSA using verses matches three terms.

Table 5.11 Effect of Text Alignment Granularity

Text Alignment Granularity	LSA	
	P	R
Verses	0.1277	0.2156
Chapters	0.0862	0.1766

Table 5.11 gives the summary of mapping result showing the effect of text alignment granularity. LSA experiments using verses yield higher precision and recall values than those using chapters. Thus, it confirms the intuition that using finer-grained segmentation of the parallel texts improves the result of bilingual term mapping.

5.1.9 Discussion

Bilingual term mapping is not the primary objective of this work, but bilingual concept mapping. It was not conceived until the results of some initial bilingual concept mapping experiments were produced. Having examined manually, the results of initial bilingual concept mapping experiments seemed poor and inconclusive. Hence, bilingual term mapping was formulated with the intention of helping to examine the LSA capability of carrying out the concept mappings.

The bilingual term mapping was explored extensively with a variety of variables with the purpose of finding out the optimal LSA configuration for conducting concept mappings. However, the bilingual term mapping results using LSA were unsatisfactory. The task of bilingual term mapping may even be harder to ask of LSA than that of bilingual concept mapping due to its finer alignment granularity. While concept mapping attempts to map a concept conveyed by a group of semantically related terms, term mapping attempts to map a term with a specific meaning to its translation in another language.

The results of bilingual term mapping should be evaluated with a resource which arranges list of terms in terms of semantic relatedness. However, such a resource is not available in this work. Conversely, what can be done is computing precision and recall of the term mapping results with bilingual dictionary. It might be similar with comparing the results

to the resource listing semantically related terms. Nevertheless, it is different enough to be a problem.

In theory, LSA makes use of rank reduction to remove noise and to extract underlying information contained in a corpus. According to the experiment results, however, generally frequency baseline, which employs full rank term-document matrix, seems to perform better than LSA. It is speculated that LSA is good to discover general pattern like clustering. For example, given a corpus with a variety of domains, LSA should be able to discern the domains. But, it may be very difficult for LSA to distinguish parts of a very specific domain. The rank reduction may perhaps remove the important details, so that the differences turn out blurred.

The LSA failure compared to frequency baseline results perhaps because frequency baseline concerns more about co-occurrence than LSA. It compares term vectors between English and Indonesian which contain pure frequency of term occurrence in each document. On the other hand, LSA concerns more about semantic relatedness. It compares English and Indonesian term vectors containing term frequency estimation in documents according to the context-meaning.

Since the purpose of bilingual term mapping is to obtain proper translations for an English term, it may be better explained as an issue of co-occurrence rather than semantic relatedness. That is, wherever an English term appears in an English document, the Indonesian translation should also appear in the corresponding document. The higher the frequency of co-occurrence between an English term and an Indonesian term, statistically, the higher the probability that they are translations of each other.

Probably LSA may yield better results in the case of finding terms with similar semantic domain. The LSA mapping results should be better assessed using a resource listing semantically related terms, rather than using bilingual dictionary listing translation pairs. Evaluating mapping results with bilingual dictionary may be quite similar to evaluating semantic relatedness. Nevertheless, it is different enough to be a problem. Bilingual dictionary restricts that the mapping results should be the translations for an English term. It demands more specific constraints than semantic relatedness.

Furthermore, polysemous terms may become a problem for LSA. By rank approximation, LSA estimates the occurrence frequency of a word in a particular document. Since

polysemy of English terms and Indonesian terms can be quite different, the estimations for terms which are mutual translation can be different.

For instance, *kali* and *waktu* are Indonesian translations for the English term *time*. Moreover, *kali* is also the Indonesian translation for the English term *river*. Suppose *kali* and *time* appear frequently in documents about multiplication, but *kali* and *river* appear rarely in documents about river. Then, *waktu* and *time* appear frequently in documents about time. As a result, LSA may estimate *kali* with greater frequency in documents about multiplication and time, but with lower frequency in documents about river. The term vectors between *kali* and *river* may not be similar. Thus, in bilingual term mapping, LSA may not suggest *kali* as the proper translation for *river*.

Polysemous terms can also be a problem for frequency baseline. However, since frequency baseline merely uses the raw term frequency vectors, the problem does not affect other term vectors. LSA, conversely, exacerbates this problem by taking it into account in estimating other term frequencies.

In general, this problem may also be caused by misaligned English-Indonesian article pairs. That is, terms which are not significant to reflect a context but appear numerously in documents about that context, may impair LSA estimation.

5.2 Bilingual Concept Mapping Experiment

The initial steps of bilingual concept mapping are akin to that of bilingual term mapping. After defining a collection of document pairs, a term-document matrix can be built from it, with or without removing the *stopwords*. Then, weighted term-document matrices can also be built using TF-IDF and Log Entropy weighting. LSA is subsequently applied to a term-document matrix so that an LSA matrix is created.

For each English and Indonesian concept, a set of textual context and the corresponding conceptual semantic vectors were constructed using the LSA matrix. Rather than term vectors, these conceptual semantic vectors were compared. For each English concept, some of the most similar Indonesian concepts are taken as the equivalent concepts. (See Section 3.4)

5.2.1 Existing Resources

For bilingual concept mapping experiments, parallel corpus 1 and the bilingual dictionary are used (see Section 5.1.1). Additionally, some other resources used are:

- **WordNet**

  The most recent version of Princeton WordNet, version 3.0, is used as the English lexicon. The application and documentation of Princeton WordNet are available at http://wordnet.princeton.edu/. In practise, the SQL version of Princeton WordNet version 3.0 is used, which is obtained from http://wnsqlbuilder.sourceforge.net. This SQL version contains 117659 distinct *synsets*. For each *synset*, the set of terms belonging to the *synset*, the glossary, and examples sentences are used. The union of these resources yields 169583 unique terms.

- **Kamus Besar Bahasa Indonesia**

  The electronic version of Kamus Besar Bahasa Indonesia (KBBI), i.e. an Indonesian machine readable dictionary, is used as the Indonesian lexicon. It was developed at the Faculty of Computer Science, Universitas Indonesia, during the mid-90s. It contains 85521 distinct word sense definitions. For each word sense definition, the sublemma, i.e. inflected word, phrase, or idiomatic expression, along with the definition and example sentences are used. The union of these resources yields 87171 unique terms.

- **List of common-based concepts**

  Based on the results of Euro WordNet and BalkaNet, the Global WordNet Association[3] provides base concepts which play the most important role in the various WordNets of different languages. For bilingual concept mapping experiments, concepts that act as Base Concepts in at least two languages are taken as the English common-based concepts to be mapped to their Indonesian concepts suggested from bilingual dictionary. These English common based concepts consist of 3547 distinct concepts.

[3] www.globalwordnet.org

5.2.2 Variables

Since a large collection size always yields better results for bilingual term mapping, only the largest collection of parallel corpus 1, P_{1000}, was used in bilingual concept mapping experiments. The same rank approximations, the removal of *stopwords* and the weighting schemes are applied. Other variables used for bilingual concept mapping experiments are:

- **Source Concepts Selection**

  For the bilingual concept mapping, a hundred English concepts with the highest score are selected to be mapped. The concept score is computed by summing the cumulative frequency of each term in its textual context set. In other way, the English concepts to be mapped are selected according to the common-based concept list obtained from Global WordNet (see Section 5.2.1).

- **Target Concept Mapping Selection**

  For bilingual concept mapping, several mapping selections are used to determine the Indonesian concepts as the equivalent concepts for an English concept. Those mapping selections include:

  1. **Average Threshold.** Firstly, the average of the similarity values of all Indonesian concepts is computed. Then, each Indonesian concept with similarity value above the average value is designated as the equivalent concept for the corresponding English concept.
  2. **MinMax *m%* Threshold.** The difference between the maximum and the minimum similarity value is computed. Then, for each English concept, Indonesian concepts with similarity values above the minimum value plus *m%* of the difference value are designated. Specifically, MinMax 10%, 25%, and 50% threshold are used.
  3. **Top *n*.** Like the bilingual term mapping selection, top *n* Indonesian concepts with the highest similarity values are designated for an English concept. Specifically, the top 1, 3, and 5 concepts are designated.

5.2.3 Sample Bilingual Concept Mapping Result

In WordNet, English concepts are represented as *synsets* and identified by their synset IDs. The ideal textual context set for an English concept contains all information about its *synset* in WordNet, i.e. the *words*, the *gloss*, and the *example sentences*. On the other

hand, the actual textual context set consists of the set of terms in *words*, the set of terms in *gloss*, and the set of terms in *example sentences* that appear in the collection. Figure 5.1 exemplifies the ideal textual context set for an English concept along with its actual textual context.

Similarly, Indonesian concepts are identified by their KBBI IDs. The ideal textual context set for an Indonesian concept contains its *sublemma*, *definition*, and *example sentences*. Figure 5.1 presents both ideal and actual textual context set of Indonesian concepts returned by LSA for the English concept.

WordNet Synset ID: 400060939
Words: before, earlier
Gloss: earlier in time, previously
Example: i had known her before, as i said before, he called me the day before but your call had come even earlier, her parents had died four years earlier, i mentioned that problem earlier

Actual Textual context set: {{earlier,before},{earlier,previously,time},{even,come, problem, years,day,call,known,earlier,i,called,before,mentioned,said,me,your,died, parents, four}}

Mapping Results

1. **KBBI ID**: k59827- **Similarity**: 0.594
   Sublemma: oleh
   Definition: kata penghubung yg dipergunakan untuk menandai pelaku
   Example: rumah ini dibeli oleh ayah bulan lalu, tidak teringat oleh ibu bahwa hari ini hari ulang tahun adik

   Actual Textual context set: {{}, {kata,yg,dipergunakan,pelaku,menandai, penghubung},{hari,ulang,tahun,dibeli,bulan,rumah,ibu,adik,ayah}}

2. **KBBI ID**: k18468- **Similarity**: 0.589
   Sublemma: dan
   Definition: penghubung satuan ujaran kata frase klausa dan kalimat yg setara yg termasuk tipe yg sama serta memiliki fungsi yg tidak berbeda
   Example: ayah dan ibu bibi dan paman serta para anak cucu dan kemenakan bersama-sama merayakan 50 tahun perkawinan nenek mereka

   Actual Textual context set: {{ }, {kata,setara,frase,kalimat,tipe,termasuk,yg,memiliki,berbeda,satuan, fungsi, penghubung},{anak,kemenakan,perkawinan,tahun,nenek,paman,cucu,ibu, merayakan,mereka,ayah }}

Figure 5.1 Sample of Bilingual Concept Mapping Result

The sample of concept mapping results given in Figure 5.1 was taken from an experiment using P_{1000} which excludes the *stopwords*, and LSA matrix with 25% rank approximation. The experiment mapped a hundred of English concepts with the highest scores were selected by computing the sum of the cumulative frequencies of their term vectors. These English concepts are referred to as a top score set.

LSA does not map the English concept conveyed by the terms *before* and *earlier* to the Indonesian equivalent concept *sebelum*. Generally, this failure is believed due to lack of semantic context provided by the collection size. Although the actual textual context for the English concept is quite large, the textual context set of Indonesian concepts do not contain the terms in *words*, whereas it is believed those terms play the most important role to define the context. This is reflected in the composition of weighting in constructing the conceptual semantic vectors, i.e. 60% for *words*, 30% for *definition*, and 10% for *example*.

Since *oleh* is one of the *stopwords* for Indonesian, the actual textual context set for the corresponding Indonesian concept does not include the term *oleh*. Nevertheless, LSA seems to find some similarity between the Indonesian and the English concepts. Notice that, actual textual context set for the Indonesian concept contains the terms *hari* and *bulan*, while the set for the English concept contains the semantically related terms *years* and *day*. Moreover, the Indonesian set contains the terms *ibu*, *adik*, *ayah*, while the English set contains the terms *parents*. These terms may be located near each other, thus the conceptual semantic vectors between the Indonesian and the English concepts may also be near each other, causing LSA to induce incorrect mappings.

The result also suggests that the task of choosing the correct Indonesian concepts from numerous Indonesian concepts is too much to ask of LSA. Thus, this task was simplified by firstly taking another set containing only the English common-based concepts to be mapped. Then, for each English concept, LSA is expected to select the most appropriate Indonesian concepts from a subset of concepts that have been derived based on their *words* appearing in the bilingual dictionary. These specific concepts are called *suggestions*.

For instance, instead of comparing the vector representing *communication* with every single Indonesian concept in the KBBI, in this task, it is compared against *suggestions* with a limited range of sublemmas, e.g. *komunikasi*, *perhubungan*, *hubungan*, etc. This

setup is thus identical to that of an experiment to manually map WordNet *synsets* to KBBI senses (Darma Putra, Arfan, & Manurung, 2008). Consequently, this facilitates assessment of the results by computing the Fleiss Kappa values to show the level of agreement between the LSA-based mappings with human annotations.

In general, Fleiss Kappa (Fleiss, 1971) measures level of agreement between independent judges, where the value 1 signifies perfect agreement between the judges and the value less than 1 signifies the opposite. In the case of bilingual concept mapping, Fleiss Kappa value represents both agreement for choosing the same concepts as the correct mapping and for not choosing other concepts as the correct mapping.

WordNet Synset ID: 100319939
Words: chase, following, pursual, pursuit
Gloss: the act of pursuing in an effort to overtake or capture
Example: the culprit started to run and the cop took off in pursuit

Textual context set: {{following, chase}, {the, effort, of, to, or, capture, in, act, pursuing, an}, {the, off, took, to, run, in, culprit, started, and}}

Mapping Results

1. **KBBI ID**: k39607 - **Similarity**: 0.804
   Sublemma: mengejar
   Definition: berlari untuk menyusul menangkap dsb memburu
   Example: ia berusaha mengejar dan menangkap saya

   Textual context set: {{mengejar}, {memburu, berlari, menangkap, untuk, menyusul}, {berusaha, dan, ia, mengejar, saya, menangkap}}

2. **KBBI ID**: k14029 - **Similarity**: 0.781
   Sublemma: memburu
   Definition: mengejar untuk menangkap binatang dl hutan dsb
   Example: memburu di daerah suaka margasatwa adalah terlarang

   Textual context set: {{memburu}, {mengejar, hutan, menangkap, binatang, untuk}, {adalah, memburu, suaka, di, terlarang, daerah, margasatwa}}

Figure 5.2 Example of Successful Mapping for English Common-Based Concept

Using the new setup, two examples of successful and unsuccessful mappings employing P_{1000} and LSA matrix with 50% rank approximation are given in Figure 5.2 and Figure 5.3, respectively. In total, there were 144 English common-based concepts or *synsets* which have been mapped manually by at least 2 human judges, thus enabling the LSA-based mapping results to be compared with the human judgements.

According to the example in Figure 5.2, LSA is able to correctly map an English concept to its equivalent Indonesian concepts. The English concept conveyed by the terms *chase*, *following*, *pursual*, and *pursuit* are correctly mapped to Indonesian concept conveyed by the term *mengejar*. Additionally, LSA suggests similar Indonesian concept conveyed by the term *memburu*.

WordNet synset ID: 201277784
Words: crease, furrow, wrinkle
Gloss: make wrinkled or creased
Example: furrow one's brow

Textual context set: {{}, {or, make}, {s, one}}

Mapping Results

1. **KBBI ID**: k02421 - **Similarity**: 0.69
   Sublemma: alur
   Definition: jalinan peristiwa dl karya sastra untuk mencapai efek tertentu pautannya dapat diwujudkan oleh hubungan temporal atau waktu dan oleh hubungan kausal atau sebab-akibat
   Example: (none)

   Textual context set: {{alur}, {oleh, dan, atau, jalinan, peristiwa, diwujudkan, efek, dapat, karya, hubungan, waktu, mencapai, untuk, tertentu}, {}}

2. **KBBI ID**: k26302 - **Similarity**: 0.688
   Sublemma: gelugur
   Definition: pohon mangga hutan buahnya berwarna merah kekuningkuningan dipakai untuk mengasami gulai garcinia macrophylla
   Example: (none)

   Textual context set: {{}, {mangga, berwarna, merah, hutan, pohon, dipakai, untuk}, {}}

Figure 5.3 Example of Unsuccessful Mapping for English Common-Based Concept

The textual context sets for both English and Indonesian concepts in Figure 5.2 are considerably large and hence provide adequate contexts for LSA to choose correct Indonesian concepts. This fact suggests that LSA is able to show some measure of semantic information provided by the adequate textual context sets.

However, when the textual context set is very limited, the result is likely to be poor. For example, Figure 5.3 shows that the English concept conveyed by the terms *crease*, *furrow*, and *wrinkle* is incorrectly mapped to the Indonesian concept conveyed by the term *alur*. The other Indonesian concept suggested by LSA is conveyed by the term *gelugur*. Neither of these concepts is semantically related to the English concept.

The actual context set for the English concept is sparsely filled with the terms *or*, *make*, *s*, and *one*. These terms are undoubtedly insufficient to explain the concept. As a result, LSA is unable to identify correct Indonesian concepts.

5.2.4 Comparison of Level of Agreement

As mentioned in Section 5.2.3, the task of mapping English common-based concepts to their suggestion is identical to that of manual mapping experiments conducted by (Darma Putra, Arfan, & Manurung, 2008). Thus, the level of agreements between LSA-based mapping results and human annotations can be assessed by computing the Fleiss Kappa value for each English concept or *synset*.

Table 5.12 Comparison of Level of Agreement

Judges	Synsets	Fleiss Kappa Values					
		Judges only	Judges + RNDM1	Judges + FREQ Top 1	Judges + LSA 10% Top1	Judges + LSA 25% Top1	Judges + LSA 50% Top1
≥ 2	144	0.4269	0.1744	0.2140	0.2092	0.2109	0.2099
≥ 3	24	0.4651	0.2536	0.2762	0.2762	0.2762	0.2762
≥ 4	8	0.5765	0.3643	0.3593	0.3593	0.3593	0.3593
≥ 5	4	0.4639	0.3254	0.3308	0.3308	0.3308	0.3308
	Average	0.4831	0.2794	0.2951	0.2939	0.2943	0.2941

Judges	Synsets	Fleiss Kappa Values					
		Judges only	Judges + RNDM3	Judges + FREQ Top 3	Judges + LSA 10% Top3	Judges + LSA 25% Top3	Judges + LSA 50% Top3
≥ 2	144	0.4269	0.1318	0.1667	0.1544	0.1606	0.1620
≥ 3	24	0.4651	0.2197	0.2282	0.2334	0.2239	0.2185
≥ 4	8	0.5765	0.3103	0.2282	0.3615	0.3329	0.3329
≥ 5	4	0.4639	0.2900	0.2297	0.3359	0.3359	0.3359
	Average	0.4831	0.2380	0.2132	0.2713	0.2633	0.2623

Table 5.12 presents a comparison of level of agreement between human judges, human judges and random baselines denoted as **RNDM**, human judges and frequency baselines denoted as **FREQ**, and human judges and LSA-based mappings. For **RNDM 1**, a suggested Indonesian concept is selected randomly for each English concept. On the other hand, **RNDM 3** selects three distinct suggested Indonesian concepts randomly for each English concept. If there are less than three suggestions, **RNDM 3** only gives all of

the existing suggestions. Additionally, **RNDM** is actually represents average of Fleiss Kappa values for over 10 runs.

Frequency baseline for bilingual concept mapping compares English common-based conceptual semantic vectors to their suggestion semantic vectors based on full rank term-document matrices. That is, a conceptual semantic vector is constructed by averaging its term vectors taken from a full rank term-document matrix. Then, the top 1 and 3 Indonesian concepts with the highest similarity values are designated as the mapping results. If there are less than three suggestions, i.e. suggested Indonesian concepts derived from bilingual dictionary, for an English concept, frequency baseline only gives the existing suggestions as mapping results for Top 3.

The LSA-based mappings were divided into three different experiments according to the rank approximation. These experiments used term-document matrices without any weighting. Like frequency baselines, the top 1 and 3 Indonesian concepts are taken as the mapping results for each experiment.

The Fleiss Kappa values were computed for each English concept which has already been mapped manually. Table 5.12 presents the averages of Fleiss Kappa values according by the number of human judges which have mapped the English concepts. For example, there are 144 English concepts which have been map by at least two human judges.

For top 1 selection, the averages of level of agreement between human judges and LSA-based mappings with different rank approximations do not differ a lot. They are better than random baseline, which suggests that LSA is indeed capturing bilingual semantic information implicit within the parallel corpus. But, the average level of agreement between frequency baseline and human judges is even better than LSA.

On the other hand, for top 3 selection, the average level of agreement between human judges and frequency baseline decreases significantly, which is even worse than random baseline. The averages of level of agreement between human judgement and LSA with different rank approximations are not as high as between the human judges themselves and the absolute values are lower than for top 1. Nevertheless, it is much better than random baseline and frequency baseline. Using top 3 selection, LSA mappings with 10% rank approximation yields higher levels of agreement than LSA with other rank approximations.

5.2.5 Effect of Target Concept Mapping Selection

The LSA-based mapping results vary due to the variety of methods used to determine the Indonesian concepts as the equivalent concepts for an English concept. methods include Average threshold, MinMax 10%, 25%, and 50% thresholds, and Top 1, 3, and 5 (see Section 5.2.2).

Table 5.13 presents the Fleiss Kappa values between human judges and LSA using each variation of the mapping selections. The mapping results are taken from LSA experiments using P_{1000} and 50% rank approximation.

Table 5.13 Effect of Target Concept Mapping Selection

Judges	English Concepts	Average	MinMax 10%	MinMax 25%	MinMax 50%	Top 1	Top 3	Top 5
≥ 2	144	0.1458	0.1458	0.1005	0.1398	0.2099	0.1620	0.1220
≥ 3	24	0.2399	0.2399	0.1952	0.2336	0.2762	0.2185	0.1945
≥ 4	8	0.3481	0.3481	0.2689	0.3564	0.3593	0.3329	0.2976
≥ 5	4	0.3308	0.3308	0.2594	0.3475	0.3308	0.3359	0.2788
Average		0.2661	0.2661	0.2060	0.2693	0.2941	0.2623	0.2232

Predominantly, LSA using Top 1 mapping selection yields higher levels of agreement than other mapping selections. The reason is because the human judges are likely to choose one Indonesian concepts for each English concept. The average number of Indonesian concepts mapped by a human judge per English concept is 1.36. Fleiss Kappa represents both agreement of choosing the same concepts as the correct mapping and not choosing other concepts as the correct mapping. Since LSA using Top 1 mapping selection only selects one Indonesian concept, the level of agreement with human judges for not choosing the other concepts as the correct mapping should be high.

5.2.6 Effect of Frequency Weighting

Table 5.14 shows the effect of weighting to the level of agreement between human judges and LSA. Two weighting schemes, TF-IDF and Log Entropy, are used. The weighting is applied on a term-document matrix before computing LSA. The mapping results used to compute the Fleiss Kappa are taken from LSA experiments using P_{1000} and 50% rank approximation.

Table 5.14 Effect of Weighting

Judges	English Concepts	Judges + LSA No Weighting	Judges + LSA TF-IDF	Judges + LSA Log Entropy
≥ 2	144	0.1466	**0.2133**	0.1957
≥ 3	24	0.2283	**0.2820**	0.2802
≥ 4	8	0.3302	**0.3744**	0.3655
≥ 5	4	**0.3163**	0.2807	0.3049
	Average	0.2553	**0.2876**	0.2866

LSA with TF-IDF weighting is inclined to yield the highest average level of agreement than others. The average level of agreement between human judges and LSA with Log Entropy weighting is slightly lower than TF-IDF. But, it is still considerably higher than no weighting.

5.2.7 Effect of Frequency Weighting and Concept Mapping Selection

Generally, the effect of concept mapping selection and the effect of weighting have been described in Section 5.2.5 and Section 5.2.6, respectively. This section describes the weighting usage in terms of concept mapping selection.

Table 5.15 Effect of Weighting and Concept Mapping Selection

Weighting Usage	Average	MinMax 10%	MinMax 25%	MinMax 50%	Top 1	Top 3	Top 5
No Weighting	**0.2661**	**0.2661**	0.2060	0.2693	0.2941	0.2623	0.2232
TF-IDF	0.2653	0.2211	**0.2525**	0.2789	**0.2942**	0.2774	0.2282
Log Entropy	0.2630	0.2145	0.2274	**0.2800**	0.2896	**0.2944**	**0.2407**
Average	0.2648	0.2339	0.2287	0.2761	0.2926	0.2781	0.2307

Table 5.15 shows the average of Fleiss Kappa values, regardless of the number of judges. The Fleiss Kappa values are taken from LSA mapping results using P_{1000} and 50% rank approximation only. The numbers in bold indicate the highest average of level of agreement for a mapping selection. It suggests the appropriate weighting usage according to the mapping selection used. For Average threshold and MinMax 10%, LSA performs optimally with term-document matrix without any weighting. On the other hand, for MinMax 25% and Top1, LSA performs better using term-document matrix with TF-IDF weighting. Lastly, for MinMax 50%, Top 3, and Top 5, LSA performs better using term-document matrix with Log Entropy weighting.