AUTOMATED COHERENCE DETECTION WITH TERM-DISTANCE PATH EXTRACTION OF THE CO-OCCURRENCE MATRIX OF A DOCUMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALİL AĞIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

AUGUST 2015

Submitted by **Halil AĞIN** in partial fulfillment of the requirements for the degree of **Master of Science in the Department of Cognitive Science**, **Middle East Technical University by**,

Prof. Dr. Nazife Baykal _____
Director, Informatics Institute

Prof. Dr. H. Cem Bozşahin _____
Head of Department, Cognitive Science

Assist. Prof. Dr. Cengiz Acartürk _____
Supervisor, Cognitive Science

Examining Committee Members

Prof. Dr. H. Cem Bozşahin _____
Cognitive Science, METU

Assist. Prof. Dr. Cengiz Acartürk _____
Cognitive Science, METU

Prof. Dr. Deniz Zeyrek Bozşahin _____
Cognitive Science, METU

Assist. Prof. Dr. Murat Perit Çakır _____
Cognitive Science, METU

Assist. Prof. Dr. Murat Ulubay _____
Department of Management, YBU

Date: August 14, 2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HALİL AĞIN

Signature :

ABSTRACT

AUTOMATED COHERENCE DETECTION WITH TERM-DISTANCE PATH EXTRACTION OF THE CO-OCCURRENCE MATRIX OF A DOCUMENT

AĞIN, Halil

M.Sc., Department of Cognitive Science

Supervisor : Assist. Prof. Dr. Cengiz Acartürk

August 2015, 100 pages

This thesis takes the distributional semantics (frequency-based semantics) approach as the theoretical framework to quantify textual coherence. Distributional semantics describes discourse sections as vectors, having dimensions are the frequency count of co-occurring words in the text within its semantic space. It quantifies the textual coherence by measuring the cosine values of vectors of successive sentences (cf. Latent Semantic Analysis, LSA). The common assumption underlying LSA based studies is that the frequency of word co-occurrence can be used as a cohesive cue to quantify textual coherence, thus leading to analyses based on a term-document matrix. In this thesis, the spatial distance of co-occurring words is considered as a new frequency event of cohesive cues and introduces a document-distance matrix, which is derived from the term-document matrix. This thesis proposes that the matrix representation of document-distance (a derivation of term-document matrix) of co-occurring words in adjacent sentences in a text can be used to quantify textual coherence. Two mathematical functions are suggested for deriving the document-distance matrix and two algorithms for the operations. The mathematical functions operate on the document-document matrix (a derivation of term-document matrix) to derive the document-distance matrix. The algorithms measure the coherence of text by operating on the newly introduced document-distance matrices.

Keywords: Distributional Semantics, Co-occurrence Matrix, Document-Distance Matrix, Latent Semantic Analysis, Coherence

ÖZ

BİR DOKÜMANIN TEKRAR MATRİSİNİN KELİME-MESAFE YOLU ÇIKARIMI İLE OTOMATİK METİN TUTARLILIĞI TESPİTİ

AĞIN, Halil
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Assist. Prof. Dr. Cengiz Acartürk

Haziran 2015, 100 sayfa

Bu tez, metinsel tutarlılığı ölçmek için dağılımsal anlambilimini teorik çerçeve olarak kabül etmektedir. Dağılımsal anlambilimi söylem sekmelerini vektör olarak alır ve vektör boyutlarını metindeki tekrarlı kelime sayılarından oluşturur. Bu sayede metinin anlam darağacının oluştu rulmasını sağlar. Metinsel tutarlılık bu vektörlerin cosine değerleri hesaplanarak ölçülür (Gizil Anlambilim analizi, LSA). Bu çalışmalardaki ortak varsayım metin tutarlılığını ölçmek için metindeki tekrarlanan kelime frekansları bir kohezif ip ucu olarak kullanılabilir. Böylece, kelime-doküman matrisleri temelli analizlere kapı aralanmış olur. Bu tez, bir metinde ardışık cümlelerdeki tekrar eden kelimelerden elde edilen kelime-mesafe matrisinin (kelime-doküman matrisinin bir türevi) metin tutatlılığının ölçümünde kullanılabileceğini ileri sürmektedir. Tez, do-küman-mesafe matrisinin elde edilebilmesi için 2 adet matematiksel fonksiyon ve fonksiyonları kullanan 2 adet algoritma önermektedir. Matematiksel fonksiyonlar doküman-doküman matrisinden doküman-mesafe matrisini üretmek için kullanılmaktadır. Algoritmalar, yeni önerilen doküman-mesafe matrisi uzerinde işleyerek metinsel tutarlılığı ölçmektedir.

Anahtar Kelimeler: Dağıtımsal anlambilimi, Gizil Anlam Analizi, Metin tutarlılığı, Kelime Tekrar Matrisi, Doküman-Mesafe Matrisi

to my parents...

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Assist. Prof. Dr. Cengiz Acartürk for his kind support, helpful criticism and guidance. Without his support, it would not have been possible to even start this study.

I would like to express my gratitude to Assist. Prof. Dr. H. Cem Bozşahin, Prof. Dr. Deniz Zeyrek Bozşahin, Assist. Prof. Dr. Murat Perit Çakır and Assist. Prof. Dr. Murat Ulubay for their notable contributions.

I am grateful to all the members of Cognitive Science Department who enlightened me about the phenomena happening around me.

I thank my friend Gökhan Gönül who has supported me, tried to help me and answered all my questions. Furthermore, Chapter 3 was begun with his guidance.

Last but not least, I am immensely grateful to my parents for their excellent support. Surprisingly, they were more excited about the progress of writing this thesis than I was.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This thesis proposes that the distance between re-occurring words in adjacent sentences can be used to measure the degree of coherence. The comprehension of a text is a qualitative production of the human mind. The current state of the art on the measurement of coherence is not based on observing the neurons of a brain and their activities while comprehending the text. Instead, it is based on the analysis of observable items (cohesive cues) which indicate an unobservable phenomenon (coherence). For instance, Halliday and Hasan (1976) categorized cohesive cues as reference, substitution, ellipsis, conjunction, and lexical cohesion (Cf. Figure-1.1). Coherence is achieved when reading results in a holistic understanding of the text. This happens when the reader builds a situational model of the textbase at the end of the reading process and this is strongly dependent on the well-designed organization of the cohesive cues (Halliday and Hasan, 1976). The cohesives listed in Figure-3.1 can be divided into two sections; syntactic based (reference, substitution, ellipsis, conjunction) and lexical based.

The cohesive cues that are based on syntax and lexis constitute the research objects of two theoretical frameworks namely the Compositional (Denotational) Semantics and Distributional Semantics, respectively. Both frameworks focus on different types of cohesive cues to respond to the question: How are cohesive cues related to the mental representation of the reader? Although these theoretical frameworks have different assumptions, they emanate from the same research question: What is meaning? Both approaches try to explain meaning through the surface structure of language. Since this study takes the distributional hypothesis as a guiding hypothesis, the aim of this chapter is to provide a brief summary of the literature to position the thesis within the existing literature.

1.1 History of theory of meaning

Coherence is a construction of the mind achieved when a person reads text. It usually indicates a well-formed mental representation of concepts. Kintsch and Van Dijk (1978) identified the three layers of mental representation as:

Representation in linguistic system / *Type of cohesive relation*	Semantic	Lexicogrammatical (typically)
Conjunction	Additive, adversative, causal and temporal relations; external and internal	Discourse adjuncts: adverbial groups, prepositional groups
Reference	Identification:	
	by speech role	Personals
	by proximity	Demonstratives
	by specificity (only)	Definite article
	Reference point	Comparatives
Lexical cohesion	Collocation (similarity of lexical environment)	Same or associated lexical item
	Reiteration (identity of lexical reference)	Same lexical item; synonym; superordinate; general word
Substitution	Identity of potential reference (class meaning) in context of non-identity of actual (instantial) reference	Verbal, nominal or clausal substitute Verbal, nominal or clausal ellipsis

Figure 1.1: Cohesion categories (Halliday and Hasan 1976, p.324).

1. Surface structure (actual wording of the text)
2. Textbase (proposition units) (Kintsch, 1988)
3. Situational model (The scenario in the text)

Kintsch states that situational model is constructed while reading, and it stores more information than what has been read. This is the point where the philosophical question arises which might be attributed to Plato in which he asks how we know more than we have been taught. (Stanford et al., 1835). Chomsky (1988) calls this question *Plato's problem*, and introduces two fictitious words as examples: *strid* and *bnid*.

According to Chomsky, strid, could be accepted as an English word by a native English speaker although they may have never heard it before. For the word *bnid*, although the native speaker has never heard it before, he knows that it is not an English word because of the sound structure. Thus, Chomsky shows that we know something about *strid* although we know nothing about it.

A relevant question was posed by Quine (1970) concerning meaning of a word: can we translate a word to another language and maintain its precise meaning? Lycan (2008) gives a pseudo-scenario for a Dutch language learner who learns the meaning of *groot* (see Appendix-A.1 for details). According to the scenario, the translation becomes problematic due to the intrinsic meaning of the word that was subject to translation (Harnad, 1990).

This problem is also called the "Symbol grounding problem" and challenged by Searle's Chinese room argument (1990). The meaning of a word is constructed individually, and not derived from symbols. It is intrinsic to the individual, which construed as the meaning always being there and cannot be shared or compared.

Since the meaning is holistic, qualitative and not observable, scholars introduced various theories for example, John Stuart Mill stated that all meanings have a reference to an entity in the world and Mill's theory is known as the "Reference theory of meaning".

The Mill's word-entity mapping was challenged by Frege (1892) who introduced the concept of "sense and reference" (Sinn und Bedeutung) (see Appendix A.2 for details). Frege argued that each sentence has a truth value that is composed of propositional statements in the sentence; therefore the meaning of a sentence is its truth-value. He rejected Mill's Referential theory through the use of tautology and truth values of sentences (see Appendix A.2 for details). Although it seems that Frege's 'sense' is like individual ideas or mental images as in Aristotle and Locke, this sense of an expression is part of thought (Riemer, 2010). For example, the sense of *Morning star* indicates a star visible in the morning. It is the same star but it is not subjective and has different mode of expression which is independent from the referent (Riemer, 2010). The expressions *Morning star* and *stars visible in the morning* are conceptually referent of an entity of a word or a sentence, where this referent is a sort of abstraction which may remain forever as a proposition.

In Frege's Theory of meaning, there is no contextual information (this is also true for Mill's Referential Theory of Meaning). It is sentence bounded, and each word in a sentence has to denote a 'thing' and the composition of the statement will denote the truth value of the sentence. This is the reason for the theory sometimes being called interchangeably as denotational/compositional/propositional semantics. There is a commonly shared feature in the theories of Frege and Mill: singularity. In both theories, the referent always is singular no matter whether it is a world entity (Mill's referential theory) or it is a truth-value of a proposition (Frege's sense and reference). Bertrand Russell's theory of descriptions challenged the singularity notion of the Frege's theory (Malpas, 2012; Riemer, 2010). Russell's main objection to the Frege's theory focuses on definite descriptions (Riemer, 2010). Russell argued that the woman in the sentence "The woman who lives there is not a biochemist" presupposes more than it denotes. The referent (woman) is singular but because of the definite noun phrase, there are three propositions as below (see Appendix-A.3 for more details).

1. At least one woman lives there.
2. At most one woman lives there.
3. Whoever lives there is a biochemist.

The sense of singular referent is not singular in a definite noun phrase, there are three senses (Lycan, 2008). Russell focused on singularity of referent in the sentence but Strawson (1950) had a direct objection to the sentence itself. Strawson argued that Russell's theory treats sentences and their properties as disembodied and ignores the

context in which they are used (Lycan, 2008). He pointed out that although there is a presupposition in the example above, we do not know whether the woman exists. How can we be sure that a woman must exist? It depends to the context. The presupposition of the sentence may not even exist. This leads us to the conclusion that it cannot be used to make a proper statement, so it has no truth value and he argued that expressions do not refer, but people do (Lycan, 2008; Strawson, 1950).

Following Strawson's critiques, it can be seen that propositional semantics cannot denote the entire meaning of the referent which is highly context-bounded in the world. In 1953, Wittgenstein gave an explanation for the relation between the meaning of a word and its context-bounded notion. Wittgenstein argued that the word in a contextual world may have more meaning which cannot be totally denoted by propositional semantics. Accordingly, the words gain their full meaning when we use them. This was later called Wittgenstein's use theory of meaning.

Wittgenstein and J.L Austin[1] argued that examining a proposition expressed by a sentence and treating it as an object of interest is not an appropriate method of investigation (Lycan, 2008; Austin, 1979). They considered that language and linguistic entities are not dull abstracts that can be examined like specimens under a microscope, on the contrary, language takes the form of behavior and social activity (Lycan, 2008). The main arguments of Wittgenstein and J.L Austin are that "Propositions expressed by sentence are fairly violent abstractions from the uttering performed by human beings in real-world contexts on particular occasions" (Lycan, 2008, p.90).

Wittgenstein's use theory points to an analogy of linguistic activity with playing games[2] (Lycan, 2008). Linguistic activity is regulated by rules in a similar way to playing a game shaped by rules. While playing a game, there is a conventional rule which are not expressed explicitly but everybody knows them. For instance, if a player says "to me" during a football game, the other players know that he is asking them to pass the ball to him. Wittgenstein offered an pseudo-scenario (1953:2) to support his idea: conversation happening between a builder and his assistant while engaged in a building project. When builder says "slab", the assistant brings the appropriate stone. Although the word slab is not fully expressive for an outsider, their engagement in non-linguistic activities helps them to share the full meaning of *slab* within their contextual world. Thus, Wittgenstein's use theory indicates that the word in use is highly conventional rather than expressing relation to abstract entities (Lycan, 2008).

Wittgenstein's use theory was challenged by aspects of non-conventional expression. The builder-assistant scenario indicates that there is non-linguistic knowledge which is conventional and contextual in the world. However, what about a genuine sentence which has never been presented before? We know that the count of words in a vocabulary is limited, but any language may produce infinite sentences of which one may never have been heard before. Moreover, humans are grammatically competent

[1] J.L Austin focused on the performative utterance of a declarative sentence. Conventional social acts which have no state or description, for example "I apologize" or (in a game of bridge) "I double". These kinds of acts are called "speech acts"(Lycan, 2008).

[2] Lycan (2008) mentioned how, first, Wittgenstein noticed the relation between words used in a game and their meanings. Lycan refers to Freeman Dyson, a Cambridge undergraduate,who reported that it was while Wittgenstein was walking through a field where a football match was in progress, that he first noticed that we play games with words (Lycan, 2008).

to understand any sentence expressed in their language. This leads us to conclude that humans comprehend the meaning of a sentence through the help of known words and grammatically governed rules, even though the sentence has never been heard before. However, Wittgenstein considered a different understanding of meaning which formal semanticians had never previously addressed before. There was a meaning outside which is not directly indicated by the grammatically-governed-words. How is it possible to know the meaning? Firth (1957) expressed his intuition to solve this problem: "You shall know a word by the company it keeps" (Firth 1957, p.11).

Although Wittgenstein and Firth did not propose a practical method to validate their intuition, Deerwester et al. (1990) did and thesis proposes a practical method which is a continuation of the work of Deerwester et al. (1990).

1.2 Situating the thesis in the literature

Wittgenstein and Firth lead to the conclusion that the words in a sentence may be understood through the means of co-occurring words. However, this does not explain the complete meaning of a word. The same issue was also pointed out by Strawson's critiques of compositional semantics which builds a meaning of a sentence from its sub-parts. However, the studies of Wittgenstein and Firth help in determining the meaning of a word by its co-occurring words. It is conceivable that a text is a qualitative verbal material (Graesser et al., 2004) and since the meaning cannot be grounded in a set of symbols, decomposing meaning is not possible. Therefore the best way to evaluate a meaning is using another addressed meaning. When Wittgenstein and Firth pointed out the contextual information of the sentences, they did not provide any mathematical modeling as in propositional semantics. In 1971, Salton developed a mathematical modeling which can produce the semantic similarities of words in a document by using the frequency information of the words adjacent in sentences. At first glance, it may not seem possible to derive the frequency-based assumption from the theories of Wittgenstein and Firth but another implication of Firth's assumption (1957) was that the co-occurring words give the frequency information of being co-occurred in a sentence. Distributional Semantics (Frequency based semantics) was developed on this assumption. The co-occurring words in a sentence tend to have similar meaning when the frequency of being that co-occurred increased. This thesis is based on this assumption. Moreover, since co-occurring words hold latent semantic information for each other, adjacent sentences also hold coherence information for themselves. The following chapter presents a brief summary of Distributional Semantics and where this thesis is located in the field of Distributional Semantics.

CHAPTER 2

DISTRIBUTIONAL SEMANTICS

2.1 Introduction

Distributional Semantics (DS) is a research area of linguistics based on the *Distributional Hypothesis* (DH). DH is a semantic theory which states that co-occurring words in the same context tend to have similar meanings (Harris, 1954). It has its theoretical roots in various traditions, including American structuralist linguistics, British lexicology and certain schools of psychology and philosophy (Firth, 1957; Harris, 1954; Miller and Charles, 1991; Wittgenstein, 1953). DS states that the degree of semantic similarity between two linguistics units (words, noun phrases, paragraph) can be modeled as a function of the degree of overlap among their linguistic contexts (Baroni and Lenci, 2010). The overlap between linguistic contexts is determined by the co-occurrence of the same words. According to the distributional hypothesis, observing more frequency values of the co-occurring words (or linguistic units) means that the targeted linguistic units are more 'similar'. Since similarity is obtained through the frequency of overlap, DS is also called as *Frequency Based Semantics*. On the other hand, the technique is referred to the technique of *Bag of Words*. It has certain fixed steps listed as below (Lund et al., 1995; Landauer and Dumais, 1997; Turney and Pantel, 2010) .

1. Building term-document matrix where the row-vector corresponds to terms and the column-vector corresponds to documents.
2. Defining each frequency of co-occurring words as term-vector's element.
3. Defining the set of frequency of co-occurring words in a sentence as document-vector.
4. Defining the similarity of term-vectors as their cosine values (Euclidean distance).
5. Defining the similarity of document-vectors as their cosine values (Euclidean distance).

Since the technique recognizes the result of euclidean distance calculation as the degree of similarity, it is also called the Vector Space Models (VSM). VSMs have mostly been popular among computational linguists and cognitive scientists, and used for semantic representation of words and documents (Grefenstette 1994; Lund and Burgess 1996; Landauer and Dumais 1997; Sahlgren 2006; Bullinaria and Levy 2007; Griffiths, Steyvers, and Tenenbaum 2007; Pado and Lapata 2007; Lenci 2008; Turney and Pantel). After the specific type of VSM called Latent Semantic Analysis (LSA) was introduced by Deerwester et al. (1990), the research area of distributional semantics has been mainly grounded on LSA. Before giving the details of LSA in Chapter 4, this chapter gives a brief summary of techniques used in the research area of distributional semantics and situate the thesis work in the literature.

2.2 Techniques used in the field of distributional semantics

The first technique was based on VSM, and used for *Information Retrieving Systems* (SMART) developed by Salton and colleagues (Salton, 1971; Salton, Wong, and Yang, 1975). In SMART, documents are marked as points on a vector space, and the similarity is measured by the Euclidean distance (Turney and Pantel, 2010). Salton, Wong, and Yang (1975) addressed the search capability of modeling a document as a matrix of term-document pairs which had great success in information retrieval systems (Turney and Pantel, 2010). The success of the VSM went beyond information retrieving system, and has been applied to some semantic tasks in natural language processing (Turney and Pantel, 2010). As an example, Rapp (2003) developed a vector space of the meaning of words which scored 92.5% on multiple choice synonym questions from the Test of English as a Foreign Language (TOEFL) whereas average human score was 64.5%. Turney (2006) developed a vector space based semantic space, which achieved a score of 56% on multiple choice analogy questions from the SAT college entrance test in contrast to the average human score of 57%.

Scholars used different approaches to build word and document vectors in a text such as; windows of words (Lund and Burgess, 1996) and grammatical dependencies (Lund and Burgess, 1996; Lin, 1998; Padó and Lapata, 2007). Lund and Burgess (1996) developed a vector space with the help of a moving window. Their method was to span a sized-window across the corpus, and accept the words within the window as co-occurring words. This method is also known as the Hyperspace Analogue to Language (HAL) (Lund, Burgess, and Atchley, 1995; Lund and Burgess, 1996). Lin (1998) created a vector space with the help of the grammatical dependencies. They built a dependency triple consisting of a head, dependency type and a modifier. The frequency of co-occurring words were obtained from the corpus of triple, and similarity of words are retrieved by a similarity function. The parameters of the similarity function are the elements of the dependency triple.

Although all methodologies created a vector space based on word co-occurrence, they do not necessarily use the same type of word co-occurrence. Some focus on word co-occurrence in an orthodox understanding but others use syntactical dependencies. Regardless of the surface data of the language is used, the method of constructing a vector space by event frequencies does not change. The methods used in VSM based modeling have been changed in 1990 by Deerwester et al. (1990). Deerwester and

his colleagues introduced a new technique which is called Latent semantic Analysis (LSA) using the same methodology but a different mathematical foundation (Singular Value Decomposition, SVD) in order to detect the word similarity. The significance of LSA was that it allows for mapping words in n^{th} dimensional space into a reduced dimensional space, and this assists in revealing hidden similarity relations between the words. It can reveal hypernymy or synonymy of words co-occurring in the same context. This capability gives conceivable reasons for cognitive scientist to believe that VSMs (LSA and HAL) might be used to model human cognitive capabilities (Landauer et al., 2013).

Although HAL and LSA use term-vector and event frequencies, they can be distinguished in two areas: the data used to construct word frequencies and mathematical foundation. HAL (Lund & Burgess, 1996) and LSA (Deerwester et al., 1990) use term-document matrices as VSMs, but they have different co-occurrence matrix representations. HAL uses the term-term matrix where a term is a word in a sized sliding window for each piece of the document. The sized sliding window spans across the word corpus. This sliding window is used to determine which words are in the neighboring in co-occurred sentence (the frequency of neighboring in the sliding window are entered into the matrix). However, LSA uses a term-document matrix where a term is a word that occurred in a sentence. LSA has no sliding window algorithm, rather it uses on SVD to decompose the term-document vectors to its orthogonal vectors (eigen-vectors), and captures the *latent* semantic similarities while reconstructing the term-document matrix from its orthogonal vectors. HAL does not use orthogonalization processes, it assumes that term row in the term-term matrix is the vector that defines the term in the document. The similarity function of HAL is the function of Euclidean distance as in LSA.

The content of this thesis is highly related to the term-document matrix and the technique of LSA. However, it differs in the method used to construct the term-document matrix. The current work does not use term-term matrix as in HAL or term-document matrix as in LSA rather, it uses document-distance matrix (derived from term-document matrix) and applies two genuine algorithms which span the document-distance matrix. Details will be given in Chapter 5.

In the next section, a brief summary will be given of Deerwester's work and extensions in the world of distributional semantics to situate the thesis in Distributional Semantics.

2.2.1 Deerwester's research (1990)

The importance of Deerwester's work using LSA technique with SVD is that it shows how a word can be defined by the help of co-occurring words in a text. It is the first practical algorithm that validates the intuition of Wittgenstein (1953), Harris (1954) and Firth (Turney and Pantel, 2010).

Deerwester et al. (1990) analyzed the term-document matrix given in Figure 2.1. The term-document matrix consists of 9 columns and 10 rows. Columns represent documents (sentences), and rows represents terms (words). Each element of the matrix represents the frequency of co-occurring word in the sentence. The sentences and

Technical Memo Example

Titles

c1:	*Human* machine *interface* for Lab ABC *computer* applications
c2:	A *survey of user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user*-perceived *response time* to error measurement
m1:	The generation of random, binary unordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors*: A *survey*

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Figure 2.1: Deerwester's data (1990): A sample data set consisting of the titles of 9 technical memoranda. Terms occurring in more than one title are italicized. There are two classes of documents; five concerning human-computer interaction (c1-c5) and four about graphs (m1-m4).

words in the matrix can be defined as vectors. For example, the column of c2 stores the elements of $\vec{c}_2$ and the row of *human* stores the elements of $\vec{v}_{human}$ is given as below.

$\vec{c}_2 = (0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0)$
$\vec{v}_{human} = (1, 0, 0, 1, 0, 0, 0, 0, 0, 0)$

Examining term-document matrix in Figure 2.1, gives the following assumptions.

1. *Human*, *interface* and *computer* co-occurred in c1
2. *Computer*, *user*, *system*, *response* and *time* co-occurred in c2
3. *System* co-occurred in c2, c3, and c4
4. Although *human* and *user* do not co-occur in the same sentence, they both co-occur with *computer*.
5. The term-document matrix consists of two paragraphs (c1-c5 and m1-m5) and only the word *survey* is found in both paragraphs.

Deerwester et al. (1990) showed how the words of the *human* and *user* are similar, although they do not occur in the same sentence. The reduced two dimensional versions of nine dimensional term-document matrix is presented in Figure 2.2. The vector of the *user* and *human* has cosine value of 0.818 which indicates high similarity (Deerwester et al., 1990). The details of LSA are given in Appendix D.

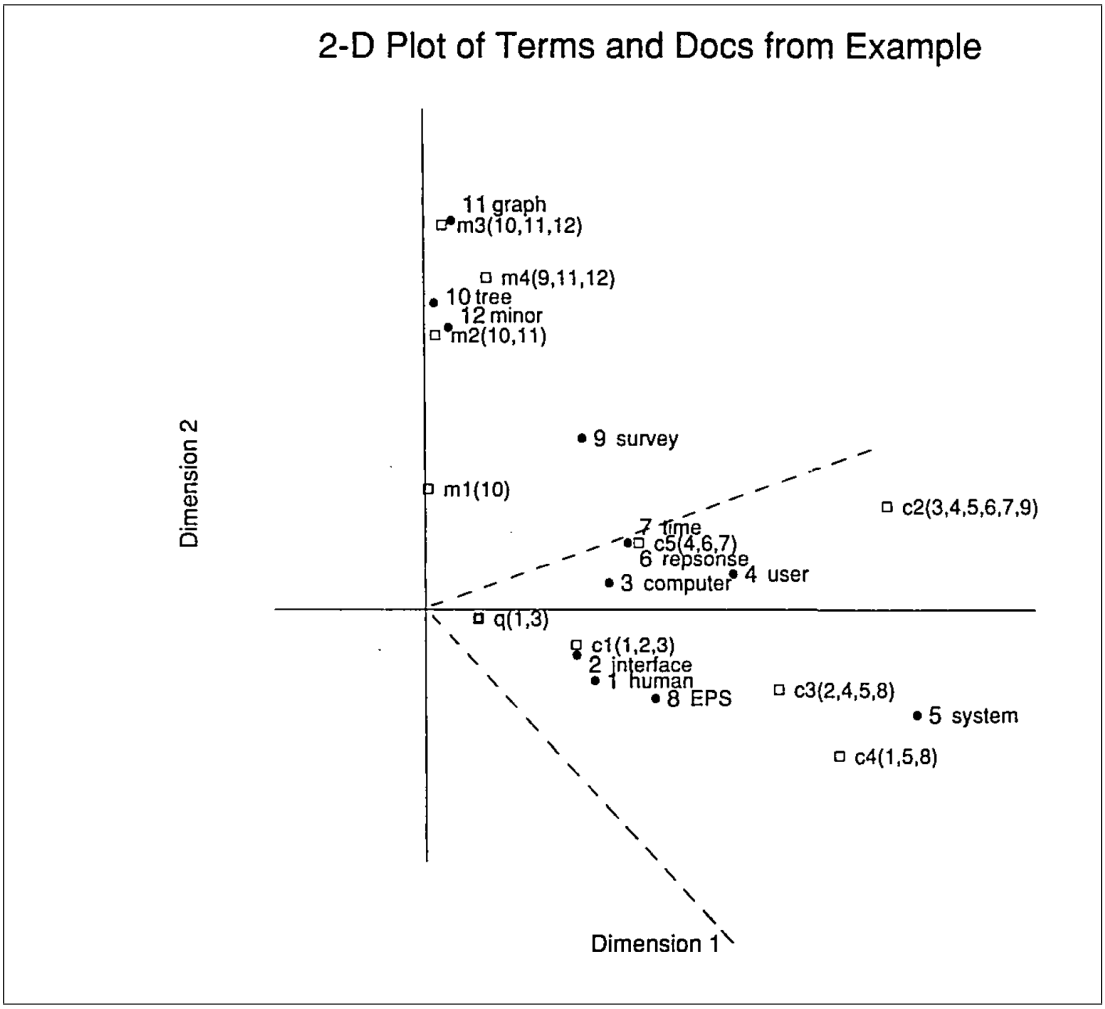


Figure 2.2: Deerwester's findings (1990): A two-dimensional plot of 12 Terms and 9 Documents from the sample TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically.

Figure-2.2 is obtained by reducing 9 dimensions to 2 dimensions for each term vector. The reduction is done with the method of singular value decomposition (see Appendix-D). In Figure-2.2, $\cos(\theta)$ values of vectors reveal the similarity of the points plotted in 2D. Cosine value of 0 or less indicates dissimilarity while positive cosine value indicates relative similarity. According to this formula, human-interface-system triple is so close to each other which means that they may have similar meanings in a contextual environment. This is what expected because human-interface, and human-system pairs are co-occurred in the same sentences.

In this study, the importance of Deerwester's work is that Deerwester's data and results are used as a baseline for the comparison of results of algorithms that are introduced in the thesis.

2.3 Related studies in distributional semantics (DS)

DS is not limited to the frequency of data. According to Erk (2013), there are four distinct approaches which extend DS as follows.

1. A single vector space representation for a phrase or sentence is computed from the representations of the individual words (Mitchell and Lapata, 2010)
2. Two phrases or sentences are compared by combining multiple pairwise similarity values (Turney, 2012)
3. Weighted inference rules integrate distributional similarity and formal logic (Garrette, Erk, and Mooney, 2011)
4. A single space integrates formal logic and vectors (Clarke, 2012)

Each approach listed above goes beyond the orthodox understanding of distributional semantics. Distributional semantics is commonly defined as frequency based semantics. Frequency events at the surface structure of the language are the object of interest in this domain. However, the approaches categorized by Erk (2013) take this understanding further. By giving a brief summary of these approaches, the aim is to obtain a complete survey of distributional semantics. Thus, it is expected to locate this thesis in the research area of distributional semantics.

2.3.1 A single vector space representation for a phrase or sentence

The work of Mitchell and Lapata (2010) is based on applying a composition function to sentence units in which the parameters of the function are fed from the units residing in the Distributional Semantic space. They claim that a phrase can have vector representation of its sub-parts. For example, the noun phrase of *practical difficulty* can be represented as a vector. Their study differs from Deerwester's work (1990) because they use information from syntactic dependency in order to define the vector of a phrase or a sentence. Mitchell and Lapata claim that compositional semantics cannot measure semantic similarity since it is based on discrete symbols (Mitchell and Lapata, 2010). Moreover, they criticize distributional semantics for discarding the word order, and syntactic relations. In order to emphasize the importance of word order and syntactic relations, they give the example below in which two sentences have exactly same words but different meanings (Mitchell and Lapata, 2010).

1. It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem.
2. That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.

The framework introduced by Mitchell and Lapata (2010) defines the relation of p between constituents of u and v as below.

$p = f(u, v)$

u and v may be word or phrase. p is a production of u and v but it resides in the semantic space of u and v. u and v may have a syntactic relation as below where R denotes the syntactic relation.

$p = f(u, v, R)$

The constituents of u and v may have syntactic relation, and may exist only in world knowledge as below. K denotes world knowledge.

$p = f(u, v, R, K)$

There are two operations proposed by the framework devised by Mitchell and Lapata (2010): addition and multiplication. The latter is not a matrix multiplication or inner product of vectors but tensor product ($\oplus$). The tensor product is defined as below.

$$\oplus(A, B) = A^T B \tag{2.1}$$

The simplest composition function is the addition of vectors: $p = u + v$. With p being defined as a cartesian product of u and v with additive additive composition function: $p = Au + Bv$. The framework also defines a composition function with the tensor product: $p = Cuv$, C denotes a rank-3 matrix. For example, the result of an additive composition function can be obtained as follows.

$u = (0, 6, 2, 10, 4)$
$v = (1, 8, 4, 4, 0)$
$p = u + v = (0, 6, 2, 10, 4) + (1, 8, 4, 4, 0) = (1, 14, 6, 14, 4)$

For the phrase *practical difficulty*, the frequency values of constituents are defined in Figure 2.3.

	music	solution	economy	craft	reasonable
practical	0	6	2	10	4
difficulty	1	8	4	4	0

Figure 2.3: A hypothetical semantic space for *practical* and *difficulty*(Mitchell and Lapata, 2010).

The tensor compositional function for the phrase of *practical difficulty* is defined as:

$\vec{u}_{practical} = (0, 6, 2, 10, 4)$
$\vec{v}_{difficulty} = (1, 8, 4, 4, 0)$

$$\vec{u}_{practical} \oplus \vec{u}_{difficulty} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 6 & 48 & 24 & 24 & 0 \\ 2 & 16 & 8 & 8 & 0 \\ 10 & 80 & 40 & 40 & 0 \\ 4 & 32 & 16 & 16 & 0 \end{bmatrix} \tag{2.2}$$

To sum up, Mitchell and Lapata (2010) extend event frequencies from term level to phrase level, and furthermore, they introduce a formalism of composition as a function of vectors. This formalization causes the result of a composition function to be an event frequency of the vector of words and phrases. In this thesis, the document-distance matrix is presented as an event frequency of the surface structure of a text. Since Mitchell and Lapata (2010) defined compositional function on the basis of vector space. This formalization can also be applied to the vector spaced defined by document-distance matrix.

2.3.2 Pairwise similarity values (Turney, 2013)

Turney (2013) proposes a framework of integrated compositional-distributional semantics by referring to an analogy between the relation of mason-stone and carpenter-wood . Moreover, syntactic dependencies of aforementioned pairs also have an analogy. Turney argues that the relational semantics between the tuples of (mason:stone) and (carpenter:wood) can be captured by building two different semantic spaces: one for syntactic, another for lexical (contextual bag-of-words) semantic, namely the *Dual space model of Semantic Relations and Compositions*. The term *dual space* refers to domain, and function similarity. Carpenter and wood are in the context of carpentry. Mason and stone are in the context of masonry. The domain similarity refers to one aspect of dual space. Similarly, the mason-carpenter pair shares the functionality of artisans and the stone-wood pair shares the functionality of materials. The dual space model builds two different semantic spaces for domain and function spaces and merges these semantic spaces as;

$$sim(a, b) = geo(sim_d(a, b), sim_f(a, b)) \tag{2.3}$$

Where sim_d denotes the similarity function in the lexical vector space and sim_f denotes the similarity function in the function vector space. The geo function denotes the geometric mean because when the similarity of the result of both similarity (sim_d,sim_f) is high, the combined similarity (sim) must be much higher(Turney, 2013). The definition of geo is given below (Turney, 2013).

$$geo(x_1, x_2, x_3..., x_n) = \begin{cases} (x_1, x_2, x_n)^{1/n}, & \text{if } x_i > 0 \; for \; all \; i = 1, ..., n \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

Both approaches presented by Turney (2013) and Mitchell and Lapata (2010) try to merge the information provided by compositional semantics and lexical semantics (bag of words). However, they differ in mathematical foundation. Mitchell and Lapata (2010) introduced a bottom up approach which can be applied to any constituent such as phrases and sentences, whereas Turney (2013) uses compositional and lexical information when calculating the degree of similarity. This thesis refers to one aspect of dual space model that is, lexical semantics. The document-distance matrix can be used as a local weighting function of the lexicals while constructing a lexical semantic space for the dual space model.

2.3.3 Weighted inference rules integrate distributional similarity and formal logic (Garrette, Erk, and Mooney, 2011)

Garrette, Erk, and Mooney (2011) integrate first-order logic, probabilistic knowledge, and distributional word meaning to achieve inferences in a text. They criticizes high dimensional semantic space because of its incapability of providing a meaning for a complete sentence. To find a solution to this incapability, the authors developed an approach consisting of the steps listed below. This approach can be used to produce the possible paraphrasing of a sentence.

1. Parse text by Boxer (an engine which produces Discourse Representation Structures, DRS) (Bos et al., 2004)
2. Use result of Boxer as a list of first order logical forms.
3. Connect logical forms f_1, f_2 by injecting new logical rules between them, if f_1 and f_2 share re-occurring words.
4. Produce possible paraphrasing of predicates of logical forms, by looking at distributional semantic similarities of predicates.
5. Rank all possible paraphrases according to Zipfian distribution.
6. Define a probability as P_k $1/k$ for logical form where k denotes the zipfian rank of the logical form.
7. Place probabilities as input to Markov Logic Network (MLN) to produce inferences.

Garrette et al. (2011) showed how first-order logic can be integrated with probabilistic knowledge for word meaning. The model they introduced is a part of Statistical Relation AI. Their study allows for the full expressivity of first-order logic, and the ability to reason with probabilities and use high-dimensional semantics space with logic-based representations (Erk and Padó, 2008; Thater, Fürstenau, and Pinkal, 2010; Erk and Padó, 2010; Hobbs et al., 1988).

The relation between current thesis and the work of Garrette, Erk, and Mooney (2011) is the use of distributional semantic similarities for pairs of linguistics constituents. Although those authors focused on the correspondence of re-occurring words between

logical forms, the current thesis does not propose an approach to calculate word similarities, it will be shown that distance between re-occurring words can be an indicator of word similarities.

2.3.4 A single space integrates formal logic and vectors

Clarke's study 2012 made a definite distinction from three extensions of distributional semantics mentioned in previous sections. He states that studies in the theory of meaning have revolved around logical and ontological representations. Clarke proposes a mathematical formalism called "meaning as context", and introduces a new set of definitions based on the vector space modeling of meaning. The main distinction is that the definitions are mathematical and algebraic. Listing the definitions are beyond the scope of the thesis. However, it is valuable to give an example of one of them, namely, the *Partially Ordered Vector Space* given below.

> **Definition 2.3.1** *(Partially Ordered Vector Space) A partially ordered vector space V is a real vector space together with a partial ordering $\leq$ such that:*
>
> *if $x \leq y$ then $x + z < y + z$*
> *if $x \leq y$ then $\alpha x < \alpha y$*
>
> *for all $x, y, z \in V$, and for all $\alpha > 0$. Such a partial ordering is called a vector space order on V. An element u of V satisfying $u \geq 0$ is called a **positive element**; the set of all positive elements of V is denoted V^+. if $\leq$ defines a lattice on V then the space is called a **vector lattice** or **Riesz space**. (Clarke, 2012, p. 49)* ■

The words of *orange* and *fruit* occur in six documents with different frequencies. The result of vector lattice operation of $orange \wedge fruit$ is listed in Figure 2.4.

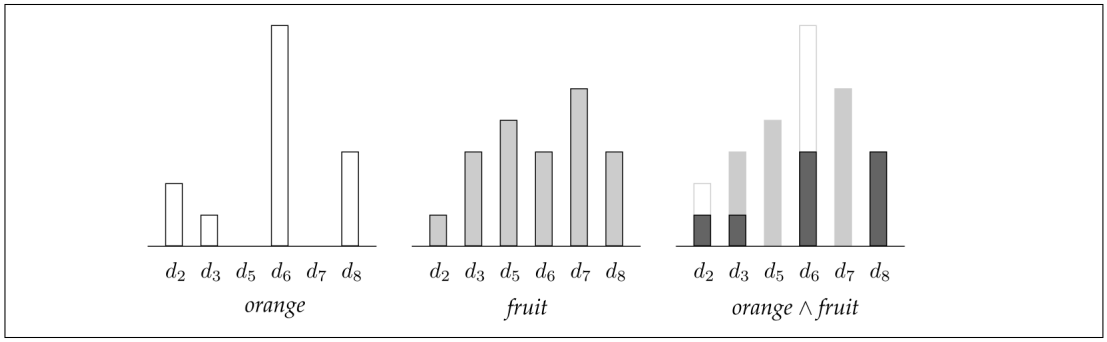


Figure 2.4: The result of vector lattice operation of the *orange* and *fruit* occurred in six documents (Clarke, 2012).

This study defined a new algebraic theory of meaning called "meaning as context", and argues that this may merge the distributional semantic and compositional semantics in the same formalism. The main idea of the theory of *context as meaning* is that

the meaning can be determined by the context with the help of the statistical properties of the language. Clarke (2012) states the purpose of the study is to use the techniques of distributional semantics in a principled manner.

The thesis and this study are both situated in the domain of distributional semantics. The thesis uses the technique of *Bag of Words* in an orthodox way, and only focused on the frequency event of the surface structure of language. Compared to Clarke's study, the current thesis does not propose a general formalism but a practical method which reveals the distributional property of the distance of re-occurred words as a cohesive cue to measure textual coherence.

2.3.5 Summary

The work reported in this thesis is situated with its philosophical background as given in Section 1.2 and it is located in the research area of DS. In this section, a brief summary of approaches of DS, and its extensions were presented. DS has its roots in the intuitions of Wittgenstein (1953), Harris (1954) and Firth (1957): you know a word with its co-occurring word, and you know the meaning when you use it. This intuition gained its initial practical result in information retrieving system (Salton, 1971) and then Deerwester et al. (1990) proved Wittgenstein's intuition in a mathematically rigorous way. The studies of Landauer et al. (2013) , Landauer and Dumais (1997) and Foltz, Kintsch, and Landauer (1998) developed new approaches based on Deerwester's study. The common ground of these studies is that they used same methodologies but there is an induced DS within different problem domains. Next, the four extensions that are listed in this section propose new approaches to overcome the insufficiencies of DS. The common feature of these extensions is that they all introduce new methodologies to integrate DS with compositional semantics. The aim of these studies is to make DS capable of quantifying the meaning of phrases and sentences. Without these extensions, DS will remain as a theoretical framework which only quantifies word level meaning, and yet cannot go beyond that.

The thesis occupies a particular place among the aforementioned studies. It does not provide any extension to DS and use methodologies with an orthodox understanding rather, the current work is based on term frequencies of the surface structure of text and it employs the distance between re-occurring words as an object of interest. The studies of Deerwester et al. (1990), Landauer et al. (2013) , Landauer and Dumais (1997) and Foltz, Kintsch, and Landauer (1998) mainly focused on the frequency event of words, but the current work utilizes a new frequency event in text; distance between re-occurring words. Therefore, the work in this thesis is an induction of DS with a new frequency event which does not bring a new extension to DS. Next chapter contains a brief summary of Latent Semantic Analysis (LSA) a practical method in DS and gives the position of this thesis in terms of LSA studies.

CHAPTER 3

TEXTUAL COHERENCE, COHESION AND DISTANCE

3.1 Introduction

Comprehension is a daily, regular activity of mind that happens every time meaning is extracted from a wide sort of media such as: conversations, pictures, videos, and texts (McNamara and Magliano, 2009). Although individuals engage in comprehension in most of the time, however, due to the ease of control, manipulation and analysis, scholars mostly focus on comprehension while reading a text (McNamara and Magliano, 2009). There are seven models of comprehension: Resonance model, Landscape model, Langston and Trabasso model , Construction-Integration model, Predication model, Sentence Gestalt model and Story Gestalt model. All these generally accept the principle which says "comprehension is affected by the coherence of reader's situational model" (McNamara and Magliano 2009, p.313). Surface representation of text and the reader's inferences while reading the text aids the reader in the production of a situational model (McNamara and Magliano, 2009). Zwaan and Radvansky (1998) described situation models as integrated mental representations of a described state of affairs. Surface representation and textbase are strictly dependent on observable cohesive cues whereas the situation model is unobservable.

The effect of cohesive cues on the production of situational model of the reader is generally recognized by the comprehension models given above. Cohesive cues were categorized in detail by Halliday and Hasan (1976). Figure-3.1 lists the upper hierarchical categories which include sub categories indicating the linguistic cohesive lexical items.

This thesis does not consider conjunction, reference and substitution rather it focuses on lexical cohesion. More specifically, it focuses on distance of re-occurring lexical cohesion of adjacent sentences. It locates the distributional frequency event of the distance of re-occurring lexicals on a distributional semantic space and investigates its hidden semantic information about coherence.

Although distance does not have a place in the categorization listed in Figure-3.1, it is defined as "Direction and distance of cohesion" (Halliday and Hasan 1976, p.339). According to authors, any distance based on tie[1] contains these characteristics given

[1] a single instance of cohesion, a term for one occurrence of a pair of cohesively related items" (Halliday and Hasan 1976, p.3). "A tie is best interpreted as a relation between these two elements" (Halliday and Hasan 1976, p.329). "A tie may be reference, substitution, ellipsis, conjunction, and lexical cohesion" (Halliday and Hasan

Representation in linguistic system / *Type of cohesive relation*	Semantic	Lexicogrammatical (typically)
Conjunction	Additive, adversative, causal and temporal relations; external and internal	Discourse adjuncts: adverbial groups, prepositional groups
Reference	Identification:	
	by speech role	Personals
	by proximity	Demonstratives
	by specificity (only)	Definite article
	Reference point	Comparatives
Lexical cohesion	Collocation (similarity of lexical environment)	Same or associated lexical item
	Reiteration (identity of lexical reference)	Same lexical item; synonym; superordinate; general word
Substitution	Identity of potential reference (class meaning) in context of non-identity of actual (instantial) reference	Verbal, nominal or clausal substitute Verbal, nominal or clausal ellipsis

Figure 3.1: Cohesion categories (Halliday and Hasan 1976, p.324).

below (Halliday and Hasan, 1976).

1. Immediate (presupposing an item in a contiguous sentence) or not immediate.
2. Mediated (meanly, having one or more intervening sentences that enter into a chain of presupposition.)
3. Remote (having one or more intervening sentences not involved in the presupposition),
4. Mediated and Remote at the same time
5. Anaphoric or Cataphoric

Halliday and Hasan (1976) consider the distance of cohesion as a direction of cohesion, presupposition or syntactic (Anaphoric-cataphoric) whereas this thesis focuses on the spatial distance between re-occurring words of adjacent sentences to quantify coherence. Since coherence affects comprehension of the reader, evaluating comprehension models according to the spatial distance of re-occurring words will help to position the distance notion of re-occurring words within the coherence phenomenon. This chapter aims to present the correlation between spatial distance of re-occurring words and coherence. In order to achieve this goal, the seven models given above were investigated to find an aspect of spatial distance within their modeling.

1976, p.4).

Before, presenting the results of investigation of models' assessment, before presenting the result, it is better to define the notion of distance in a more abstract way.

"The real number system is ordered by the relation <" (Trench 2003, p.2). The ordered n-tuples of real numbers are also defined as *Euclidean space* (Trench, 2003). Euclidean distance is defined on *Euclidean space* (Trench, 2003). Therefore, having an ordered set of numbers brings definition of Euclidean distance. Any coherent text is a set of ordered sentences. Since sentences can be considered as a set of ordered tuples, the definitions of *Euclidean space* and *Euclidean distance* for ordered sentences can appear there. Therefore, while investigating on seven models of comprehension, finding the order of tuples out will lead us to conclude that the distance notion of re-occurring words are intrinsic for these models.

This thesis investigates the distance notion of the seven models, rather than giving all the details which is beyond the sphere of this study. However, for a detailed description of models, the reader is referred to the study of McNamara and Magliano (2009).

3.1.1 Finding the distance notion in seven models of comprehension

The Construction-Integration model assumes that there are 3 levels of mental representation as shown in 3.1.1.

1. Surface representation (Literal wording of the text)
2. Textbase (Meaning of text expressed as propositional units) (Kintsch, 1988; Kintsch, 1998)
3. Situation model (derived from the organization of the textbase into facts that are matched to knowledge frames stored in long-term memory) (Kintsch and Van Dijk, 1978)

It has two phases; construction and integration. The aim of construction phase is to produce a network of proposition derived from sentences in the text. An example of a network of propositions is given in Figure 3.2.

Table 3.1: Example propositions for Construction-Integration Model (Lennart 2004, p.49)

LABELS	PROPOSITIONS
T1	DISCUSS(LAWYER, JUDGE, CASE)
T2	SAY(LAWYER, T3)
T3	SEND(LAWYER, DEFENDANT, PRISON)
T4	SAY(JUDGE, T5)
T3	SEND(JUDGE, DEFENDANT, PRISON)

In Figure 3.2, the links between nodes indicate the association of concepts. Each link has a weight value. The weight value of a link is determined by the researcher (or

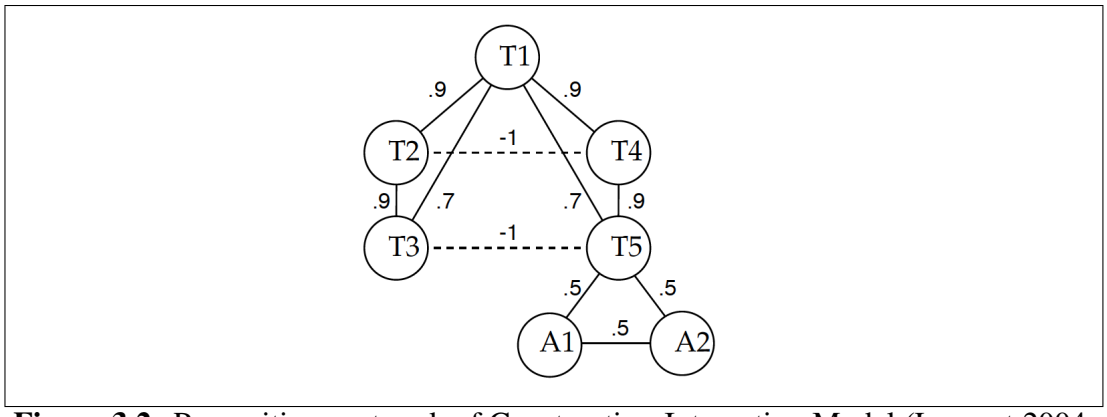


Figure 3.2: Propositions network of Construction-Integration Model (Lennart 2004, p.51)

modeler). Lennart (2004) states that weighting a link is a subjective task. For example; Schmalhofer, McDaniel, and Keefe (2002) showed that two different subjective decisions that are made on the proposition network of the Construction-Integration model yielded two different inferences (cited in Lennart, 2004). The integration phase of the model is based on the elimination of low weighted links. According to the brief summary of the Construction-Integration model, there is no definition of an ordered set and distance.

The Resonance model focuses on the activation of textual information which does not take place in working memory at the time of reading the focal sentence (McNamara and Magliano, 2009). The model tries to explain how information held in distant sentence is reactivated on focal sentence (McNamara and Magliano, 2009). This assumes that all sentences are ordered, and the following sentence (distant sentence) has distance relation with the previous sentence.

The Landscape model assigns activation values to the concepts which are activated while reading (McNamara and Magliano, 2009). There are three types of activation: 1) mentioning, 2) inferring and 3) mentioned or inferred in sentence t but not in sentence $t + 1$. Each type of activation has a different degree of activation. All three types of activation assume that sentences are ordered, and the activation of concept is performed successively. A focal sentence might contain activation of same concept in a distant sentence if the intervening sentences carry out the activations. This also indicates that there is a distance notion of the concepts which is carried in successive sentences.

The Langston and Trabasso model focuses on causality between the statements expressed through sentences. It states that two sentences (p, q) are causally connected if they pass the counterfactual test. “If q would not have occurred without p (all other things being equal), and there is no intervening event caused by p and causing q, then p and q are causally connected” (Lennart 2004, p.41). The model implies that causality occurs when successive sentences p and q cause the same event. The sentences p and q are ordered and their distance is zero.

Predication model locates discourse items (concepts, propositions) and their relations (casual, associative etc.) into vectors in a vectors space (Lennart, 2004). This model

uses LSA to present a vector space. The use of LSA means that the Euclidean distance and ordered set is already defined in this model.

The Story Gestalt, and Sentence Gestalt Models focus on the distributional representation of propositions. Unlike words, propositions are not observable in the surface structure of a text. Therefore, the distributional representation of propositions cannot be produced rather, an artificial corpus of propositions is built and the distributional representation of propositions are derived from that corpus. Both of the Gestalt models are based on a artificial corpus of propositions. The Story Gestalt Model produces a representation of propositions of the complete story for the input sentences whereas the Sentence Gestalt Model produces a representation of the proposition of the input sentence. The frequency of distributional space is obtained from sentence/event pairs, and sentences are converted into vector spaces by a neural network set up. Although there is a vector space definition for the propositions there is no notion exist of the distance amount sentence/event pairs. Therefore, this leads us to the conclusion that there is no notion of ordered set and distance in either models.

While examining comprehension models for distance definition, it is undertaken for the distance definition of Euclidean space which is different from the distance definition given by Halliday and Hasan (1976). Therefore, the research reported in this thesis should be located according to its own definition of distance.

3.1.2 Conclusion

In conclusion, the definition of distance can be found in four of the models of comprehension; Resonance model, Landscape model, Langston and Trabasso model and Predication model. Despite the fact that these models do not focus on distance, their assumptions have an intrinsic definition of distance. Moreover, Halliday and Hasan (1976) define the distance of cohesion as a cohesive clue but at the time of writing no study was found in the literature which uses this definition for the quantification of coherence. The thesis takes distance as a spatial distance measurement of re-occurring words among adjacent sentences whereas Halliday and Hasan (1976) take it as distance of syntactic cohesive cue (presupposition, anaphoric and cataphoric). Therefore, the distance definition of this thesis differs from that of Halliday and Hasan (1976). This thesis does not claim that the distance can be a base for the situational model and mental representation of text. However, it does claim that it might be a parameter to quantify the coherence of the mental representation of a text.

CHAPTER 4

LATENT SEMANTIC ANALYSIS (LSA)

4.1 Introduction

This thesis uses LSA in two ways. First, as a baseline to make a comparison of thesis result. Second, one of the algorithms proposed by the thesis produces an input matrix for LSA. Moreover, the motivation behind this thesis is mainly driven by the success of LSA in research concerning distributional semantics. Therefore, it is appropriate to situate the thesis among the studies of LSA. This section provides a detailed look over LSA, and establishes a position for this thesis within the LSA studies. The details of mathematical foundation of LSA are not given but requires the reader to be familiarity with the basics of LSA and for this reason a brief introduction to LSA can be found in Appendix D in which there is also an introduction of matrix terminology, orthonormalization and singular value decomposition (SVD).

LSA is a technique to extract the meaning of a word from its adjacent words based on statistical computations applied to a large corpus of text (Landauer and Dumais, 1997; Landauer et al., 2013). It is mainly built on "Wittgenstein's Use Theory" (1953) and the "Distributional Hypothesis" theory (Harris, 1954; Firth, 1957). The leading assumption of LSA is that the meaning of a word can be determined through the other words int he same paragraph. Thus, that the meaning of a word is contextual and not defined by itself. LSA defines a vector for each word in a text so that each element of the vector has a numerical value which indicates the frequency value of co-occurring word of that particular word. This technique can be explained by the analogy of mapping (Landauer et al., 2013). The coordinates of buildings A, B, C located in a city are known and their positions can be drawn on a sphere. Although, the exact distances between the points is not known, it is possible to determine which one is the north or south, or their relative directions to each other. The same is applicable for LSA, the meaning of a word is unknown but it is known how the meaning of a word differs from others in a particular context. As a result, the LSA technique provides a practical method to implement the intuitions of Wittgenstein, Harris and Firth. The following section shows the process of LSA.

4.2 A sample of LSA

LSA basically focuses on the term-document matrix of the text listed in Figure 4.1. To prepare the term-document matrix, the text in Figure 4.2 is converted into a term-document matrix as shown in Figure 4.2. The columns store frequency data of sentences and the rows store frequency data of the content words of the text. Just the content words are selected because only these words can have definite meaning across different contexts. Since meaning of functional words *such as, of, the ,and* differs in each sentence LSA does not offer a method for these words.

Label	*Titles*
M1	*Rock* and *Roll Music* in the 1960's
M2	Different *Drum Rolls*, a Demonstration of Techniques
M3	*Drum* and Bass *Composition*
M4	A Perspective of *Rock Music* in the 90's
M5	*Music* and *Composition* of Popular Bands
B1	How to Make *Bread* and *Rolls*, a *Demonstration*
B2	*Ingredients* for Crescent *Rolls*
B3	A *Recipe* for Sour*dough Bread*
B4	A Quick *Recipe* for Pizza *Dough* using Organic *Ingredients*

Figure 4.1: The italicized words are content-words subject to be the input of LSA.

Types	*Documents*								
	M1	M2	M3	M4	M5	B1	B2	B3	B4
Bread	0	0	0	0	0	1	0	1	0
Composition	0	0	1	0	1	0	0	0	0
Demonstration	0	1	0	0	0	1	0	0	0
Dough	0	0	0	0	0	0	0	1	1
Drum	0	1	1	0	0	0	0	0	0
Ingredients	0	0	0	0	0	0	1	0	1
Music	1	0	0	1	1	0	0	0	0
Recipe	0	0	0	0	0	0	0	1	1
Rock	1	0	0	1	0	0	0	0	0
Roll	1	1	0	0	0	1	1	0	0

Figure 4.2: The 10x9 term-by-document matrix with type frequencies.

In Figure 4.2, there are 9 documents (sentences), and 10 terms (types). Each term is defined according to their repeating frequency in the documents. LSA only works correctly if the number of terms are greater than the number of documents. The sampling of this restriction is the remit of the this section.

According to the Figure 4.2, in the text, the $\vec{v}_{bread}$ has the coordinates given below in the vector definition of the word *Bread*.

$\vec{v}_{bread} = [0, 0, 0, 0, 1, 0, 1, 0]$

Before applying LSA to the term-document matrix in Figure 4.2, because LSA is actually an application of SVD, this process needs to be explained. SVD is a mathematical process which takes a Matrix M_{mn} and decompose it into three matrices such as $M_{mn} = USV^T$ where each decomposed matrix has definition as given below.

U_{mn}	Term (type) matrix of original term-document matrix M_{mn}
S_{mn}	Characteristic vectors (eigen vectors) of original matrix M_{nn}.
	The matrix S_{nn} is diagonal and symmetric.
	The elements of diagonals are eigen values of intended eigen vectors.
V^T_{mn}	Document (sentence) matrix of original term-document matrix M_{mn}

The decomposition of matrix M_{mn} is carried out by algebraic orthonormalization. An example of orthonormalization (Gramm-schmidt) is given in section D.1.3.7. The process of orthonormalization of a matrix yields a set of eigen vectors from the original matrix. Eigen vectors have eigen values which are scalars of the vector space defined by eigen vectors. These eigen vectors are characteristic vectors of the column vectors of the original matrix. Characteristic vectors hold information about the variance of scalars in the vector space (or in a dynamic systems). They hold same information as the co-variance matrix hold in a dynamic system. However, they are defined as vectors not a set of numerical values as in co-variance matrix. By using SVD, it is possible to define a vector v_i by the characteristic vectors of the vector space within a range of error values. This is what intended when defining a word by other words in text. Therefore, SVD is a method to quantify the meaning of a word by its distributional frequency. The same aim can be achieved by Principle component analysis (PCA) which also helps to find the characteristic vectors of a set of vectors but the matrix representation of vectors has to be symmetric. SVD does not have such a restriction.

Mostly, the first step in LSA is application of a weighting function to the original matrix. The weighting matrix is used to decrease the effect of the most frequent word, and increase the effect of least frequent word in the matrix. The intuition in this process is that frequent words are tend to be ambiguous and have less effect on the composed meaning of the sentence, on the contrary, less frequent words are tend to be less ambiguous and give more information about the topic of a sentence. For example, a text about music consists the word *music* repeatedly many times. The *music* will become mostly co-occurring word in the text. This implies that music can be defined by other words and other words can be defined by *music*. This is not plausible because there is no such word that can process all meanings, and yet, all words cannot share the same sense of a word. The same reasoning can be produced for all the least frequent words in a text. If a word occurs only in one paragraph in a book, and all the other words in that paragraph occur in other parts of the book, it would be plausible to conclude that the paragraph is about that particular word. Therefore, most frequent and least frequent words have to be normalized before the process of LSA. This is why local and global weighting functions are required. Local weighting functions are defined with the help of the document itself whereas global weighting functions are defined by the frequencies of a particular corpus. To give an example, there is a paragraph containing 10 sentences about psychology and words are distributed equally. To make them unequal, a global weighting function may be defined using a collection of psychology books. In order to apply LSA on the matrix represented in Figure 4.2, original matrix is multiplied with weighted matrix. The

weighted value of a_{ij} of Matrix A is calculated below.

$a_{ij} = local(a_{ij}) * global(i)$

The weighting matrix in Figure 4.3 is derived from the input matrix in Figure 4.2 with the help of a weighting function.

Types	*Documents*								
	M1	M2	M3	M4	M5	B1	B2	B3	B4
Bread	0	0	0	0	0	.474	0	.474	0
Composition	0	0	.474	0	.474	0	0	0	0
Demonstration	0	.474	0	0	0	.474	0	0	0
Dough	0	0	0	0	0	0	0	.474	.474
Drum	0	.474	.474	0	0	0	0	0	0
Ingredients	0	0	0	0	0	0	.474	0	.474
Music	.347	0	0	.347	.347	0	0	0	0
Recipe	0	0	0	0	0	0	0	.474	.474
Rock	.474	0	0	.474	0	0	0	0	0
Roll	.256	.256	0	0	0	.256	.256	0	0

Figure 4.3: The weighting Matrix Music-Baking.

LSA is applied to the matrix listed in Figure 4.3. The result of LSA on the input matrix is listed in Figure 4.4.
In Figure 4.4, three matrices given are the type vectors (U), the eigen vectors (S) and the document vectors (V^T) of the original matrix. The multiplication of the matrices in Figure 4.4 will give the original matrix A.

$A = U * S$ brings semantic space for words and $A = S * V^T$ gives semantic space for documents. Since both semantic spaces have 9 dimensions, it cannot be draw it on a 2-dimensional space. To explain this, here is a rank-2 matrix of S matrix and U matrix and the result of their multiplication.

$$S = \begin{bmatrix} 1.10 & 0 \\ 0 & 0.96 \end{bmatrix}$$

$$U = \begin{bmatrix} 1.10 & 0 \\ 0 & 0.96 \\ .04 & -.34 \\ .21 & -.44 \\ .55 & .22 \\ .10 & -.46 \\ .35 & .12 \\ .04 & -.35 \\ .55 & .22 \\ .05 & -.33 \\ .17 & -.35 \end{bmatrix}, US = \begin{bmatrix} 1.10 & 0 \\ 0 & 0.96 \\ .04 & -.34 \\ .21 & -.44 \\ .55 & .22 \\ .10 & -.46 \\ .35 & .12 \\ .04 & -.35 \\ .55 & .22 \\ .05 & -.33 \\ .17 & -.35 \end{bmatrix} \begin{bmatrix} 1.10 & 0 \\ 0 & 0.96 \end{bmatrix} = \begin{bmatrix} 4.62 & -.09 \\ .04 & -.32 \\ .23 & -.42 \\ .56 & .21 \\ .11 & -.44 \\ .38 & .11 \\ .04 & -.32 \\ .56 & .21 \\ 0.05 & -.31 \\ .19 & -.33 \end{bmatrix}$$

The vectors of (US) are represented in 2-dimensional space as shown in Figure 4.5.

In Figure 4.5, rock, composition and music are aligned near to each other whereas

Matrix U-Type Vectors									
Bread	**.42**	**–.09**	–.20	.33	–.48	–.33	.46	–.21	–.28
Composition	**.04**	**–.34**	.09	–.67	–.28	–.43	.02	–.06	.40
Demonstration	**.21**	**–.44**	–.42	.29	.09	–.02	–.60	–.29	.21
Dough	**.55**	**.22**	.10	–.11	–.12	.23	–.15	.15	.11
Drum	**.10**	**–.46**	–.29	–.41	.11	.55	.26	–.02	–.37
Ingredients	**.35**	**.12**	.13	–.17	.72	–.35	.10	–.37	–.17
Music	**.04**	**–.35**	.54	.03	–.12	–.16	–.41	.18	–.58
Recipe	**.55**	**.22**	.10	–.11	–.12	.23	–.15	.15	.11
Rock	**.05**	**–.33**	.60	.29	.02	.33	.28	–.35	.37
Roll	**.17**	**–.35**	–.05	.24	.33	–.19	.25	.73	.22
Matrix Σ-Singular Values									
	1.10	0	0	0	0	0	0	0	0
	0	**.96**	0	0	0	0	0	0	0
	0	0	.86	0	0	0	0	0	0
	0	0	0	.76	0	0	0	0	0
	0	0	0	0	.66	0	0	0	0
	0	0	0	0	0	.47	0	0	0
	0	0	0	0	0	0	.27	0	0
	0	0	0	0	0	0	0	.17	0
	0	0	0	0	0	0	0	0	.07
	0	0	0	0	0	0	0	0	0
Matrix V-Document Vectors									
M1	**.07**	**–.38**	.53	.27	.08	.12	.20	.50	.42
M2	**.17**	**–.54**	–.41	.00	.28	.43	–.34	.22	–.28
M3	**.06**	**–.40**	–.11	–.67	–.12	.12	.49	–.23	.23
M4	**.03**	**–.29**	.55	.19	–.05	.22	–.04	–.62	–.37
M5	**.03**	**–.29**	.27	–.40	–.27	–.55	–.48	.21	–.17
B1	**.31**	**–.36**	–.36	.46	–.15	–.45	.00	–.32	.31
B2	**.19**	**–.04**	.06	–.02	.65	–.45	.41	.07	–.40
B3	**.66**	**.17**	.00	.06	–.51	.12	.27	.25	–.35
B4	**.63**	**.27**	.18	–.24	.35	.10	–.35	–.20	.37

Figure 4.4: LSA Result of Music-Baking Matrix. $A = U * S * V^T$

dough, recipe and ingredients are far from rock-composition-music but relatively near to each other.

$$U_{rank-1} = \begin{bmatrix} Bread \\ Composition \\ Demonstration \\ Dough \\ Drum \\ Ingredients \\ Music \\ Recipe \\ Rock \\ Roll \end{bmatrix} = \begin{bmatrix} .42 \\ .04 \\ .21 \\ .55 \\ .10 \\ .35 \\ .04 \\ .55 \\ .05 \\ .17 \end{bmatrix}$$

The same can be observed in the Rank-1 matrix of the original data as below. In Rank-1 matrix, the values of rock, composition and music are .05, .04 and .04 respectively whereas the values of dough, recipe and ingredients are .55, .55, .35, respectively. The same approximated values can be seen in all the Ranks. One of the characteristic of rank-k matrices is that the smaller rank-k reduction makes terms more similar whereas the larger rank-k reduction makes terms more dissimilar. If rank-k is equal

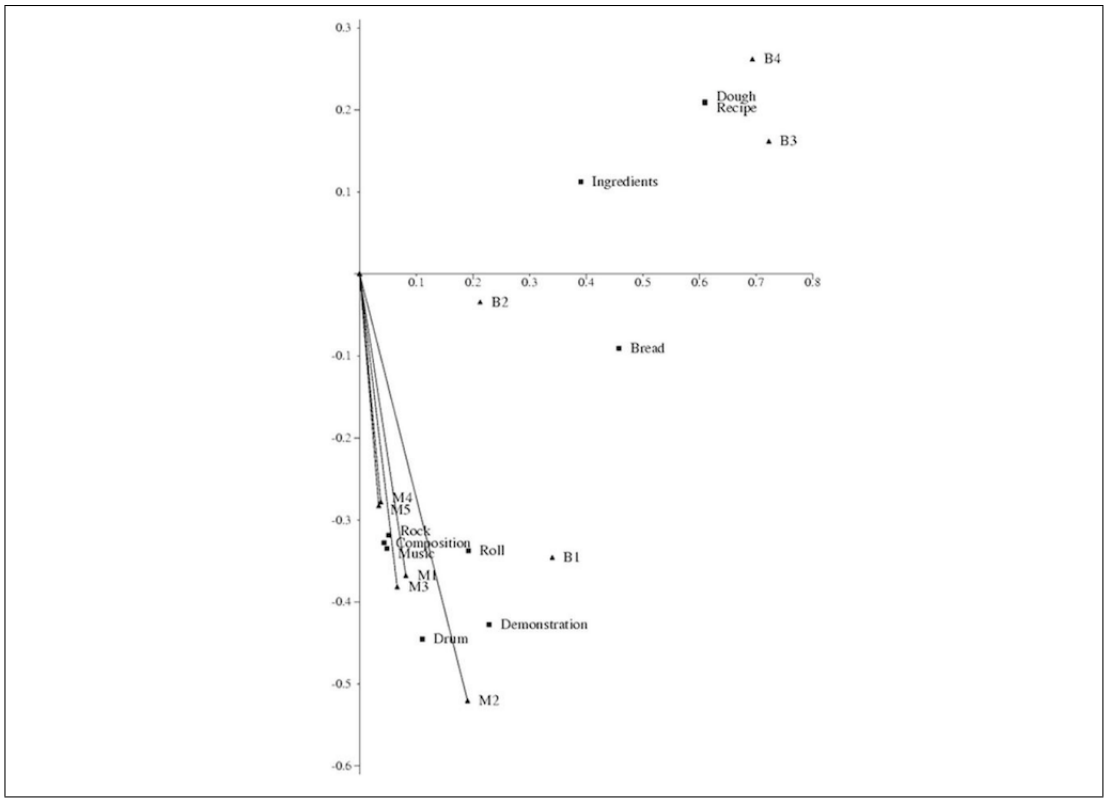


Figure 4.5: LSA Result of the Music-Baking Matrix.

to the rank-m of the original matrix, the values of terms will be equal to the original values. This means that there is no latent semantic information in original matrix, it is only obtained by reducing the original matrix into rank-k matrix.

The 2-dimensional drawing in Figure 4.5 represents the similarity of terms but the actual estimation of similarity between two terms can be obtained as given below. This is called the formula of Euclidean distance.

$$sim(\vec{v}, \vec{u}) = cos(\theta)_{\vec{u},\vec{v}} = \frac{\vec{v}.\vec{u}}{||v||*||u||} = \frac{\sum_{i=1}^{n} v_i * u_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} * \sqrt{\sum_{i=1}^{n} (v_i)^2}}$$

According to the formula given above, there are similarities of word-pairs in the text given below. When the value of $cos(\theta)$ of vectors is approaches to 1, it means they are more similar. If $cos(\theta)$ is negative or near to zero, it means that the vectors are dissimilar.

$sim(music, rock) = 0.99$, $sim(composition, rock) = 0.99$
$sim(music, dough) = -0.2$, $sim(dough, recipe) = 1$

Here $sim(music, composition) = 0.99$ at rank-2 whereas $sim(music, composition) = 1$ at rank-1. The similarity degree increases when the k value of rank-k decreases. This is the expected result because discarding more dimensions implies that the effects of noisy variables are removed in the text. An analogy can be drawn of removing noisy frequencies in the Fourier transform which is a popular transformation in digital image processing. Removing noise from original data is not language specific.

In addition to the Fourier transform, SVD and PCA are also used for noise removing in different areas. However, the best method for error removing in term-document is SVD. Until now, it has not been mentioned that there is a noise notion in the surface structure of text but it is noted that a word keeps its meaning from its co-occurring words. The words which do not have great effects on the sense of a word in a text are accepted as noise. This assumption is at the heart of LSA. Without a threshold value of noisiness, a reduction of the original matrix cannot occur and the hidden semantic relation among words cannot be revealed. The noisiness of a text can be observed by observing the S matrix , the characteristic vectors of text. As shown in Figure 4.4, S matrix is a symmetric and diagonal matrix. The diagonal values are in descending order which means that the a_{kk} has greater effect than ${}_{a}mm$, if $k < m$. If reduction is undertaken at the level of k, the values of $a_{mm}, m > k$ are accepted as noise. Landauer et al. (2013) state that having $S_{kk}, k = 300$ for 100,000 words is sufficient to reveal the hidden semantic relation of words in text. This means that $S_{kk}, k > 300$ are accepted as the eigen values of noisy frequencies and are removed from the semantic space.

To sum up, LSA reflects phenomena which are familiar to human beings. Words occurring in many sentences tend to be ambiguous. It means that it may have many meanings and in this case LSA reduces some of them. Conversely, if a word is less frequent, it means that it has an authentic meaning for a specific topic. For less frequent words, LSA produces high eigen values for those type of words and keep them to use for use in building a meaning for other words. Similarly, Landauer et al. (2013) states that until now LSA is used for synonymy, hypernymy and coherence till now because of its approach in relation to the surface structure of text. These properties of LSA lead to being used in two ways in the thesis work: as a baseline for comparison purposes and to build a semantic space for the document-distance matrix.

4.3 Finding the similarity of documents with LSA

it has been shown that how LSA can reveal the similarities of words in a text. LSA is also used to measure the similarities between the documents (sentences). Landauer et al. (2013) showed that measuring successive sentence similarities reveals the coherence of a text. They achieved this by comparing gradual change of cosine values of successive sentences. If there is a gradual change in the cosine values of sentences, it means that the topic in successive sentences smoothly changes which indicates a coherent text. This section shows how document similarity is performed by LSA.

In section 4.2, the details of the U part of the equation $A = USV^T$ is presented and V was not given. V holds the scalar values of the vector space S for the documents (sentences). Therefore, measuring document similarities should be performed on V. However, V does not hold the frequency values of sentences because there is no notion of the frequencies of sentences. Therefore, V is defined as a function composition of U and S as given below. This means that document similarities are performed with

the help of word frequencies in a text.

$$
\begin{aligned}
A &= USV^T \\
U^TA &= U^TUSV^T \\
&= ISV^T \\
&= SV^T \\
S^{-1}SV^T &= S^{-1}U^TA \\
V^T &= S^{-1}U^TA \\
(V^T)^T &= (S^{-1}U^TA)^T \\
V &= A^TUS^{-1}
\end{aligned}
\tag{4.1}
$$

In equation 4.1, A is the term-document matrix. U is the reduced term matrix. S^{-1} is the matrix of singular values (eigen values) which defines the characteristic vectors of the text. US^{-1} gives the semantic space of text. Multiplying A and US^{-1} projects the scalars of A into the vector space of US^{-1}. Therefore, the term frequencies of A locate the document defined by A in the semantic space of US^{-1}. Positioning a document in a semantic space helps to compare documents. Since A can be any term-document matrix, a pseudo-document can be placed in semantic space. The only restriction is to use same words of the text to define a pseudo-document. For example, taking a document entitled ”Recipe for white bread”. When vector representation of the pseudo-document is put into A in equation 4.1, the pseudo-document will be aligned in the vector space of the documents. The result of the pseudo-document comparison is given in Figure 4.6

Document	*Cosine*
B2: Ingredients for Crescent Rolls	.99800
B3: A Recipe for Sourdough Bread	.90322
B1: How to make Bread and Rolls, a Demonstration	.84171
B4: A Quick Recipe for Pizza Dough using Organic Ingredients	.83396

Figure 4.6: Result of LSA Query

Although there were no re-occurring words between the pseudo-document and B2 in Figure 4.6, B2 is in the result list and in fact, this is the expected result. Despite the lack of re-occurring words, the vector space defined by the words co-occurrence holds latent similarity information. The words ”recipe” and ”ingredients” have co-occurrence in B4 and the bread-rolls pair is co-occurred in B2. Since some components of term-vectors are shared and A is multiplied by the term vector, observing documents that have no co-occurring words in the list (Figure 4.6) is an expected result.

To sum up, the importance of LSA document comparison within this thesis is that

coherence measurement is performed by comparing the cosine values of successive documents in text. Moreover, this thesis does not rely only documents defined by word frequency as in equation 4.1 but it also uses a document-document (doc-doc) matrix to measure coherence. Both comparison methods of coherence (doc-doc and classic LSA style) are used as the baseline to be able to compare the results of research in thesis.

4.4 Studies on Term-Term matrix

The previous section has shown that the similarities of terms and documents are derived from term-vectors U in equation $A = USV^T$. Mill and Kontostathis (2004) and Kontostathis and Pottenger (2006) contributed that the special matrix called term-term matrix which provides information about term similarities. Equation 4.2 shows how the term-term matrix (T) is derived from the term-document matrix (A).

$$T = AA^T \tag{4.2}$$

Similarly, doc-doc matrix can also be derived from the term-document matrix as shown in equation 4.3. D denotes the doc-doc matrix and A denotes the term-document matrix.

$$D = A^T A \tag{4.3}$$

The term-term and doc-doc matrix are derivations of the term-document matrix but they reveal different information about the word frequency of document. Only the doc-doc matrix reveals the distance information of re-occurring words. Since distance information in a text is an object of interest in this thesis, the research here is built on top of information retrieved from the doc-doc matrix. However, without an understanding of the term-term matrix, the importance of the doc-doc matrix cannot be addressed. Therefore, this section focuses on the examination of the term-term matrix which will be a basis for explanation of how the current thesis can be distinguished from studies on the term-term matrix.

The term-term matrix derived from Deerwester's term-document matrix is given in Figure 4.8. The derivation is achieved by equation 4.2. Deerwester's term-document matrix can be found in Figure 4.7.

The term-term matrix in Figure 4.8 is a symmetric matrix and the values on the diagonals are uninformative because of the tautology that exists. If the a_{ij} value of the term-term co-occurrence matrix is non-zero, it means that there is a shared path of co-occurrence between the $a_{i.}$ and the $a_{.j}$ terms in the term-document matrix. For example, $(t1, t5)$ in the term-term matrix has value of 2 which means that *human* and *system* co-occurred in a sentence and this can be seen in column c4 in Figure 4.7. Kontostathis and Pottenger (2006) expanded this observation, and give a mathematical proof which shows that SVD encapsulates the term co-occurrence information

Deerwester term by document matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Figure 4.7: Term-Document matrix of Deerwester (1990).

of the term-term matrix in reduced version of original the term-term matrix. They proved that a connectivity path exists among terms for every nonzero element in the reduced version of the term-term matrix. An observation on the term-term matrix helps to see that there is a transitive co-occurrence path between *human* and *user*: Although *human* and *user* have 0 value in the term-term matrix, *human* co-occurs with *interface*and *interface* co-occurs with *human*.

Deerwester term-to-term matrix

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
human (t1)	×	1	1	0	2	0	0	1	0	0	0	0
interface (t2)	1	×	1	1	1	0	0	1	0	0	0	0
computer (t3)	1	1	×	1	1	1	1	0	1	0	0	0
user (t4)	0	1	1	×	2	2	2	1	1	0	0	0
system (t5)	2	1	1	2	×	1	1	3	1	0	0	0
response (t6)	0	0	1	2	1	×	2	0	1	0	0	0
time (t7)	0	0	1	2	1	2	×	0	1	0	0	0
EPS (t8)	1	1	0	1	3	0	0	×	0	0	0	0
survey (t9)	0	0	1	1	1	1	1	0	×	0	1	1
trees (t10)	0	0	0	0	0	0	0	0	0	×	2	1
graph (t11)	0	0	0	0	0	0	0	0	1	2	×	2
minors (t12)	0	0	0	0	0	0	0	0	1	1	2	×

Figure 4.8: Term-Term co-occurrence matrix of Deerwester (1990).

A nonzero value in the term-term matrix indicates a first order co-occurrence relation among terms. If there is a zero value in the first order co-occurrence matrix but there is a nonzero value in the second order co-occurrence matrix, it means that there is a second order co-occurrence relation between the terms. This is what happens during the observation of *human* and *user* since *human* and *user* has zero value in the first-order term-term co-occurrence matrix, and there is one connectivity (*interface*) between *human* and *user* regarding the term-term matrix, *human* and *user* have a second-order co-occurrence relation (Kontostathis and Pottenger, 2006). *Human* and *user* have a zero value in the first order co-occurrence matrix but have a nonzero value in the second order co-occurrence matrix. Kontostathis and Pottenger (2006) proved that the higher order co-occurrence relation is preserved in the truncated version of the term-term matrix. The authors give a rank-2 truncated version of term-term matrix produced by SVD in Figure 4.9.

The matrix in Figure 4.9 shows how the frequency value and connectivity path among terms are preserved. *human-system* and *human-user* have respectively values of 1.69 and 0.94 whereas *human-tree*, *human-graph* and *human-minors* have negative value, indicating that there are no connective paths. Indeed, this is what was expected because *tree, graph* and *minors* are located in different paragraphs in the term-document matrix. It is reasonable that a connection path between two words, which are located on different paragraphs, does not exist. The word *survey* is a special case in relation to the connectivity path. It is the only word that occurred in both paragraphs listed in

Deerwester term-to-term matrix, truncated to two dimensions

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
human (t1)	×	0.54	0.56	0.94	1.69	0.58	0.58	0.84	0.32	−0.32	−0.34	−0.25
interface (t2)	0.54	×	0.52	0.87	1.50	0.55	0.55	0.73	0.35	−0.20	−0.19	−0.14
computer (t3)	0.56	0.52	×	1.09	1.67	0.75	0.75	0.77	0.63	0.15	0.27	0.20
user (t4)	0.94	0.87	1.09	×	2.79	1.25	1.25	1.28	1.04	0.23	0.42	0.31
system (t5)	1.69	1.50	1.67	2.79	×	1.81	1.81	2.30	1.20	−0.47	−0.39	−0.28
response (t6)	0.58	0.55	0.75	1.25	1.81	×	0.89	0.80	0.82	0.38	0.56	0.41
time (t7)	0.58	0.55	0.75	1.25	1.81	0.89	×	0.80	0.82	0.38	0.56	0.41
EPS (t8)	0.84	0.73	0.77	1.28	2.30	0.80	0.80	×	0.46	−0.41	−0.43	−0.31
survey (t9)	0.32	0.35	0.63	1.04	1.20	0.82	0.82	0.46	×	0.88	1.17	0.85
trees (t10)	−0.32	−0.20	0.15	0.23	−0.47	0.38	0.38	−0.41	0.88	×	1.96	1.43
graph (t11)	−0.34	−0.19	0.27	0.42	−0.39	0.56	0.56	−0.43	1.17	1.96	×	1.81
minors (t12)	−0.25	−0.14	0.20	0.31	−0.28	0.41	0.41	−0.31	0.85	1.43	1.81	×

Figure 4.9: Rank-2 version of Term-Term co-occurrence matrix of Deerwester (1990).

the term-document matrix appearing in C2 and M4. the occurrence value of *survey* points zero, the two paragraphs have no common words. Figure 4.10 shows the finding of Kontostathis and Pottenger (2006) in relation to the reduced term-term matrix when the frequency of *survey* was set to zero.

Modified Deerwester term-to-term matrix, truncated to two dimensions

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
human (t1)	×	0.50	0.60	1.01	1.62	0.66	0.66	0.76	0.45	–	–	–
interface (t2)	0.50	×	0.53	0.90	1.45	0.59	0.59	0.68	0.40	–	–	–
computer (t3)	0.60	0.53	×	1.08	1.74	0.71	0.71	0.81	0.48	–	–	–
user (t4)	1.01	0.90	1.08	×	2.92	1.19	1.19	1.37	0.81	–	–	–
system (t5)	1.62	1.45	1.74	2.92	×	1.91	1.91	2.20	1.30	–	–	–
response (t6)	0.66	0.59	0.71	1.19	1.91	×	0.78	0.90	0.53	–	–	–
time (t7)	0.66	0.59	0.71	1.19	1.91	0.78	×	0.90	0.53	–	–	–
EPS (t8)	0.76	0.68	0.81	1.37	2.20	0.90	0.90	×	0.61	–	–	–
survey (t9)	0.45	0.40	0.48	0.81	1.30	0.53	0.53	0.61	×	–	–	–
trees (t10)	–	–	–	–	–	–	–	–	–	×	2.37	1.65
graph (t11)	–	–	–	–	–	–	–	–	–	2.37	×	1.91
minors (t12)	–	–	–	–	–	–	–	–	–	1.65	1.91	×

Figure 4.10: Rank-2 version of Term-Term co-occurrence matrix of Deerwester (1990).

When *survey* is removed from the paragraphs in the term-document matrix, the disjunction of paragraphs is observed. This shows how the transitive connectivity path of Kontostathis and Pottenger (2006) affects similarity among words and documents.

To sum up, this section has shown the information can be retrieved from a term-term matrix, and furthermore, this information is solely focused on word similarity. This is the reasons why scholars tried to extend LSA to make it also applicable on similarity comparisons of phrases and sentences. Unlike the studies of Mill and Kontostathis (2004) and Kontostathis and Pottenger (2006), this thesis focuses on the doc-doc matrix. The term-term matrix preserves the connectivity paths of the terms in the text but the doc-doc matrix preserves the distance between re-occurring terms. This thesis concerns the doc-doc matrix, and how distance information in this matrix can be used to measure coherence.

4.5 Coherence and LSA

Textual coherence is a production of the mind while reading a text. It addresses how the information flow gradually changes from one part of discourse to another (Landauer et al., 2013). Some of the researches in the field of coherence are; discourse

modeling (Grosz, Weinstein, and Joshi, 1995), effects of coherence on comprehension (Foltz, Kintsch, and Landauer, 1998) and techniques for automated segmentation of discourse (Choi, Wiemer-Hastings, and Moore, 2001). All these studies make assumptions in order to explain the phenomenon of coherence. The current work is limited to the quantification methods of coherence performed through LSA. Discourse coherence in LSA is achieved by comparing the cosine values of successive sentences. Small changes among the cosine values in successive sentences indicate high coherence whereas high changes in these cosine values indicates low coherence (Landauer et al., 2013). In the section 4.3, it was demonstrated that document similarity can be obtained by the frequency values of co-occurring words and this makes the LSA based coherence quantification dependent on word co-occurrence. Chapter 5 reveals that besides word re-occurrence the document-distance matrix can be used while measuring coherence.

Measuring coherence through LSA is performed in two ways as below.

1. Considering the size of textual unit
2. Considering the physical distance of textual unit

The size of textual units mainly focuses on the question of ”which of the textual units (sentence, paragraph, chapter) have to be accepted as an LSA-document?”. The answer will help to measure the varying lengths of the coherence between textual units such as sentence-to-sentence, paragraph-to-paragraph or chapter-to-chapter. In addition, there may be a coherence measurement of sentence-to-paragraph. If there are cosines of sentences in a paragraph and a unique value of cosine of a paragraph, the sentence-to-paragraph coherence can be measured to reveal the relatedness of the target sentences against the topic of the paragraph. Another method is to use moving window technique. A moving window may consist of k or k+1 sentences and compares the next k or k+1 sentences to measure the coherence. The moving window reduces the $m * k$ sentences to k sentences by grouping the m sentences (Landauer et al., 2013).

In this thesis, until now it was the adjacent text units that were the target of measuring the coherence. However, paragraphs which are not adjacent may also contain the concept of coherence. Two distant paragraphs can be used to measure how a topic is persistent across a chapter and this type of coherence is called ”lag coherence”. The same method can also be used to detect the boundaries of chapters in a book. Figure 4.11 shows the changes in cosines that occur when increasing the distance between paragraphs (Landauer et al., 2013). The smoothness in topic changes indicates how neatly the writer organized the topics in a text (Landauer et al., 2013).

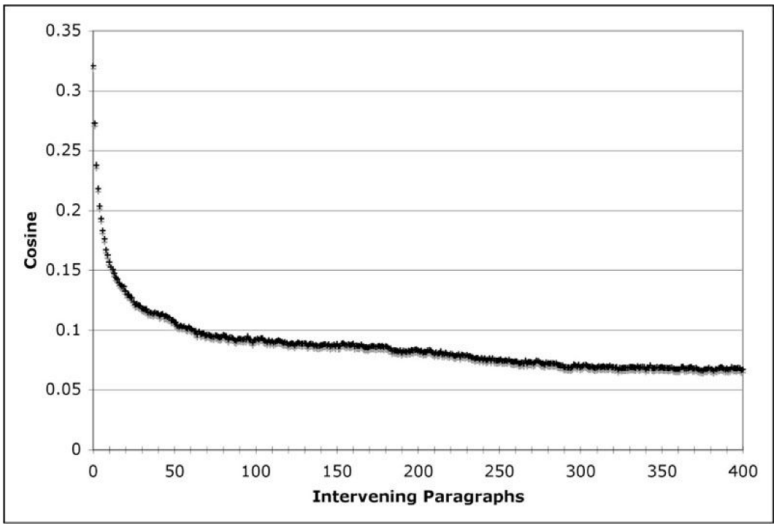


Figure 4.11: Average log cosine as a function of the log distance paragraphs for two textbooks.

4.6 Conclusion

This thesis makes two contributions to LSA studies on coherence. First, in the use of the document-document matrix to measure sentence similarity. Second, the distance of of re-occurring words as a different frequency event. Moreover, in this thesis a method is proposed which uses the document-distance matrix directly without LSA analysis. Chapter 5 contains an explanation about the contributions of distance as event frequency on the quantification of coherence and shows how these contributions differ from current studies.

CHAPTER 5

THESIS WORK

5.1 Introduction

This thesis proposes that the spatial distance between re-occurring words in adjacent sentences can be used to quantify coherence. A study of Kontostathis and Pottenger (2006) focused on quantifying co-occurring word meaning in adjacent sentences and applied LSA on the term-term matrix to reveal the word similarity in the semantic space of the contexted words. The main methodology of Kontostathis and Pottenger (2006) is to compare the quantitative values of words in the semantic space denoted by $A' = (US)(US)^T$ where A denotes the term-term matrix, U denotes term matrix and S denotes the diagonal eigen vectors. The authors proposed a framework to quantify the similarities on term-matrix U of $A' = USV^T$ but did not provide any framework to quantify the document matrix V in the text. It was noted that the word, and document similarity are undertaken on $A' = (US)(US)^T$ and $A' = q^T US^{-1}$ respectively. Since S holds the characteristic vectors of both the term and document matrix, only the term matrix U is used in both equations. Both similarity comparisons are made with the help of the frequency distribution of terms. Sentence similarity is obtained by querying on word similarity in the semantic space denoted by LSA which means that ordering or distance is totally omitted. Section-4.5 explained how coherence detection is accomplished by comparing the sentences in semantic space denoted by $A' = q^T US^{-1}$. This study suggests that the spatial distance of re-occurring words of adjacent sentences posits the degree of similarity of nearby sentences which helps to quantify the coherence in the text. This chapter shows how the document-distance matrix can be used to detect the coherence using the methods explained in (Kontostathis and Pottenger, 2006), and how the document-distance matrix differs from the quantification revealed in Section4.5. This thesis proposes two algorithms. The first one is directly applied on document-distance matrix , and second one is applied to the LSA semantic space of the document-distance matrix.

This section is divided into the following five subsections.

1. Examination of Deerwester's Data
2. Research question in relation to the document-document matrix
3. Algorithm-I assuring the hypothesis
4. Algorithm-II assuring the hypothesis

5.1.1 Examination of Deerwester's Data

The term-term matrix was investigated by Kontostathis and Pottenger (2006) and a brief explanation of this matrix was given in section 4.4. In this section, the same data will be examined to exploit the distance information of re-occurring words. For the examination the reader is referred to Figures 4.7 and 4.8

The *human-system* pair co-occurs in sentence *C4* and the *interface-user* pair co-occurs in sentence *C3* (see Section-4.2 for details).

Multiplying the matrix in Figure-4.7 with its transpose produces the term-term matrix in Figure-4.8. Kontostathis and Pottenger (2006) proved that a nonzero value in the term-term matrix indicates at least one co-occurrence path between the row-term and column-term in the term-document matrix. According to this rule, the nonzero value of t1-t5 in Figure-4.8 reveals that they co-occur in the same sentence. The authors' work also revealed the n^{th} order relations of terms in the term-document matrix by observing the n^{th} order term-term matrix which is obtained by n time multiplication of the term-document matrix with its transpose. Again, according to this rule, although the frequency value of t1-t4 (*human-user*) is zero in Figure-4.8, there is a path of $human \rightarrow interface \rightarrow user$ in Figure-4.8 which indicates that there is a second level term-term relation between *human-user*. Indeed, the nonzero value of *human-user* is observed in the second order term-term matrix which is obtained by two times multiplication with its transpose.

This thesis focuses on the doc-doc matrix listed in Figure-5.1. The matrix listed in Figure-5.1 is symmetric and the diagonals are uninformative due to the tautology. The column and row names of the matrix denote the sentences of the term-document matrix. The nonzero value of a_{ij} of the matrix given in Figure 5.1 denotes the number of shared terms between sentences of $a_{i\_}$ and $a_{\_j}$.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	1	1	1	1	0	0	0	0	0
s2	1	1	2	2	3	0	0	0	1
s3	1	2	1	3	1	0	0	0	0
s4	1	2	3	1	0	0	0	0	0
s5	0	3	1	0	1	0	0	0	0
s6	0	0	0	0	0	1	1	1	0
s7	0	0	0	0	0	1	1	2	1
s8	0	0	0	0	0	1	2	1	2
s9	0	1	0	0	0	0	1	2	1

Figure 5.1: Doc-Doc matrix of Deerwester (1990).

The doc-doc matrix reveals different information about the text when it is compared with the term-term matrix. This can be observed in the frequency values of the doc-doc matrix. The frequency values of matrices of the doc-doc and term-term denote different statistical values of the term-document matrix. The doc-doc frequency indicates which sentences share how many words whereas the term-term frequency indicates the number of times words co-occur. For example, $(s_2, s_5) = 3$ in the doc-doc

matrix indicates that second and fifth sentences share 3 words. $(s_1, s_6) = 0$ indicates that first and sixth sentences do not share any words. The number of shared words among sentences cannot be obtained from the term-term matrix. Moreover, the doc-doc matrix reveals the distance of shared words among sentences. For example, $(s_2, s_5) = 3$ indicates that there are two sentences sharing three words and their distance value is two $(5 - 2 - 1 = 2)$. This observation is the main motivation of this thesis for both the proposed algorithms. The motivation is restated as two statements to clarify the distinction between the doc-doc matrix and the term-term matrix as below.

1. The doc-doc matrix reveals the shared words but does not give any information about which word is shared whereas the term-term matrix indicates the co-occurring words but does not give any information about which words re-occurred in which sentences.

2. The doc-doc matrix indicates that how far the sentences spatially share the same words whereas the term-term matrix does not give any information about distance.

With the help of these observations, it is possible to take closer look at the doc-doc matrix in Figure 5.1. The first row of the doc-doc matrix shows that sentences sharing the re-occurring words of s_1. S_1 has the row vector of $\vec{s_1} = [1, 1, 1, 0, 0, 0, 0, 0]$ which indicates that it shares words with s_2, s_3 and s_4 but has no shared words with s_5,s_6,s_7,s_8. $\vec{s_1}$ has one less dimension since the first element of row vector of $\vec{s_1}$ is removed because of uninformativity. Since the re-occurrence of words of sentences is preserved in the doc-doc matrix, sentence to sentence similarity can be revealed with the help of re-occurrence words.

According to these explanations, it can be said that s_1 may have a close similarity with s_2,s_3,s_4 because of the shared words but may have not a close similarity with s_5,s_6,s_7,s_8 because there are no shared words. This inference can be generalized on any a_{ii} element of the doc-doc matrix. Next, the doc-doc matrix in Figure-5.2 gives information about how close the shared words are to each other. For example, s_2 has two words shared with s_3 and s_4 but has three words shared with s_5. This observation may be interesting because intuition leads to the idea that says that close sentences tend to share more words than more distant sentences. However, although s_5 is further away according to s_3 and s_4 it shares one more word with s_2. The distance between s_2 and s_5 appears to be an exception but this raises question is whether this is true. Considering a coherent paragraph in which the last sentence of the paragraph may share similar words with first sentence due to the aim of supporting the topic mentioned in the first sentence. This may also happen between the abstract and summary sections of an article. This observation may lead our intuition to infer that having more shared words between distant sentences may be a cohesive cue about textual coherence. This thesis follows this intuition and next section describes building an hypothesis based on this observation and intuition to discover a practical method that allows for the quantification of textual coherence of sentences in a text using distance information of the doc-doc matrix.

5.1.2 Research question of this thesis

After observing the doc-doc matrix, it can be seen that the doc-doc matrix is more informative than term-term matrix on the basis of sentence similarity. According to these observations, the following research question was constructed.

1. Can the spatial distance of re-occurring words in adjacent sentences quantify coherence?

After observation of the doc-doc matrix, it can be seen that sentences having more shared words tend to be closer to each other and distant sentences tend to have less shared words. Moreover, although there are distant sentences having more shared words, they may still exist in the same paragraph such as first and last sentences of a paragraph. According to these inferences, after observing the doc-doc matrix, this thesis proposes two algorithms which aim to validate the observation done on doc-doc matrix. To validate the algorithms, the steps below are followed.

1. Generate pseudo-random data which each sentence has two shared words.
2. Apply Algorithm-I and Algorithm-II to pseudo-random data and see that algorithms cannot detect any coherence cue quantitatively.
3. Apply Algorithm-I and Algorithm-II to real data and observe that algorithms can detect coherence quantitatively.
4. Compare results of proposed algorithms with the LSA results.

The thesis proposes two algorithms that are based on different approaches. Algorithm-I uses an authentic approach and Algorithm-II uses the distance cues of re-occurring words to build an input matrix for LSA. The data on which the algorithms operate are purified by operations below.

1. Pronoun resolution
2. Anaphora resolution
3. Inflected words are introduced as lexeme
4. Simple sentences and clauses are introduced as sentences

Algorithm-I is presented in the next section.

5.2 Algorithm-I

Algorithm-I operates on a reduced version of the doc-doc matrix given in Figure 5.2. The doc-doc matrix is the derivation of the term-document matrix given in Figure 4.7.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	x	1	1	1	0	0	0	0	0
s2	0	x	2	2	3	0	0	0	1
s3	0	0	x	3	1	0	0	0	0
s4	0	0	0	x	0	0	0	0	0
s5	0	0	0	0	x	0	0	0	0
s6	0	0	0	0	0	x	1	1	0
s7	0	0	0	0	0	0	x	2	1
s8	0	0	0	0	0	0	0	x	2
s9	0	0	0	0	0	0	0	0	x

Figure 5.2: The reduced version of Doc-Doc matrix of Deerwester (1990).

Since the doc-doc matrix is a symmetric matrix, the left triangle of the matrix is set to zero.

Algorithm-I is a set of mathematical operations carried out on the doc-doc matrix while traversing its diagonal. Each traverse step creates 4 hypothetical rectangles: top diagonal, bottom diagonal, bottom left rectangle and top right rectangle. The rectangles are shown in Figure 5.3.

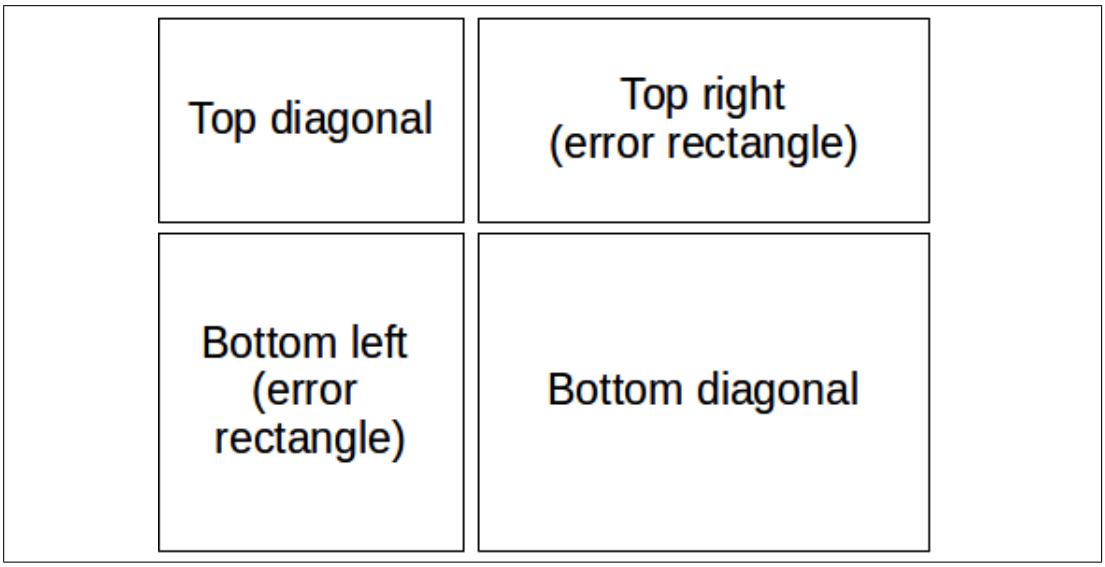


Figure 5.3: Four rectangles of Algorithm-I

The rectangles produced at third step of traversing is given in Figure 5.4. The traversing is undertaken from a_{00} to a_{88}.

The top diagonal holds the frequencies of shared words of first three sentences. The bottom diagonal stores the frequencies of the shared words of the last six sentences. The top right rectangle stores the frequencies of the shared words between the first three (in top diagonal) and the last six sentences (in top right rectangle). Assuming that the third sentence perfectly divides 9 sentences into two paragraphs and the paragraph has no shared words as shown in Figure 5.5. In this scenario, the top right rectangle will be a zero matrix. Since there is no such perfection in a natural text, there will be some frequency values in the top right rectangle. Moreover, sentences far from each other do not frequently share words. Therefore, the top right rectangle

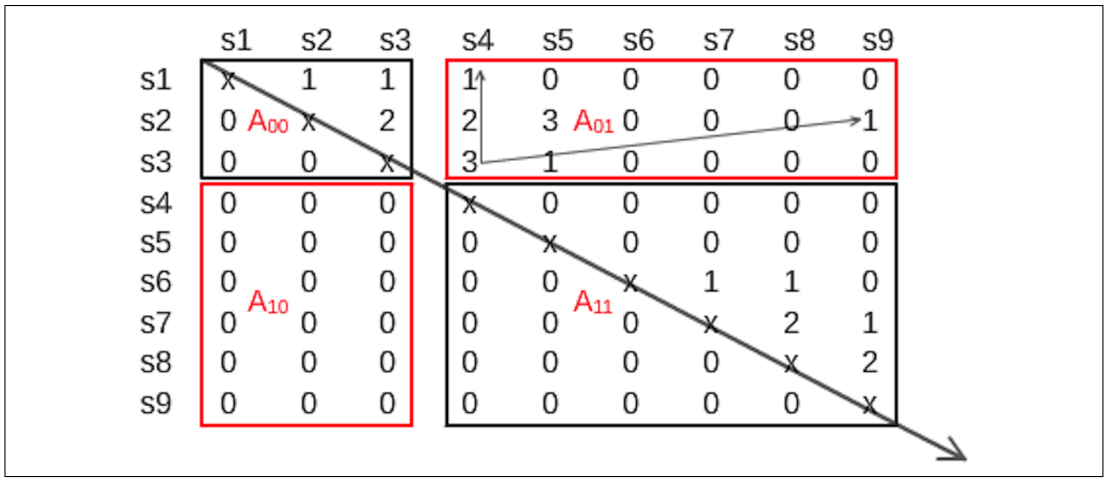


Figure 5.4: Algorithm-I for Doc-Doc matrix of Deerwester (1990).

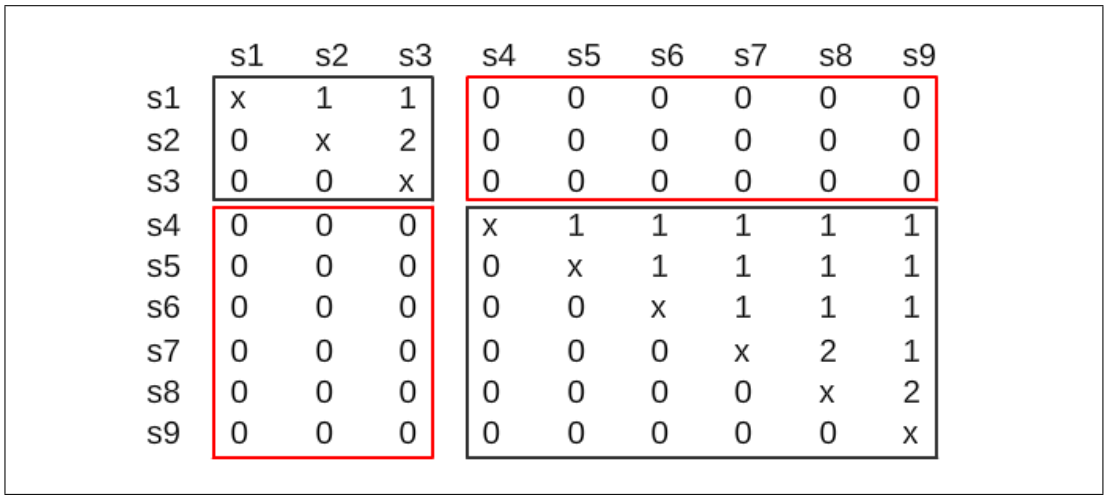


Figure 5.5: Hypothetically perfect data.

should be a sparse matrix which is shown in Figure 5.4. There is one further observation in that the left part of the top right rectangle has more nonzero value than the right part of the rectangle has. This is also expected because left part of the top right rectangle tends to be at the center of the text. Through these explanations, Algorithm-I assumes that the top right and bottom left rectangles indicate an inconsistency in the coherence of the text. Thus, the top right and the bottom left rectangles are labeled *Error Rectangles*. Algorithm-I introduces the mathematical formula; *Error Function* for *Error rectangles*. This error function is basically a weighting function that makes the left part of the error rectangle less erroneous and makes the frequencies of the right part of the error rectangle more erroneous. This weighting is performed according to the spatial distance of elements in the doc-doc matrix. For example, for the third traversing step, $(s_1, s_4) = 1$ should have a lower error value than $(s_2, s_9) = 1$. For the purpose of weighting, a euclidean distance function is introduced as given in equation 5.1.

$$|\vec{a}_{mn}| = 1 - tf(a_{mn})\frac{1}{|\vec{a}_{mn}|}, \qquad tf : term\ frequency \tag{5.1}$$

The distance function calculates (s_1, s_4) and (s_2, s_9) as below.

$|(0,2)| = 1 - \frac{1}{1*\sqrt{0^2+2^2}} = 0.5$
$|(1,5)| = 1 - \frac{1}{1*\sqrt{1^2+5^2}} = 0.8$

The distance function shows that a distant shared word $(1,5) = 0.8$ has more error value than the nearby shared word $(0,2) = 0.5$ at the time of third step of traversing.

According to these explanations, Algorithm-I has assumptions given below.

1. Moving on the diagonal means that sentences are read successively.
2. Adjacent sentences have to share more words than spatially distant sentences.
3. On the i^{th} move, the elements of the error rectangle are likely to be zero when they are becoming spatially distant.
4. On the i^{th} move, the far element in the error rectangle has less effect on coherence.
5. The top and bottom diagonal rectangles are tend to be two distinct paragraphs if the diagonal rectangles tend to be full of nonzero values and the error rectangles tend to be full of zero values.

There are the following plausible explanations of these assumptions. Moving on the diagonal of a symmetric matrix is the same as moving along rows or columns. Therefore, Item-1 is true by definition. Item-2 is likely to be true because at the point of (s_3,s_5), s_3 shares words with its adjacent sentences (s_1,s_2,s_4 and s_5) and shares no words with s_6,s_7,s_8,s_9 which are not adjacent relatively. Item-3 implicitly indicates that the sentences distant from the current sentences are likely to have no shared words which is expected from a coherent paragraph. Indeed, the shared words have to change gradually. Therefore, violating this assumption has to be considered as erroneous. This makes Item-3 plausible.

In fact, Item-4 is a result of Item-3 and Item-5 is about change of topics in a paragraph. This implies that if paragraph boundary is reached at a certain step while traversing the doc-doc matrix, the two diagonal rectangles should indicate two distinct paragraphs. Given that paragraph-1 (P1) and paragraph-2 (P2) have two different topics such as T1 and T2. The P1 sentences will have more shared words among themselves and P2 sentences will have more shared words among themselves. Letting S_n^{p1} be the last sentence of P1 and S_1^{p2} is the first sentence of P2, the cosine value of S_n^{p1} will be close to the mean cosine value of P1 sentences. This is the same for S_1^{p2}. The cosine value of S_1^{p2} will be close to the mean cosine value of P2 sentences. The cosine values of S_n^{p1} and S_1^{p2} must differ because their semantic space is most likely constructed by different words. This inference can also be satisfied by observing the four rectangles of doc-doc matrix. If there is a paragraph boundary, the top and bottom diagonals must contain nonzero values and error rectangles must contain zero values. This occurred at (s_5, s_5) in Figure 5.4. The top and bottom diagonals have are nearly full of nonzero values and the error rectangles are nearly full of zero values. There is

only one nonzero value at (s_2, s_9) in the top right error rectangle. This makes Item-5 plausible.

Using the assumptions given above, the definitions below are introduced for the m^{th} step of traversing in the doc-doc matrix.

Top Diagonal: $a_{00} = \sum_{i=0}^{m} \sum_{j=0}^{m} a_{ij}$

Bottom Diagonal: $a_{11} = \sum_{i=m+1}^{n} \sum_{j=m+1}^{n} a_{ij}$

Top Right: $a_{10} = \sum_{i=m}^{n} \sum_{j=0}^{n} ErrFunc(a_{(i-m),(n-m-j+1)})$

Bottom Left: $a_{01} = a_{10}$

ErrFunc: *Euclidean distance function*

For each traversing step, the definitions given above are merged into the one matrix; *Distance Matrix* (DM).

An outline of the merge operation is given in Figure-5.6.

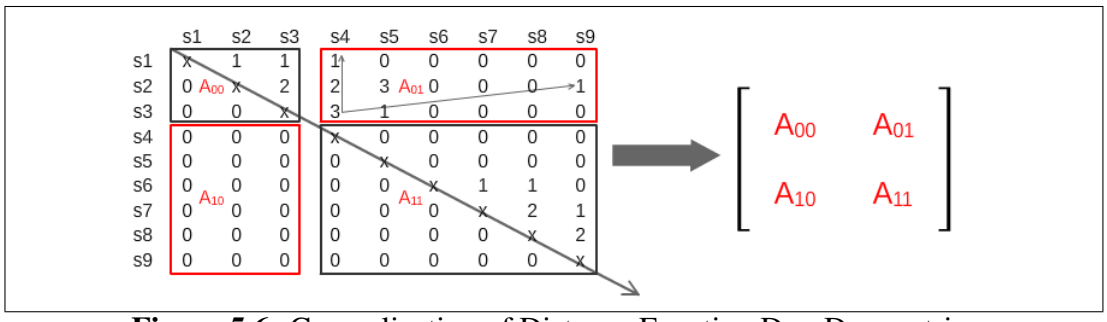


Figure 5.6: Generalization of Distance Function Doc-Doc matrix.

The similarity between sentences is obtained performed by the equation 5.2. T and B denotes top and bottom diagonals at step m of Algorithm-I.

$$sim(A_{mn}) = \frac{ErrorFunc(A_{mn}) * ErrorFunc(A_{mn})}{|T_{mn}| * |B_{mn}|} \tag{5.2}$$

The steps of the Algorithm-I is defined as below.

Data: The frequency matrix of a text
Result: The list of coherence distance of the frequency matrix
initialization;
DistanceValues = [];
docDocMatix = docdoc(Data);
while *not end of the matrix* **do**
    m = read next diagonal ;
    m00 = buildTopDiagonal(m);
    m01 = buildBottomDiagonal(m);
    m10 = buildTopRightErrorRectangle(m);
    m11 = m10;
    MM = buildDistanceMatrix(m00,m91,m10,m11);
    DistanceValue = $1/|m[0][0] - m[1][1]| - m[0][1]$;
    put DistanceValue in DistanceValues;
end
return DistanceValues;

The algorithm is applied to the following data.

1. Random Data
2. Deerwester's Data (1990)
3. Music-and-Baking Data of Landauer et al. (2013)
4. Data of *Word Meaning and Discourse understanding Lecture* of University of Cambridge
5. 7 pairs of paragraphs taken from the book "Introduction to Psychology" (Stangor, 2010)

5.2.1 Results of the application of Algorithm-I applied to random data

While constructing pseudo-random data, the observations are obtained for a term-document matrix contained 26 sentences and 78 words. The values of 26 and 78 are selected for two reasons; first, these are the largest numbers in the real data set for this study and second, the result of Algorithm-I on random data can easily be compared with the result obtained from the real data.

1. One shared word between sentences results in a diagonal doc-doc matrix. This makes the error rectangles zero and sentences highly dissimilar.
2. Two shared words between sentences results in random dissimilarities among sentences.
3. Three shared words between sentences results in a high similarities among sentences.

In relation to the observations given above, Item-1 and Item-3 are discarded because of uninformativity. The purpose of producing pseudo-random data is to have a pseudo document which has randomly connected cues on surface structure of text, resulting in a coherent pseudo-document for each random data set. Therefore, algorithm-I is tested on data having two re-occurrence of words for successive sentences.

The result of Algorithm-I on random data is given in Figure 5.7. The horizontal labels in Figure 5.7 denotes which successive sentences were compared. Number 1 on horizontal label refers the comparison of sentence-1 and sentence-2 of the text similarly, sentence-25 refers the comparison of sentence-25 and sentence-26 of the text. The result of Algorithm-I applied to random data indicates that it is is highly dependent on the re-occurring words in a text. If there is no re-occurrence of words, it detects high dissimilarity and if there are re-occurrence of words, it detects similarities.

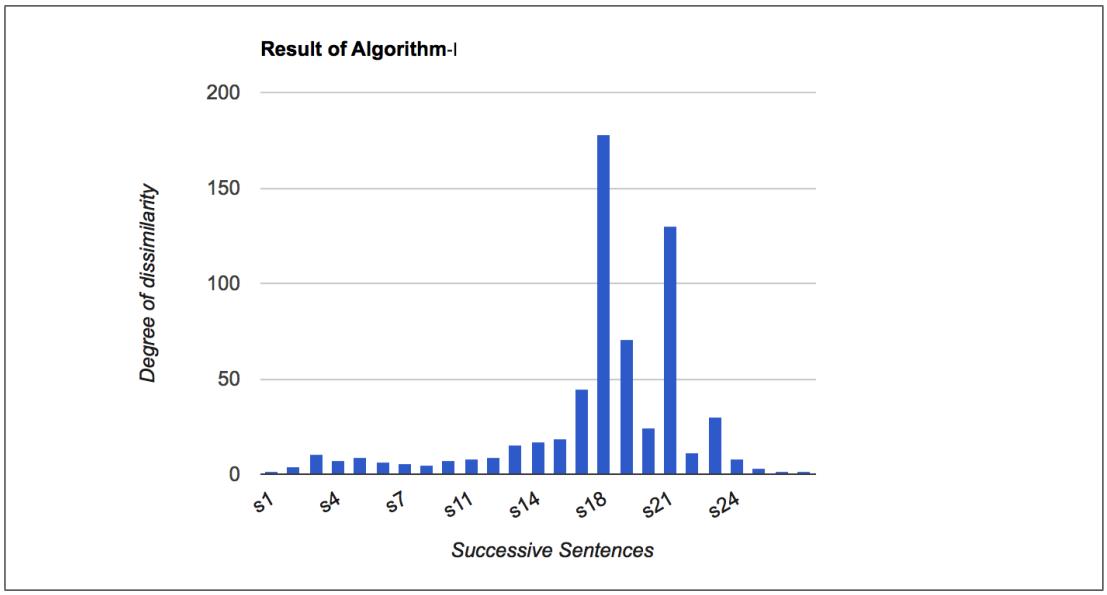


Figure 5.7: The result of Algorithm-I on Random Data.

5.2.2 Results of the application of Algorithm-I applied to Deerwester's data

Algorithm-I is applied on Deerwester's Data (1990) listed in Table 5.1. In this thesis, the labels s1-s9 in Table 5.1 were added to help in the matching of the result of Algorithm-I.

Table 5.1: Deerwester's data (1990).

*s*1	c1	Human machine interface for Lab ABC computer applications
*s*2	c2	A survey of user opinion of computer system response time
*s*3	c3	The EPS user interface management
*s*4	c4	System and human system engineering testing of EPS
*s*5	c5	Relation of user-perceived response time to error measurement
*s*6	m1	The generation of radon, binary, unordered trees
*s*7	m2	The intersection graph of paths in trees
*s*8	m3	Graph minors IV: Widths of trees and well-quasi-ordering
*s*9	m4	Graph minors: A survey

There are two paragraphs in Deerwester's Data labeled as s1 to s5 and s6 to s9. Algorithm-I is expected to detect the boundaries of two paragraphs listed in Table 5.1: s5-s6. The result of Algorithm-I is listed in Figure-5.8 in which the horizontal labels denote the sentences listed in Table 5.1. The performance of Algorithm-I is as expected with the high value of s5-s6 in Figure-5.8 indicating the dissimilarity between s5 and s6.

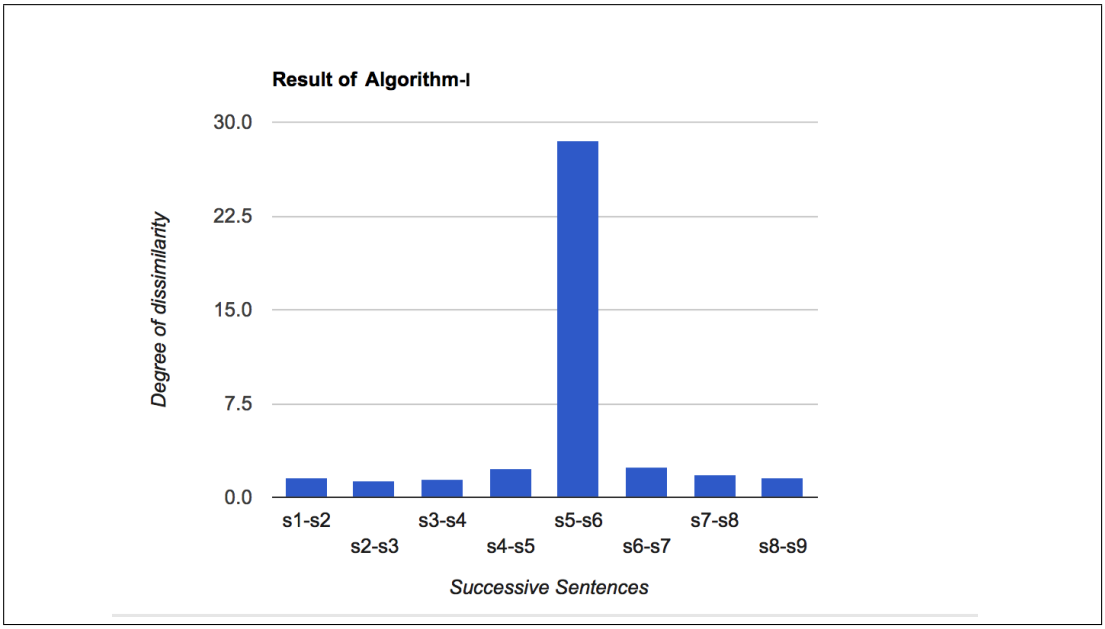


Figure 5.8: The result of Algorithm-I on Deerwester's Data.

5.2.3 Results of the application of Algorithm-I applied to Music And Baking Data (Landauer et al., 2013)

Algorithm-I was applied to Music and Baking Data of Landauer et al. (2013). The data is listed in Table 5.2 in which labels s1 to s9 have been added for the purposes of this thesis.

Table 5.2: Music and Baking data (Landauer et al., 2013).

$s1$	c1	Rock and Roll music in 1960's
$s2$	c2	Different drum rolls, a demonstrations of techniques
$s3$	c3	Drum and bass composition
$s4$	c4	A perspective of rock music in the 90's
$s5$	c5	Music and composition of popular bands
$s6$	m1	How to make bread and rolls, a demonstration
$s7$	m2	Ingredients for crescent Rolls
$s8$	m3	A recipe for sourdough bread
$s9$	m4	A quick recipe for pizza dough using organic ingredients

There are two paragraphs in the Music and Baking Data of Landauer et al. (2013) labeled as: s1-s5 and s6-s9, inclusive. Algorithm-I is expected to detect the boundaries of the two paragraphs s5-s6 in Table 5.2. The results of Algorithm-I is given in Figure-5.9. In Figure 5.9, the horizontal labels denote the sentences listed in Table 5.2. The result of Algorithm-I is as expected with the high value of s5-s6 in Figure-5.9 indicating the dissimilarity between the s5 and s6.

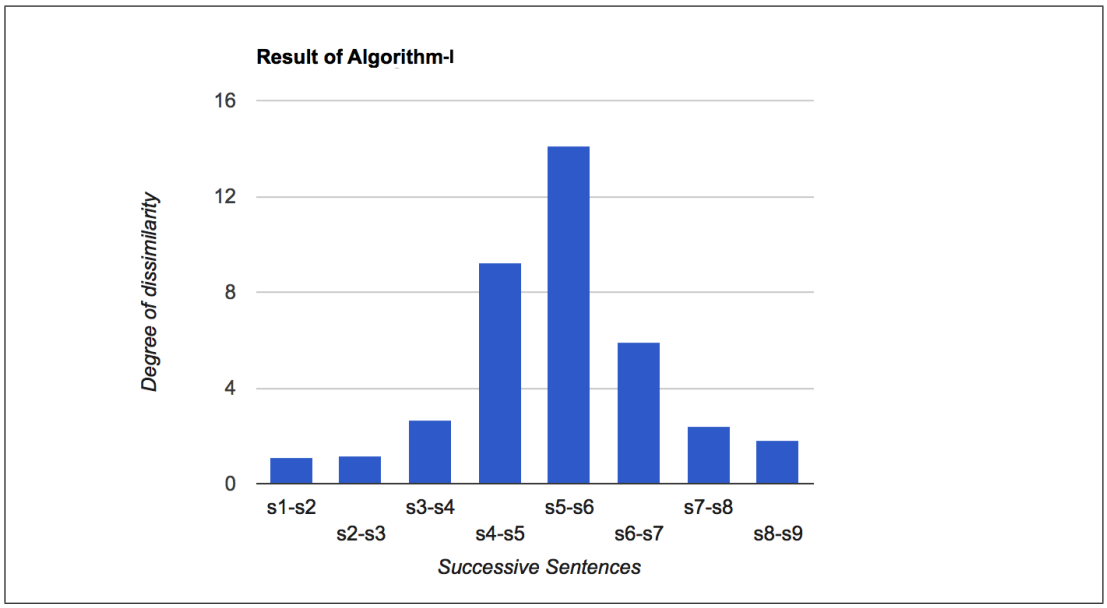


Figure 5.9: The result of Algorithm-I applied to Music and Baking Data.

5.2.4 Result of the application of Algorithm-I applied to a chapter of a book

Algorithm-I was applied to text taken from a linear algebra book. The data listed in Figure 5.10 is a truncated version of the real data which is presented in Figure C.5. The numbers in Figure 5.10 denote the order of sentences in the doc-doc matrix. The number in parenthesis denotes the original location of the sentences in the book. According to the numbers in parenthesis, the original location of sentence-3 was at 28, the real location of sentence-4 was at 51. Moreover, sentence-16 is first sentence

of a new paragraph. As a result, there are three locations in the text for paragraph boundaries: sentence-3, sentence-4 and sentence-16.

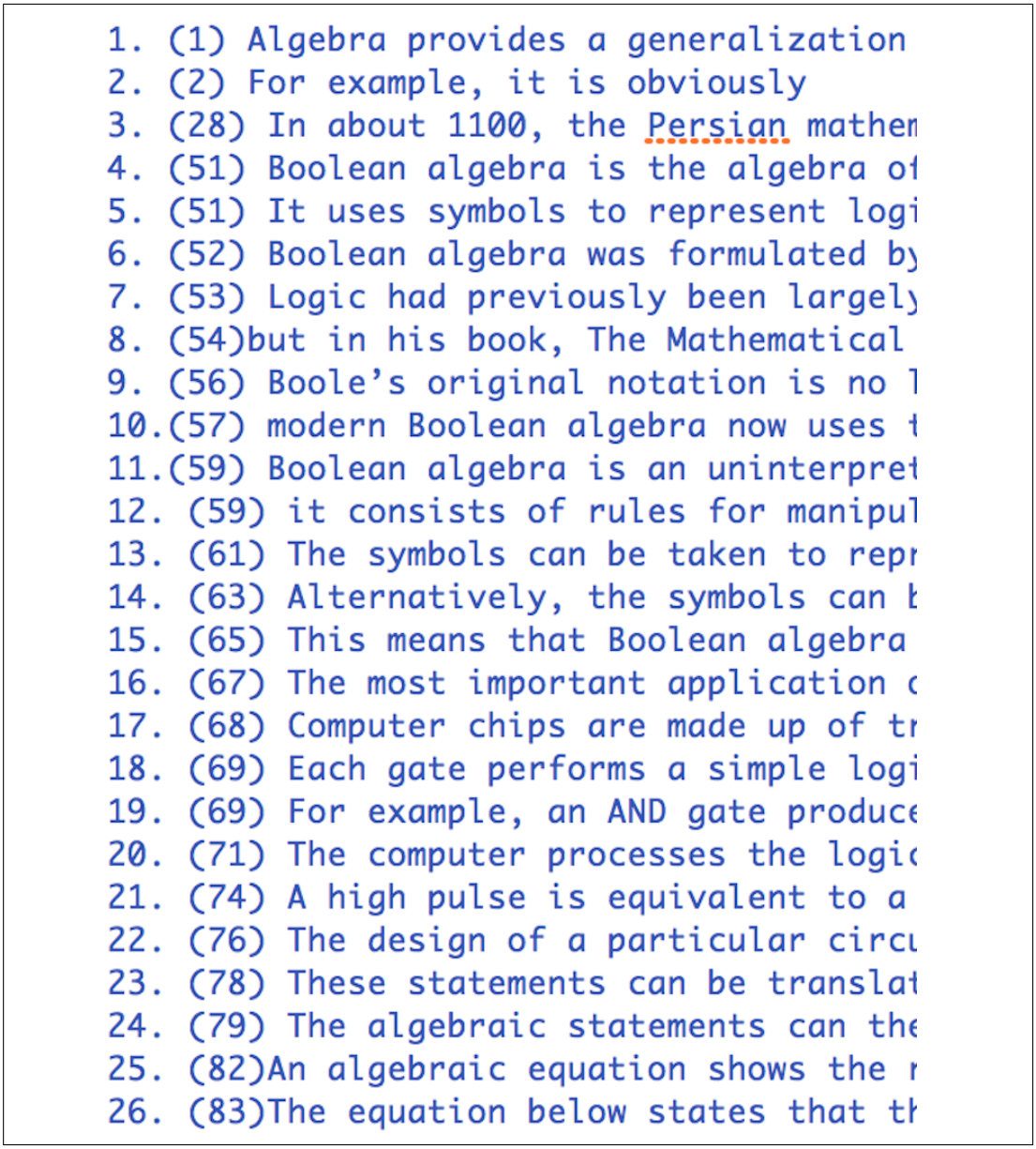


Figure 5.10: The result of Algorithm-I applied to text from a linear algebra book.

Algorithm-I was expected to detect sentence dissimilarity of the data given in Figure 5.10. The result the application of Algorithm-I is given Figure 5.11. The horizontal labels denotes the successive sentences, and the y-axis shows the dissimilarity between successive sentences. According to the results shown in Figure 5.10, Algorithm-I detects sentence dissimilarities at sentence-3 and sentence-20. However, the original text has paragraph boundaries at sentence-3 and sentence-16. Algorithm-I detects the paragraph boundary at sentence-3 but it detects the paragraph boundary at sentence-16 with an error rate of 18% for 22 sentences.

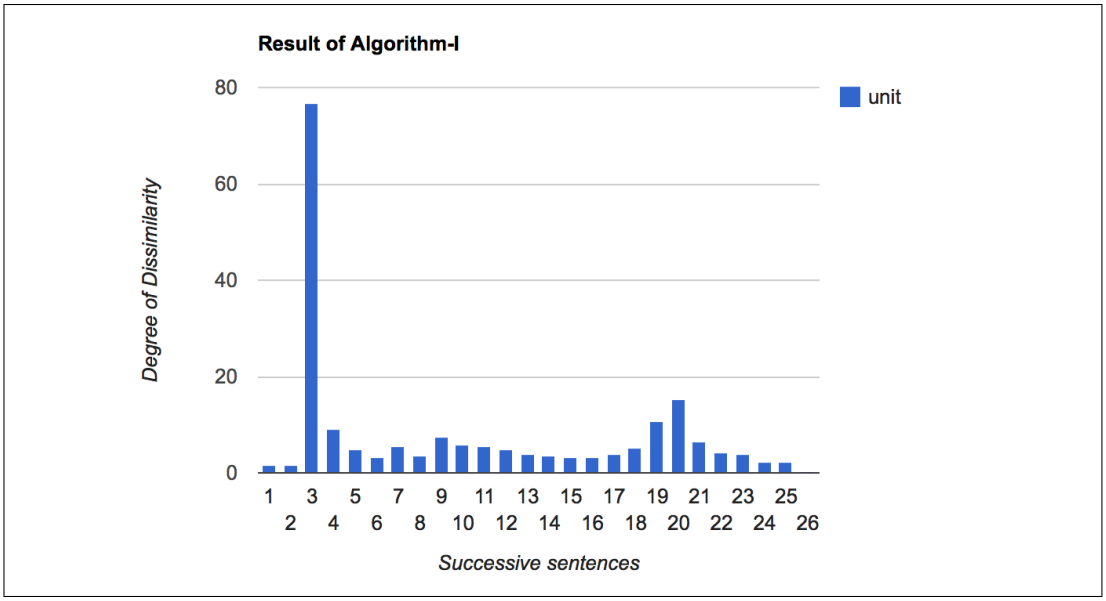


Figure 5.11: The result of Algorithm-I applied to a book chapter.

5.2.5 Results of the application of Algorithm-I applied to the book "Introduction to psychology" (Stangor, 2010)

Algorithm-I is applied to 7 pairs of paragraphs taken from the book. The data is provided in Appendix C. Below are the characteristics of the data.

1. Sample-1 has a paragraph boundary between sentence-6 and sentence-7 (s6-s7)
2. Sample-2 has a paragraph boundary between sentence-8 and sentence-9 (s8-s9)
3. Sample-3 has a paragraph boundary between sentence-3 and sentence-4 (s3-s4)
4. Sample-4 has a paragraph boundary between sentence-6 and sentence-7 (s6-s7)
5. Sample-5 has a paragraph boundary between sentence-9 and sentence-10 (s9-s10)
6. Sample-6 has a paragraph boundary between sentence-9 and sentence-10 (s9-s10)
7. Sample-7 has a paragraph boundary between sentence-9 and sentence-10 (s9-s10)

The results of the Algorithm-I are as follows.

1. Algorithm-I detects a paragraph boundary at s6-s7 and s3-s4 for sample-1.
2. Algorithm-I detects a paragraph boundary at s2-s3 and s8-s9 for sample-2.
3. Algorithm-I detects a paragraph boundary at s2-s3 and s3-s4 for sample-3.
4. Algorithm-I detects a paragraph boundary at s6-s7 for sample-4.

5. Algorithm-I detects a paragraph boundary at s3-s4 and s9-s10 for sample-5.

6. Algorithm-I detects a paragraph boundary at s9-s10 for sample-6.

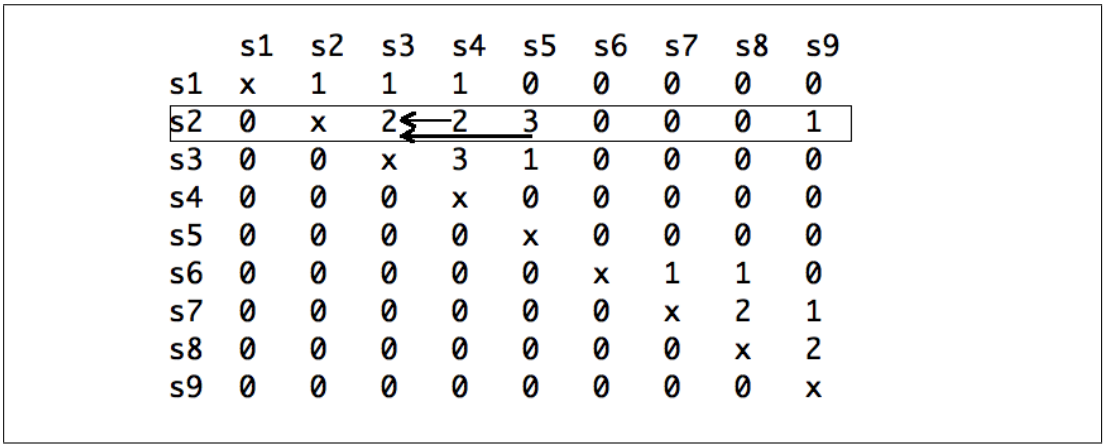
7. Algorithm-I detects a paragraph boundary at s9-s10 for sample-7.

5.2.6 Conclusion

To sum up, Algorithm-I detects 9 real paragraph boundaries for the 14 detected paragraph boundaries giving a success rate of $9/14 = 0.64$. In previous sections, it was explained that *Error Rectangles* correlates current sentences and further sentences. A new definition of the *Error Function* on *Error Rectangle* may reduce error range of boundary detection capability of Algorithm-I. Moreover, Algorithm-I is not designed for large corpus, currently. This is a limitation because the algorithm was designed to detect the paragraph boundary of two successive paragraphs. For multiple paragraphs, it must be iterated on paragraphs while a sized window is spanned over the paragraphs as implemented in HAL (Lund, Burgess, and Atchley, 1995; Lund and Burgess, 1996). Overall, it can be seen the results show that Algorithm-I has promise for further studies on the document-distance matrix. The next section introduces Algorithm-II which used the document-distance matrix as a weighting matrix of LSA.

5.3 Algorithm-II

The target of Algorithm-II is to create a local weighting matrix of the document-distance matrix in which each item of the document-distance matrix is determined according to relative distance among the re-occurring words of the sentences of the doc-doc matrix.



	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	x	1	1	1	0	0	0	0	0
s2	0	x	2	2	3	0	0	0	1
s3	0	0	x	3	1	0	0	0	0
s4	0	0	0	x	0	0	0	0	0
s5	0	0	0	0	x	0	0	0	0
s6	0	0	0	0	0	x	1	1	0
s7	0	0	0	0	0	0	x	2	1
s8	0	0	0	0	0	0	0	x	2
s9	0	0	0	0	0	0	0	0	x

Figure 5.12: Term-Document matrix of Deerwester (1990).

One of the assumption of Algorithm-I is that distance sentences having more shared sentences may indicate high relatedness when compared to intervening sentences and the example of the first and summary paragraph of a chapter of a book was given as

an example. The same assumption and argumentation are applicable to Algorithm-II. However, Algorithm-II uses a different approach. Figure-5.12, shows that s_5 has more shared words with s_2 compared to s_4. However, s_4 is spatially closer to s_2. This raises question of which indicates the degree of similarity of sentences: shared words or spatial distance? Algorithm-II uses both of the shared words and spatial distance. To achieve this, a *Distance Closure Function* is introduced as a local weighting function defined as below. A denotes the document-distance matrix and D denotes the document-document matrix.

Distance Closure Function: $A_{ij}^{dist} = 2^{-1*|j-i|+D_{ij}-1}$

The *Distance closure function* derives a weighting matrix from the document-distance matrix with respect to the degree of distance of shared words. The result of the *Distance Closure Function* on the matrix presented in Figure-5.12 is listed in Figure-5.13.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	1	0.5	0.25	0.125	0	0	0	0	0
s2	0.5	1	1	0.5	0.5	0	0	0	0.007813
s3	0	0	1	2	0.25	0	0	0	0
s4	0	0	0	1	0	0	0	0	0
s5	0	0	0	0	1	0	0	0	0
s6	0	0	0	0	0	1	0.5	0.25	0
s7	0	0	0	0	0	0	1	1	0.25
s8	0	0	0	0	0	0	0	1	1
s9	0	0	0	0	0	0	0	0	1

Figure 5.13: Result of Distance Closure Function.

After the application of the *Distance Closure Function*, s_4 and s_5 have the same weighting values (0.5) for s_2. This is what was intended by introduction of the *Distance Closure Function*. Below is a generalized version of assumptions introduced for Algorithm-II.

1. The similarity of two sentences is positively affected by the increasing function of spatial distance, and dissimilarity of two sentences is negatively affected by the decreasing function of spatial distance.
2. A low distance between sentences in the doc-doc matrix is an indication of their relatedness to the same topic.
3. A high frequency of shared terms of distant sentences should make distant sentences close to each other as in example of the first paragraph, and the summary paragraph of a chapter of a book.

By introducing the *Distance Closure Function*, in fact, assumption-1 has been validated. Assumption-2 can be generally accepted. Assumption-3 can be accepted as a result of assumption-1 because although the $distance(s_2, s_5) = 3$, s_5 has the same distance value 0.5 as s_4.

Since these assumptions are sufficiently plausible to operate on the obtained data , the weighting matrix derived from document-distance matrix can be used to measure

sentence similarities. Algorithm-II uses the weighting matrix as an input matrix of LSA. In Algorithm-II, the semantic space (vector space) is constructed by applying the SVD process to the weighting matrix.

The steps for Algorithm-II are given below.

Data: The term-document matrix of a text
Result: The list of coherence distance of the frequency matrix
initialization;
//derive doc-doc matrix from term-doc matrix
docDocMatix = docdoc(Data);
//derive weighting matrix
docDistanceMatrix = calculateDocDistance(docDocMatrix);
// use weighting matrix as input matrix of LSA
(u,s,v) = applySingularValueDecompositionofLSA(docDistanceMatrix);
sreduced2 = reduceDimensionTo2(s);
//build semantic space
vs= v * sreduced2;
cosineValuesOfSentences = findCosineValuesOfVectors(vs);
return cosineValuesOfSentences;

The algorithm differs from the classical LSA document comparison in two ways: First, it uses the document-distance matrix. Second, it is based on the document-document matrix.

The algorithm was applied to the following data.

1. Random Data
2. Deerwester Data(1990)
3. Music-and-Baking Data of Landauer et al. (2013)
4. Data from the lecture; *Word Meaning and Discourse understanding Lecture* from the University of Cambridge
5. 7 pairs of paragraphs taken from the book ”Introduction to Psychology” (Stangor, 2010)

5.3.1 Result of Algorithm-II on Random Data

Algorithm-II used the same random data as Algorithm-I. Since Algorithm-II calculates similarities in vector space, the mean cosine values of adjacent sentences are used for comparison. Algorithm-II was expected to find a low cosine value which indicates dissimilarity between adjacent sentences for random data in accordance with Landauer et al. (2013) who found that a mean cosine value of 0.08 for random data. Algorithm-II produced mean cosine value of 0.02.

5.3.2 Result of Algorithm-II applied to Deerwester's Data

Deerwester's data is divided into two sections: s1-s5 and s6-s8. Algorithm-II was expected to find sentence dissimilarity between s5 and s6 because they are in different sentence groups. The result of Algorithm-II is listed in Figure-5.14.

```
Doc-Doc matrix:
1 1 1 1 0 0 0 0 0
1 1 2 2 3 0 0 0 1
1 2 1 3 1 0 0 0 0
1 2 3 1 0 0 0 0 0
0 3 1 0 1 0 0 0 0
0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 1 2 1
0 0 0 0 0 1 2 1 2
0 1 0 0 0 0 1 2 1

Doc-Distance matrix:
1     0.5      0.25 0.125 0    0    0    0    0
0.5   1        1    0.5   0.5  0    0    0    0.007813
0.25  1        1    2     0.25 0    0    0    0
0.125 0.5      2    1     0    0    0    0    0
0     0.5      0.25 0     1    0    0    0    0
0     0        0    0     0    1    0.5  0.25 0
0     0        0    0     0    0.5  1    1    0.25
0     0        0    0     0    0.25 1    1    1
0     0.007813 0    0     0    0    0.25 1    1


Reduced U*S matrix (Semantic Space)
s1  0.620981 -0.000343      0 0 0 0 0 0 0
s2  1.526947  0.002005      0 0 0 0 0 0 0
s3  2.332703 -0.004104      0 0 0 0 0 0 0
s4  2.121458 -0.004366      0 0 0 0 0 0 0
s5  0.518816 -1.417964E-005 0 0 0 0 0 0 0
s6  0.000663  0.691471      0 0 0 0 0 0 0
s7  0.001895  1.43476       0 0 0 0 0 0 0
s8  0.003094  1.723234      0 0 0 0 0 0 0
s9  0.00597   1.253802      0 0 0 0 0 0 0

Sentence comparisons:

s1-s2: cos((0.62,-0.00),(1.53,0.00)):       1.00
s2-s3: cos((1.53,0.00),(2.33,-0.00)):       1.00
s3-s4: cos((2.33,-0.00),(2.12,-0.00)):      1.00
s4:s5: cos((2.12,-0.00),(0.52,-0.00)):      1.00
s5-s6: cos((0.52,-0.00),(0.00,0.69)):       0.00
s6-s7: cos((0.00,0.69),(0.00,1.43)):        1.00
s7-s8: cos((0.00,1.43),(0.00,1.72)):        1.00
s8-s9: cos((0.00,1.72),(0.01,1.25)):        1.00
```

Figure 5.14: Result of Algorithm-II applied to Deerwester's Data.

In Figure 5.14, the title 'Sentence comparisons' denotes the cosine values of sentences. The label 's5-s6' denotes the comparison of sentence-5 and sentence-6. The low cosine value indicates dissimilarity and a high cosine value indicates similarity. The cosine value of s5-s6 is 0 and all others have cosine value of 1. Thus, algorithm-II detects paragraph boundaries as expected.

5.3.3 Result of Algorithm-II applied to the Music and Baking data (Landauer et al., 2013)

The Music and Baking Data is divided into two section s1-s5 and s6-s9, inclusive. Algorithm-II was expected to find a high paragraph boundary at s5-s6 because they are in different sentence groups. The result of Algorithm-II is listed in Figure-5.15. In Figure 5.14, the title 'Sentence comparisons' denotes comparison of the cosine values of sentences. The label 's5-s6' denotes the sentence comparison of sentence-5 and sentence-6. Low cosine value indicates dissimilarity and high cosine value indicates similarity. The cosine value of s5-s6 is 0.35 which is the smallest cosine. Thus, algorithm-II detects paragraph boundaries as expected.

```
Reduces U*S matrix (Semantic Space):
9 x 9 matrix
0.119455 -0.848481 0 0 0 0 0 0 0
0.17858  -1.00381  0 0 0 0 0 0 0
0.091824 -0.798316 0 0 0 0 0 0 0
0.049413 -0.659266 0 0 0 0 0 0 0
0.04976  -0.689378 0 0 0 0 0 0 0
0.592484 -0.172605 0 0 0 0 0 0 0
0.580693 -0.090692 0 0 0 0 0 0 0
1.360005  0.206226 0 0 0 0 0 0 0
1.358606  0.217327 0 0 0 0 0 0 0

Sentence comparisons:
s1-s2  : cos((0.12,-0.85),(0.18,-1.00)):        1.00
s2-s3  : cos((0.18,-1.00),(0.09,-0.80)):        1.00
s3-s4  : cos((0.09,-0.80),(0.05,-0.66)):        1.00
s4-s5  : cos((0.05,-0.66),(0.05,-0.69)):        1.00
s5-s6  : cos((0.05,-0.69),(0.59,-0.17)):        0.35
s6-s7  : cos((0.59,-0.17),(0.58,-0.09)):        0.99
s7-s8  : cos((0.58,-0.09),(1.36,0.21)) :        0.95
s8-s9  : cos((1.36,0.21),(1.36,0.22))  :        1.00
```

Figure 5.15: Result of Algorithm-II on Music and Baking Data.

5.3.4 Results of Algorithm-II applied to a chapter of a book

This data is the same data used in Algorithm-I. the data has three paragraph boundaries: sentence-3,sentence-4 and sentence-16 (see Figure C.5). To compare the result of Algorithm-II with classical LSA, the term-doc and doc-doc matrices are used. There are three results: LSA for term-doc matrix, LSA for doc-doc matrix and Algorithm-II on document-distance matrix. The results are given in Figure 5.16.

A bar chart representation of numerical results of Algorithm-II is given in Figure 5.17.

The classical LSA found nine paragraph boundaries (s2, s3, s6, s9, s13, s16, s18, s20, s21). LSA-on-Doc-Doc found 3 paragraph boundaries (s9, s16, s21) and Algorithm-II (LSA-on-Doc-Distance) found 3 paragraph boundaries (s9, s16, s21). The re-

Sentence comparisons	LSA on Term-Doc	LSA on Doc-Doc	LSA on Doc-Distance
s1-s2	0.47	0.99	0.98
s2-s3	-0.0	0.801	0.95
s3-s4	-0.0	0.601	0.84
s4-s5	0.70	0.93	0.50
s5-s6	0.50	0.97	0.85
s6-s7	0.02	0.43	0.98
s7-s8	0.47	0.71	0.90
s8-s9	0.25	0.39	0.91
s9-s10	-0.0	0.340	0.23
s10-s11	0.39	0.97	0.98
s11-s12	0.46	0.98	0.80
s12-s13	0.54	0.87	0.58
s13-s14	0.11	0.68	0.89
s14-s15	0.17	0.71	0.98
s15-s16	0.33	0.98	1.00
s16-s17	0.00	0.23	0.06
s17-s18	0.90	0.99	1.00
s18-s19	0.01	0.33	0.98
s19-s20	0.36	0.36	0.26
s20-s21	0.11	0.85	1.00
s21-s22	-0.0	0.210	0.03
s22-s23	0.21	0.72	0.99
s23-s24	0.55	0.96	1.00
s24-s25	0.41	0.93	1.00
s25-s26	0.52	0.50	1.00

Figure 5.16: Numerical comparisons of results of Algorithm-II.

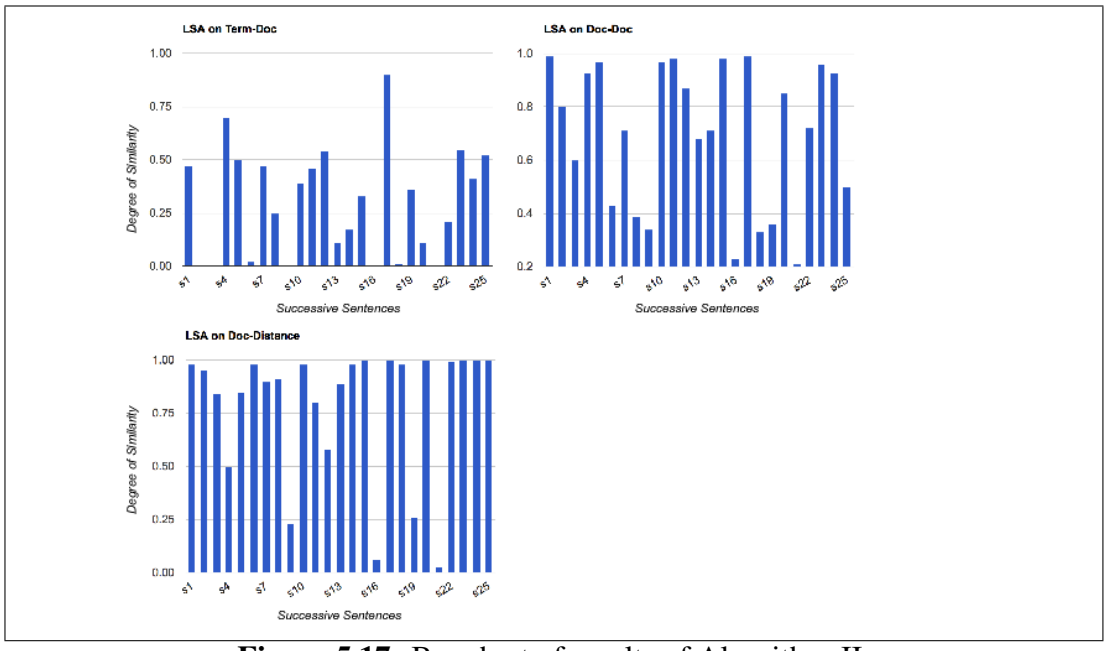


Figure 5.17: Bar chart of results of Algorithm-II.

sult of Algorithm-II and LSA-on-Doc-Doc is the same but Algorithm-II produced less paragraph boundaries with threshold value of 0.25. Although the LSA-on-Doc-Doc has better results that LSA-on-Term-Doc, it still has more paragraph boundaries than Algorithm-II, moreover, Algorithm-II produces sharper indication of paragraph boundaries.

The classical LSA detected nine paragraph boundaries which is not true for the data. LSA-on-Doc-Doc detected 3 paragraph boundaries for threshold value of 0.25 but fails for larger threshold values. Algorithm-II outperformed LSA-on-Doc-Doc method for larger threshold values.

5.3.5 Results of the application of Algorithm-II applied to text from "Introduction to psychology" (Stangor, 2010)

Algorithm-II is applied to seven pairs of paragraphs taken from 'Introduction to psychology' as shown in Appendix C. The data was found to have the following characteristics.

1. Sample-1 a has paragraph boundary between sentence-6 and sentence-7 (s6-s7)
2. Sample-2 a has paragraph boundary between sentence-8 and sentence-9 (s8-s9)
3. Sample-3 a has paragraph boundary between sentence-3 and sentence-4 (s3-s4)
4. Sample-4 a has paragraph boundary between sentence-6 and sentence-7 (s6-s7)
5. Sample-5 a has paragraph boundary between sentence-9 and sentence-10 (s9-s10)
6. Sample-6 a has paragraph boundary between sentence-9 and sentence-10 (s9-s10)
7. Sample-7 a has paragraph boundary between sentence-9 and sentence-10 (s9-s10)

Results are as below.

1. Algorithm-II detects a paragraph boundary at s6-s7 and s3-s4 for sample-1.
2. Algorithm-II fails to detect a paragraph boundary at s2-s3 and s8-s9 for sample-2.
3. Algorithm-II fails to detect a paragraph boundary at s2-s3 and s3-s4 for sample-3.
4. Algorithm-II detects a paragraph boundary at s6-s7 for sample-4.
5. Algorithm-II detects a paragraph boundary at s3-s4 and s9-s10 for sample-5.
6. Algorithm-II detects a paragraph boundary at s9-s10 for sample-6.
7. Algorithm-II detects a paragraph boundary at s9-s10 for sample-7.

Although Algorithm-II detects the intended boundaries with 80% success, it detects more than it needed and the same phenomenon is observed in the result of of the classical LSA. Classical LSA makes comparison of two sentences with the help of word frequencies. Since weighting function and corpus data are not provided in this scenario, the effect of Algorithm-II cannot be seen directly. Therefore, the mean cosine values of Algorithm-II and classical LSA were compared to see the effect of Algorithm-II. It was expected to observe that Algorithm-II produces larger mean cosine values than classical LSA. The mean cosine value comparisons are given in Figure 5.18 which validate the expectation.

Samples	Mean Cosine of Algorithm-II	Mean Cosine of Classical LSA	% of improvement
Sample-1	0.585	0.107	442
Sample-2	0.35	0.189	84
Sample-3	0.78	0.267	200
Sample-4	0.461	0.058	666
Sample-5	0.451	0.099	350
Sample-6	0.199	0.022	850
Sample-7	0.648	0.287	125

Figure 5.18: Mean cosine comparisons of the results of Algorithm-II.

According to Figure 5.18, Algorithm-II produces an improvement on the similarities of successive sentences with compared to the classical LSA. However, this change does not greatly improve the precision of paragraph boundary detection. Classical LSA finds many paragraph boundaries if corpus data is not provided furthermore it becomes more precise if corpus data is provided. In this thesis, corpus data was not provided thus finding a large number of false-true paragraph boundary is to be expected. Algorithm-II was predicted to detect less false-true paragraph boundaries than the classical LSA when no corpus data is provided. According to Figure 5.19, the classical LSA detected 53 paragraph boundaries and Algorithm-II detected 42. The improvement of Algorithm-II is 20%. In addition, Algorithm-II detected 5 intended paragraph boundaries and failed to detect 2 intended paragraph boundaries. However, since Algorithm-II detects false-true paragraph boundaries, the evaluation of the results of Algorithm-II is undertaken by comparing the results from the perspective of reducing the false-true paragraph boundaries. Accordingly, although Algorithm-II detected 8/10 paragraph boundaries, it detected 42 paragraph boundaries, which was not expected. However, the same results are obtained with classical LSA. Therefore, it can be concluded that Algorithm-II has an improvement of 20% over the classical LSA but it still needs to be enhanced to minimize the false-true paragraph boundaries.

5.3.6 Conclusion

Algorithm-II has provided an improvement of 20% on the reduction of the false-true paragraph boundaries detected by classical LSA, and Algorithm-I has 64% success rate in detecting paragraph boundaries. Accordingly, these results demonstrate that applying LSA to the document-distance matrix has an improvement over detection of paragraph boundaries performed by the classical LSA.

5.4 Discussion

This section introduced two algorithms based on the doc-doc matrix. Examination on the doc-doc matrix helped to build several plausible assumptions. According to these assumptions, a defendable hypothesis was introduced which introduced the idea that the spatial distance of re-occurring words between adjacent sentences in the doc-

Samples	Algorithm-II	Classical LSA
Sample-1	5	3
Sample-2	8	9
Sample-3	2	4
Sample-4	6	4
Sample-5	7	8
Sample-6	7	15
Sample-7	7	10
Sum	**42**	**53**

Figure 5.19: The numbers of paragraph boundaries detected by Algorithm-II

doc matrix has an effect on the degree of similarity between adjacent sentences. To test this hypothesis, two algorithms have been introduced and was shown that the hypothesis is defendable.

When developing the algorithms, the work of Kontostathis and Pottenger (2006) was utilized in terms of showing how the term-term matrix can be derived from the document-distance matrix and then used to measure term similarity. The same approach is used to derive the doc-doc matrix from the term-document matrix in the current thesis. Thus, it is demonstrated that the document-distance matrix (derived from the term-document matrix) reveals the spatial distance of re-occurring words in adjacent sentences. In the literature, no study was found concerning the document-distance matrix prior to the research reported in this thesis. Therefore, we may name the algorithm introduced a complementary of the study of Kontostathis and Pottenger (2006). Kontostathis and Pottenger (2006) introduced an approach on U in the equation of $A = USV^T$ whereas in this thesis a new approach is introduced in relation to V in the equation of $A = USV^T$. This thesis shows that the studies on V help to measure sentence similarity which may help to detect paragraph boundaries. Detecting paragraph boundaries with the document-distance matrix is a new method to measure coherence contained within this current work.

CHAPTER 6

CONCLUSION

Coherence is a cognitive phenomenon that happens while reading a text and it indicates how well a text is comprehended by the reader. Comprehending a text consists of successive steps of creating situational models while reading, and ends when the reader constructs a final mental representation of the text. Since mental representation cannot be constructed without understanding the meaning of words being read, comprehending and coherence are also topics of *theory of meaning*. The theory of meaning has two main theoretical frameworks: distributional semantics and compositional semantics. Both frameworks introduce theories to explain the quantification of coherence. However, none of the theories are complementary because of the nature of the intrinsic meaning of words. A word in a text is symbolic, and reader cannot construct a mental representation of text by solely using the dictionary meaning of each word and this is also referred to as the *Symbol grounding problem* (Searle, 1990). Therefore, all attempts to measure coherence can be considered invalid due to philosophical questions concerning meaning. In the relevant literature, there are theories about coherence and mathematical models proposed its quantification but there is no clear definition of the phenomenon of coherence. There are also theories about the relation between coherence and mental representation however, due to the unknown nature of mind coherence remains an obscure concept.

The existing unknowns about meaning and limits concerning the quantification of coherence are also applicable to this thesis in which two algorithms proposed that are based on assumptions of distributional semantics to quantify textual coherence. However, the quantification undertaken using the measurement of the similarities of symbols (words) in the same textbase rather than measuring the coherence of situational model of the reader. This thesis measures how well cohesive cues are linked together. The positive effect of well connected cohesive cues on reader comprehension is generally accepted by comprehension models and this thesis assumes that revealing the well connectedness of cohesive cues indicates how well the coherence of reader's situational model is organized. Although there is limitation, the gap between measuring cohesive cues and measuring coherence is not filled, however, the same limitation is true for the methods of quantification of coherence in the literature. As a result, this thesis inherited the limitation of theoretical frameworks that it is based on. Another problematic is hidden assumption in which a text being measured by thesis algorithms is a result of a coherent mental representation of a writer or it is random data. This dichotomy of true coherent data and random data makes the proposal in this thesis an ad-hoc method for the quantification of coherence, since it does not pro-

vide a clear definition for the input data. This implies that the algorithms proposed will work for some of the data. Despite the lack of true modeling, this thesis does address the effect of the spatial distance of re-occurring words as a cohesive cue and points out its significant effect on the quantification of coherence.

This thesis assumes that sentences sharing same words should be aligned nearby, and states that the spatial distance of re-occurring words in adjacent sentences can be used to quantify coherence. It uses the spatial distance of re-occurring words as cohesive cues. Halliday and Hasan (1976) introduceed spatial distance as cohesion in their inventory of cohesive cues however, they did not introduce spatial distance as lexical cohesion but considered it the *Direction and distance of cohesion* focusing on the distance between sentences. In this thesis, spatial distance is considered as a cohesive cue in the category of lexical cohesion since the distance is measured between re-occurring lexicals. At the time of writing, no study was found that accepts the spatial distance of re-occurring words in adjacent sentences as a cohesive cue. Moreover, there appeared to be no study which proposed a practical method showing that spatial distance of re-occurring words has a significant effect on measuring of coherence. Moreover, the thesis introduces *Distance Closure Function* as a local weighting function which might be used in the conjunction of other local weighting functions. It can be used with other coherence detection methods.

For the algorithms introduced in the thesis, two mathematical functions were utilized as follows: *Error Function* for Algorithm-I and *Distance Closure Function* for Algorithm-II. Although these functions help to reveal the significant effect of distance on coherence, they could be improved in the future work. The *Error Function* summed spatial distance values of elements in the error rectangle. While summing up, it gives a lower error value to elements on the left of the error rectangle and higher error value to elements on the right of the error rectangle. A new Error function could be defined that uses a skewed Gaussian distribution function. Thus, assigning a distributional error to the elements in the error rectangle can be controlled. This approach may regulate the error rectangle and yield better results. An improvement for *Distance Closure Function* that promotes the distant sentence having more shared words could discount the distant sentence that has no shared words and may yield a better result.

To sum up, this thesis work proposes that the spatial distance of re-occurring words in adjacent sentences hold a cohesive cue. This has also addressed at sentence level by Halliday and Hasan (1976). Two mathematical functions are proposed to build a document-distance matrix of re-occurring words in adjacent sentences and two algorithms that operate on the derived matrices to quantify the similarity of successive sentences. The results show that spatial distance between re-occurring words has a significant effect on successive sentence similarity measurement which indicates well connected cohesive cues among sentences.

Bibliography

[1] John Langshaw Austin. “Philosophical papers”. In: (1979).

[2] Marco Baroni and Alessandro Lenci. “Distributional memory: A general framework for corpus-based semantics”. In: *Computational Linguistics* 36.4 (2010), pp. 673–721.

[3] Johan Bos et al. “Wide-coverage semantic representations from a CCG parser”. In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics. 2004, p. 1240.

[4] Freddy YY Choi, Peter Wiemer-Hastings, and Johanna Moore. “Latent semantic analysis for text segmentation”. In: *In Proceedings of EMNLP*. Citeseer. 2001.

[5] N. Chomsky. *Language and Problems of Knowledge: The Managua Lectures*. Current studies in linguistics series. MIT Press, 1988. ISBN: 9780262530705. URL: http://books.google.com.tr/books?id=hwgHVRZtK8kC.

[6] Daoud Clarke. “A context-theoretic framework for compositionality in distributional semantics”. In: *Computational Linguistics* 38.1 (2012), pp. 41–71.

[7] Scott C. Deerwester et al. “Indexing by latent semantic analysis”. In: *JASIS* 41.6 (1990), pp. 391–407.

[8] Katrin Erk. “Towards a semantics for distributional representations”. In: *Proceedings of the Tenth International Conference on Computational Semantics (IWCS2013)*. 2013.

[9] Katrin Erk and Sebastian Padó. “A structured vector space model for word meaning in context”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2008, pp. 897–906.

[10] Katrin Erk and Sebastian Padó. “Exemplar-based models for word meaning in context”. In: *Proceedings of the acl 2010 conference short papers*. Association for Computational Linguistics. 2010, pp. 92–97.

[11] John R. Firth. *Papers in Linguistics, 1934-1951*. Oxford, UK: Oxford University Press, 1957.

[12] Peter W Foltz, Walter Kintsch, and Thomas K Landauer. “The measurement of textual coherence with latent semantic analysis”. In: *Discourse processes* 25.2-3 (1998), pp. 285–307.

[13] Gottlob Frege. *On sense and nominatum, reprinted[1986] in Martinich, A. P. (ed.) The philosophy of language*. Oxford University Press, 1892.

[14] Dan Garrette, Katrin Erk, and Raymond Mooney. “Integrating logical representations with probabilistic information using markov logic”. In: *Proceedings of the Ninth International Conference on Computational Semantics*. Association for Computational Linguistics. 2011, pp. 105–114.

[15] Arthur C Graesser et al. “Coh-Metrix: Analysis of text on cohesion and language”. In: *Behavior Research Methods, Instruments, & Computers* 36.2 (2004), pp. 193–202.

[16] Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. “Centering: A framework for modeling the local coherence of discourse”. In: *Computational linguistics* 21.2 (1995), pp. 203–225.

[17] M.A.K. Halliday and R. Hasan. *Cohesion in English*. English language series. Longman, 1976. URL: `https://encrypted.google.com/books?id=zMBZAAAAMAAJ`.

[18] Stevan Harnad. “The symbol grounding problem”. In: *Physica D: Nonlinear Phenomena* 42.1 (1990), pp. 335–346.

[19] Zellig S Harris. “Distributional structure.” In: *Word* (1954).

[20] Jerry R Hobbs et al. “Interpretation as abduction”. In: *Proceedings of the 26th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics. 1988, pp. 95–103.

[21] Walter Kintsch. *Comprehension: A paradigm for cognition*. Cambridge university press, 1998.

[22] Walter Kintsch. “The role of knowledge in discourse comprehension: a construction-integration model.” In: *Psychological review* 95.2 (1988), p. 163.

[23] Walter Kintsch and Teun A Van Dijk. “Toward a model of text comprehension and production.” In: *Psychological review* 85.5 (1978), p. 363.

[24] April Kontostathis and William M Pottenger. “A framework for understanding Latent Semantic Indexing (LSI) performance”. In: *Information Processing & Management* 42.1 (2006), pp. 56–73.

[25] Thomas K Landauer and Susan T Dumais. “A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.” In: *Psychological review* 104.2 (1997), p. 211.

[26] Thomas K Landauer et al. *Handbook of latent semantic analysis*. Psychology Press, 2013.

[27] Frank Stefan Lennart. *Computational modeling of discourse comprehension*. 2004.

[28] Dekang Lin. "Automatic retrieval and clustering of similar words". In: *Proceedings of the 17th international conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. 1998, pp. 768–774.

[29] Kevin Lund and Curt Burgess. "Producing high-dimensional semantic spaces from lexical co-occurrence". In: *Behavior Research Methods, Instruments, & Computers* 28.2 (1996), pp. 203–208.

[30] Kevin Lund and Curt Burgess. "Producing high-dimensional semantic spaces from lexical co-occurrence". In: *Behavior Research Methods, Instruments, & Computers* 28.2 (1996), pp. 203–208.

[31] Kevin Lund, Curt Burgess, and Ruth Ann Atchley. "Semantic and associative priming in high-dimensional semantic space". In: *Proceedings of the 17th annual conference of the Cognitive Science Society*. Vol. 17. 1995, pp. 660–665.

[32] William G Lycan. *Philosophy of language: A contemporary introduction*. Routledge, 2008.

[33] Donald Davidson Malpas J. *Stanford Encyclopedia of Philosophy(Winter 2012 Edition), Edward N. Zalta (ed.)* The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, 2012.

[34] Danielle S McNamara and Joe Magliano. "Toward a comprehensive model of comprehension". In: *Psychology of learning and motivation* 51 (2009), pp. 297–384.

[35] William Mill and April Kontostathis. *Analysis of the values in the LSI term-term matrix*. 2004.

[36] Jeff Mitchell and Mirella Lapata. "Composition in distributional models of semantics". In: *Cognitive science* 34.8 (2010), pp. 1388–1429.

[37] Sebastian Padó and Mirella Lapata. "Dependency-based construction of semantic space models". In: *Computational Linguistics* 33.2 (2007), pp. 161–199.

[38] Willard Van Orman Quine. *Word and object*. MIT press, 2013.

[39] William V Quine. "On the reasons for indeterminacy of translation". In: *The Journal of Philosophy* (1970), pp. 178–183.

[40] Reinhard Rapp. “Word sense discovery based on sense descriptor dissimilarity”. In: *Proceedings of the Ninth Machine Translation Summit*. 2003, pp. 315–322.

[41] Nick Riemer. *Introducing semantics*. Cambridge University Press, 2010.

[42] Gerard Salton. “The SMART retrieval system –experiments in automatic document processing”. In: (1971).

[43] Gerard Salton, Anita Wong, and Chung-Shu Yang. “A vector space model for automatic indexing”. In: *Communications of the ACM* 18.11 (1975), pp. 613–620.

[44] Franz Schmalhofer, Mark A McDaniel, and Dennis Keefe. “A unified model for predictive and bridging inferences”. In: *Discourse Processes* 33.2 (2002), pp. 105–132.

[45] John R Searle. “Is the brain’s mind a computer program”. In: *Scientific American* 262.1 (1990), pp. 26–31.

[46] Charles Stuart Stanford et al. *Plato’s Apology of Socrates, Crito and Phædo. Translated... by CS Stanford*. W. Curry, Jun., and Company, 1835.

[47] Charles Stangor. “Introduction to psychology”. In: (2010).

[48] Peter F Strawson. “On referring”. In: *Mind* (1950), pp. 320–344.

[49] Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. “Contextualizing semantic representations using syntactically enriched vector models”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 948–957.

[50] William F Trench. *Introduction to real analysis*. Prentice Hall/Pearson Education, 2003.

[51] Peter D Turney. “Domain and function: A dual-space model of semantic relations and compositions”. In: *Journal of Artificial Intelligence Research* (2012), pp. 533–585.

[52] Peter D Turney. “Domain and function: A dual-space model of semantic relations and compositions”. In: *arXiv preprint arXiv:1309.4035* (2013).

[53] Peter D Turney. “Similarity of semantic relations”. In: *Computational Linguistics* 32.3 (2006), pp. 379–416.

[54] Peter D Turney and Patrick Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of artificial intelligence research* 37.1 (2010), pp. 141–188.

[55] Peter D Turney and Patrick Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of artificial intelligence research* 37.1 (2010), pp. 141–188.

[56] Ludwig Wittgenstein. *Philosophical investigations*. John Wiley & Sons, 1953.

[57] Rolf A Zwaan and Gabriel A Radvansky. “Situation models in language comprehension and memory.” In: *Psychological bulletin* 123.2 (1998), p. 162.

APPENDIX A

Appendix A

This chapter gives the detailed explanations of explanations of examples presented in Chapter 1.

A.1 Dutch Learner

Let there be an English girl trying to learn Dutch. Explain to her the meaning of *groot* in the following sentence.

1. Dirk is groot, maar Lou is klein.
   'Dirk is tall, but Lou is short.'

The Dutch learner may be satisfied with the meaning of *groot*. However, a linguist knows that the whole meaning of *groot* is not the English translation of it, it is obvious that the world context of the word when uttering it is totally omitted. It is a word-to-word translation, not a translation of the whole meaning (Lycan, 2008).

Let's go further and try to explain the meaning of "humorous" using only English words as int the conversation below.

G what is humorous?

T It means droll.

G what is droll?

T amusing.

G what is amusing?

T funny.

Eventually, you will run out the distinct words to help you and you will end up repeating a word. Let's assume that there are infinite number of distinct words to explain the meaning of the "humorous". Although you use as many words as you want, it does

not mean that you convey the exact meaning constructed by the word ”humorous” in mind to the English girl. This problem was firstly addressed by Quine (1961:47). Actually, when we generalize the problem, we can ask the following question:

1. Is there a gap in the constructed meaning comparing the English and Dutch meaning of the word 'humorous'?
2. If we had an infinite number of distinct words, would we be able to convey the exact meaning?

There is one more question which was already addressed by Stanford et al. (427 B.C. - 347 B.C.). How do we know something although it is not taught? This question was named as *Plato's Problem* by Chomsky(1988).

A.2 Frege's Reference and Sense

1. a. The morning star is the morning star.
2. b. The morning star is the evening star.

According to Frege, the sentences (a) and (b) have the same form of the statement: a=b and (a) is in a tautology and (b) is informative. If we have the assumption that the meaning is simply reference, there should be no difference between the sentences (a) and (b). However, it is obvious that they are not the same. Since the 'a' form of statement has the same referent to an entity in the world, reference-entity mapping has to be broken in the sentences to preserve the difference between these two sentence. As a result, he differentiated reference from entity and introduced the abstraction of the 'reference and sense' (Sinn und Bedeutung).

Although it seems that 'senses' can be considered like individual ideas or mental images as in Aristotle and Locke, the senses of expression are part of thought but are not subjective entities which vary from one to another(Riemer, 2010). They are conceptually the referent of a word or a sentence, where this referent is a sort of abstraction which may remain for ever like a proposition. In addition, there is no contextual information in Frege's Theory of meaning. It is sentence bounded and each word in a sentence has to denote a 'thing' and the composition of the statement will denote the truth value of the sentence. This is why the theory is sometimes called interchangeably as denotational/compositional/propositional semantics. In 1953, this theory was to be challenged by Bertrand Russell (1905/1956, 1918/1956, 1919/1971) (Malpas, 2012; Riemer, 2010).

A.3 Russell's example for the theory of descriptions

1. At least one woman lives there
2. at most one woman lives there

3. whoever lives there is a biochemist

The example above was given against the singular terms denoted in the Referential Theory. Although it was known that Referential Theory is not applicable to all entities, it may work for singular terms such as proper names; eg., John, the woman. Russell powerfully showed that a definite noun phrase may refer to more than one proposition which breaks the essential assumption of the Referential Theory. Here is the contextual definition of "The" in the following sentence(Riemer, 2010).

1. The present King of France is bald.

(a) at least one person is presently King of France, and (b) at most one person is presently King of France, and (c) whoever is presently King of France is Bald

Note: W indicates the predicate. B means Bald.

1.

$$(\exists x)Wx \tag{A.1}$$

2.

$$(x)(Wx \rightarrow (y)(Wy \rightarrow y = x)) \tag{A.2}$$

3.

$$(x)(Wx \rightarrow Bx) \tag{A.3}$$

The three proposition given above are conjointly equivalent to

1.

$$(\exists x)(Wx\&((y)(Wy \rightarrow y = x\&Bx)) \tag{A.4}$$

Since compositional semantics of three propositions given above exactly indicate the truth value of the sentence, it is obvious that "the definite noun phrase may not mean what they mean in virtue of denoting what they denote" (Lycan, 2008). Russell did not only challenge the singular terms argumentation of the Referential Theory but he also argued that his analysis could also be applied to four logical puzzles; namely, the Problems of Apparent Reference to Nonexistents and Negative Existentials, Frege's Puzzle about Identity and Subsitutivity.

APPENDIX B

Appendix B

This chapter presents how to build an LSA based semantic space.

B.1 Creating Your Own LSA Space

[1] To have an LSA space, you need the following.

1. Utilities that parse text.
2. Libraries that perform LSA computing.
3. Vector manipulation utilities.

B.1.1 Parsing utilities for LSA

You can use any language which supports *RegExp*. Here are some examples.

1. Python
2. Perl
3. Java
4. C++

If you are not developing a commercial product but a Proof of Concept (PoC), perl & Python can be a choice. Perl language has a special focus on text parsing. You may find several libraries for parsing in Perl.

B.1.1.1 Parsing

Parsing is breaking a string into its tokenized lists. When parsing, you may have the following concerns.

[1] This section is a summary of Chapter 4 in (Landauer et al., 2013).

1. The minimum length of acceptable token
2. Whether accept punctuation of not?
3. Whether accept stop words or not?
4. Whether keep numbers?
5. Determining the boundary of a word and whether to remove the derivational affixes?
6. Syntagmatic concerns

Here are some tools for parsing:

1. *mkey* tool of Telcordia which was originally used to develop LSI [2]
2. General Text Parser (GTP). This can be considered a reference program for LSA since it is a rewritten version of the older Telcordia Suite.
3. Text to Matrix Generator of Matlab(TMG). TMG is a Matlab toolbox that was designed to mimic parsing that is standard in the informational retrieval conferences such as Text Retrieval Conference (TREC)
4. R programming language, which has a textual data analysis library designed for corpus processing.

B.1.2 Computing SVD

There are two ways to have an SVD utility: your own or using already tested software. You may need a customized SVD if you are building performance proved and scalable SVD software. However, if you need a Proof of Concept, it is best to use an already tested SVD library. Here are some:

1. Matlab
2. Colt (java library used by CERN)
3. Mathematica
4. R language
5. GTP/pindex
6. Pyhton pysci

[2] LSI denotes the Latent Semantic Indexing. LSI deals with document querying and informational retrieval whereas LSA deals with word similarity.

B.1.3 Operating with Vectors

Once you have the result from SVD, you need to operate mostly on U (type vectors) and V (document vectors). You can save all vectors in memory or disk. If you need an operation for a small subset of the vectors, writing the vectors to the disk and retrieving them upon request may be the best choice.

Matlab, Mathematica and R are all matrix oriented programming languages. Once you have the result from the SVD, you have all vectors in memory. Operating on vectors is easier using one of these languages.

APPENDIX C

Appendix C

This chapter presents the input data and the detailed results of algorithms introduced in this thesis.

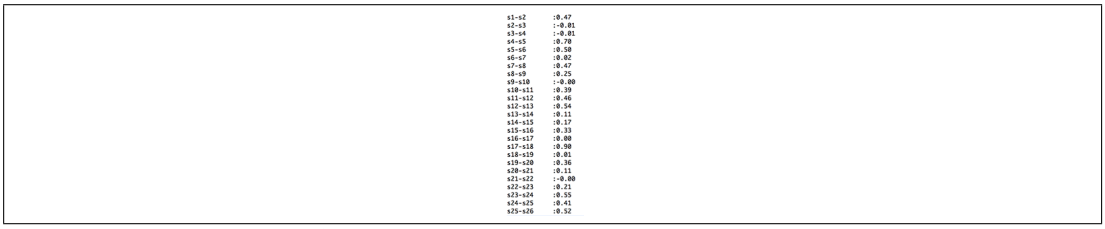


Figure C.1: The result of Classic LSA on Real Data.

```
Reduced U*S matrix (Semantic Space):
 6.153426 -1.027132  0.142401 -0.881377 -0.05621   0.140673
 4.228367 -0.961557 -0.033121 -0.217849  0.013916 -0.074268
 0.214496 -0.107124  0.014614  0.005423  0.102734 -0.134567
12.377756 -0.411054 -0.088244  0.927175 -3.902528  0.140663
10.633888  0.302498  0.614935  0.442115  0.420438  0.924597
 7.204969 -1.225078 -0.127302  0.035656 -0.71082   0.816009
 1.741164  1.873436 -0.023065  2.114012  0.001483  0.433418
 8.732316  1.913445  0.179551  3.011049  2.854231 -2.336106
 0.734428  0.227509 -0.026609  0.921321 -0.722306  0.635922
10.561555 -0.333748 -0.161593 -1.025547  0.158136 -1.647101
 6.687592 -1.518861 -0.075581 -0.774698  0.323716 -0.210296
 9.916661 -1.443742  0.746812 -1.45642  -0.807443 -1.645623
10.720532 -0.759352 -0.58613  -0.215077  1.770212  2.876712
 5.52438   4.049825 -3.236476 -0.284548 -0.592222  0.635364
 7.647656 -0.354087 -0.517639 -1.121001  0.641183  0.020658
 6.687592 -1.518861 -0.075581 -0.774698  0.323716 -0.210296
 2.01377   3.070049 -0.825432  2.344739  0.205974  0.043356
 1.908103  2.613531 -0.339199  2.381702  0.07529   0.255004
 0.740937  5.502384 -4.529286 -2.911279  0.026243 -0.659203
 3.760702  6.133816  3.263194 -0.316966 -0.642836  0.860127
 1.01412   5.376086  3.856569 -2.926173  0.505434 -0.050543
 4.802874  1.839166  0.453047  2.383465 -1.282838 -0.815125
 9.314572 -1.187238  0.522161 -1.030374  0.273397 -0.586777
 9.106581 -1.0804   -0.109819  0.143803  1.616675  1.146168
 4.802396 -0.861357  0.022615  0.216045 -0.715139 -0.265754
 0.402936  0.091997 -0.023208  0.490807 -0.309162  0.42907

Sentence Comparisons:
s1-s2       : cos((6.15,-1.03),(4.23,-0.96))    :       0.99
s2-s3       : cos((4.23,-0.96),(0.21,-0.11))    :       0.80
s3-s4       : cos((0.21,-0.11),(12.38,-0.41))   :       0.60
s4-s5       : cos((12.38,-0.41),(10.63,0.30))   :       0.93
s5-s6       : cos((10.63,0.30),(7.20,-1.23))    :       0.97
s6-s7       : cos((7.20,-1.23),(1.74,1.87))     :       0.43
s7-s8       : cos((1.74,1.87),(8.73,1.91))      :       0.71
s8-s9       : cos((8.73,1.91),(0.73,0.23))      :       0.39
s9-s10      : cos((0.73,0.23),(10.56,-0.33))    :       0.34
s10-s11     : cos((10.56,-0.33),(6.69,-1.52))   :       0.97
s11-s12     : cos((6.69,-1.52),(9.92,-1.44))    :       0.98
s12-s13     : cos((9.92,-1.44),(10.72,-0.76))   :       0.87
s13-s14     : cos((10.72,-0.76),(5.52,4.05))    :       0.68
s14-s15     : cos((5.52,4.05),(7.65,-0.35))     :       0.71
s15-s16     : cos((7.65,-0.35),(6.69,-1.52))    :       0.98
s16-s17     : cos((6.69,-1.52),(2.01,3.07))     :       0.23
s17-s18     : cos((2.01,3.07),(1.91,2.61))      :       0.99
s18-s19     : cos((1.91,2.61),(0.74,5.50))      :       0.33
s19-s20     : cos((0.74,5.50),(3.76,6.13))      :       0.36
s20-s21     : cos((3.76,6.13),(1.01,5.38))      :       0.85
s21-s22     : cos((1.01,5.38),(4.80,1.84))      :       0.21
s22-s23     : cos((4.80,1.84),(9.31,-1.19))     :       0.72
s23-s24     : cos((9.31,-1.19),(9.11,-1.08))    :       0.96
s24-s25     : cos((9.11,-1.08),(4.80,-0.86))    :       0.93
s25-s26     : cos((4.80,-0.86),(0.40,0.09))     :       0.50
```

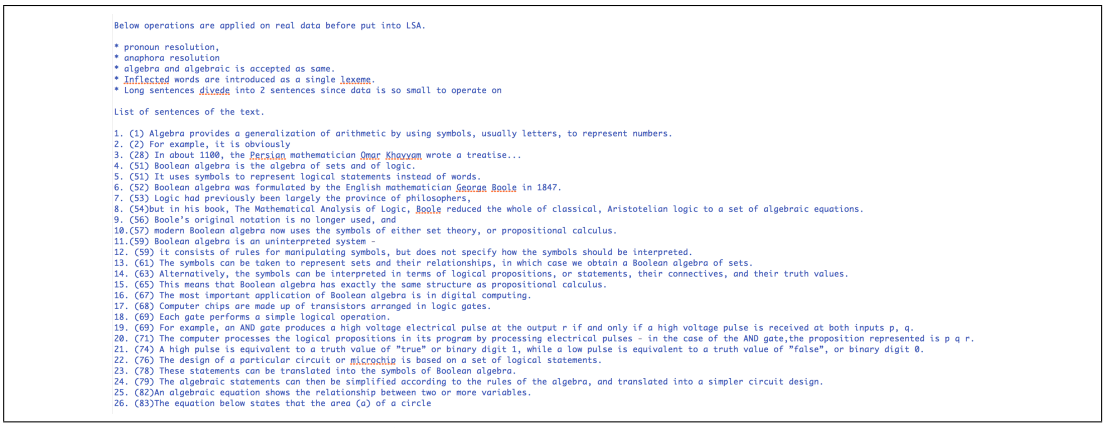
Figure C.2: The result of Algorithm-II on the doc-doc matrix of real data.

```
Reduced U*S matrix (Semantic Space):
2.419926E-005  9.683189E-008  0.077116 -0.345232  0.000282      0.037237      -0.023975
3.424971E-005  7.543801E-008  0.117426 -0.528005  0.00041       0.163735      -0.108552
3.614421E-006  5.232358E-008  0.011715 -0.045951  8.373729E-006  0.001621      -0.00161
0.00047       -6.195408E-006  0.970442 -3.664795  0.001655     -1.818966       1.185952
0.00049        1.779047E-005  0.940599 -3.553063  0.001685      1.778523      -1.144092
0.000342      -4.536079E-006  0.567353 -1.798336  0.000183     -0.052906       0.051868
0.000283       4.360582E-006  0.193727 -0.601025 -0.00011      -0.119388       0.092313
0.001313       5.790389E-005  0.809945 -1.595379 -0.001132      0.269892      -0.234055
0.000114      -5.291037E-006  0.145511 -0.372075 -0.000406     -0.064516       0.056521
0.007175      -0.000102       3.12149   0.416881 -0.00938      -0.062828      -0.017994
0.002535       3.591778E-005  1.58003   0.260014 -0.005506      0.15799        0.237933
0.010363       0.00336        4.097517  0.908501 -0.00851      -1.198546      -1.782045
0.014213      -0.001186       4.045357  0.90553  -0.00601       1.194394       1.76512
0.07695       -0.050797       0.809541  0.199433  0.011373      0.006192       0.019811
0.039275      -0.025703       0.802821  0.203691  0.019975     -0.069838      -0.109306
0.00378        0.002349       0.417775  0.113689  0.026428     -0.021511      -0.030488
0.304485      -0.202805       0.019435  0.004085  0.044126     -0.000373      -0.000258
0.304457      -0.202777       0.012027  0.002671  0.054913     -4.864578E-005  1.17266E-005
8.864455      -7.100098      -0.016988 -0.003555 -0.024648     -0.000391      -0.000645
6.57237        4.820603       0.001062  0.000907  0.046491     -0.000625      -0.000865
6.545674       4.797015      -0.013917 -0.003081 -0.054907      0.000825       0.00119
0.142762      -0.110468       0.006344  0.002049  1.310691     -0.001238      -0.001848
0.008423       0.003367       0.01152   0.00377   2.162542      0.000178       0.000289
0.013619       0.009674       0.007903  0.002909  2.482502      0.00036        0.000536
0.001335      -0.000971       0.002506  0.000995  1.145123     -0.000185      -0.000277
5.639021E-005  4.478115E-005  0.000207  0.0001    0.209607      2.237494E-005  3.459395E-005

Sentence Comparisons:
s1-s2       : cos((0.00,0.00),(0.00,0.00))  :       0.98
s2-s3       : cos((0.00,0.00),(0.00,0.00))  :       0.95
s3-s4       : cos((0.00,0.00),(0.00,-0.00)) :       0.84
s4-s5       : cos((0.00,-0.00),(0.00,0.00)) :       0.50
s5-s6       : cos((0.00,0.00),(0.00,-0.00)) :       0.85
s6-s7       : cos((0.00,-0.00),(0.00,0.00)) :       0.98
s7-s8       : cos((0.00,0.00),(0.00,0.00))  :       0.90
s8-s9       : cos((0.00,0.00),(0.00,-0.00)) :       0.91
s9-s10      : cos((0.00,-0.00),(0.01,-0.00)):       0.23
s10-s11     : cos((0.01,-0.00),(0.00,0.00)) :       0.98
s11-s12     : cos((0.00,0.00),(0.01,0.00))  :       0.80
s12-s13     : cos((0.01,0.00),(0.01,-0.00)) :       0.58
s13-s14     : cos((0.01,-0.00),(0.08,-0.05)):       0.89
s14-s15     : cos((0.08,-0.05),(0.04,-0.03)):       0.98
s15-s16     : cos((0.04,-0.03),(0.00,0.00)) :       1.00
s16-s17     : cos((0.00,0.00),(0.30,-0.20)) :       0.06
s17-s18     : cos((0.30,-0.20),(0.30,-0.20)):       1.00
s18-s19     : cos((0.30,-0.20),(8.86,-7.10)):       0.98
s19-s20     : cos((8.86,-7.10),(6.57,4.82)) :       0.26
s20-s21     : cos((6.57,4.82),(6.55,4.80))  :       1.00
s21-s22     : cos((6.55,4.80),(0.14,-0.11)) :       0.03
s22-s23     : cos((0.14,-0.11),(0.01,0.00)) :       0.99
s23-s24     : cos((0.01,0.00),(0.01,0.01))  :       1.00
s24-s25     : cos((0.01,0.01),(0.00,-0.00)) :       1.00
s25-s26     : cos((0.00,-0.00),(0.00,0.00)) :       1.00
```

Figure C.3: The result of Algorithm-II on the document-distance matrix of real data.

```
Example Text: “The history of algebra”

1. Algebra provides a generalization of arithmetic by using symbols,
2 usually letters, to represent numbers. For example, it is obviously
...
28 In about 1100, the Persian mathematician Omar Khayyam wrote a treatise...
...
51 Boolean algebra is the algebra of sets and of logic. It uses symbols
52 to represent logical statements instead of words. Boolean algebra was
53 formulated by the English mathematician George Boole in 1847. Logic
54 had previously been largely the province of philosophers, but in his
55 book, The Mathematical Analysis of Logic, Boole reduced the whole of
56 classical, Aristotelian logic to a set of algebraic equations. Boole’s
57 original notation is no longer used, and modern Boolean algebra now
58 uses the symbols of either set theory, or propositional calculus.
59 Boolean algebra is an uninterpreted system - it consists of rules for
60 manipulating symbols, but does not specify how the symbols should be
61 interpreted. The symbols can be taken to represent sets and their
62 relationships, in which case we obtain a Boolean algebra of
63 sets. Alternatively, the symbols can be interpreted in terms of
64 logical propositions, or statements, their connectives, and their
65 truth values. This means that Boolean algebra has exactlythe same
66 structure as propositional calculus.
67 The most important application of Boolean algebra is in digital
68 computing. Computer chips are made up of transistors arranged in logic
69 gates. Each gate performs a simple logical operation. Forexample, an
70 AND gate produces a high voltage electrical pulse at the output r if
71 and only if a high voltage pulse is received at both inputs p, q. The
72 computer processes the logical propositions in its program by
73 processing electrical pulses - in the case of the AND gate,the
74 proposition represented is p q r. A high pulse is equivalent to a
75 truth value of ”true” or binary digit 1, while a low pulse is
76 equivalent to a truth value of ”false”, or binary digit 0. The design
77 of a particular circuit or microchip is based on a set of logical
78 statements. These statements can be translated into the symbols of
79 Boolean algebra. The algebraic statements can then be simplified
80 according to the rules of the algebra, and translated intoa simpler
81 circuit design.
...
82 An algebraic equation shows the relationship between two or more
83 variables. The equation below states that the area (a) of a circle
```

Figure C.4: Real Data with 2 paragraphs.

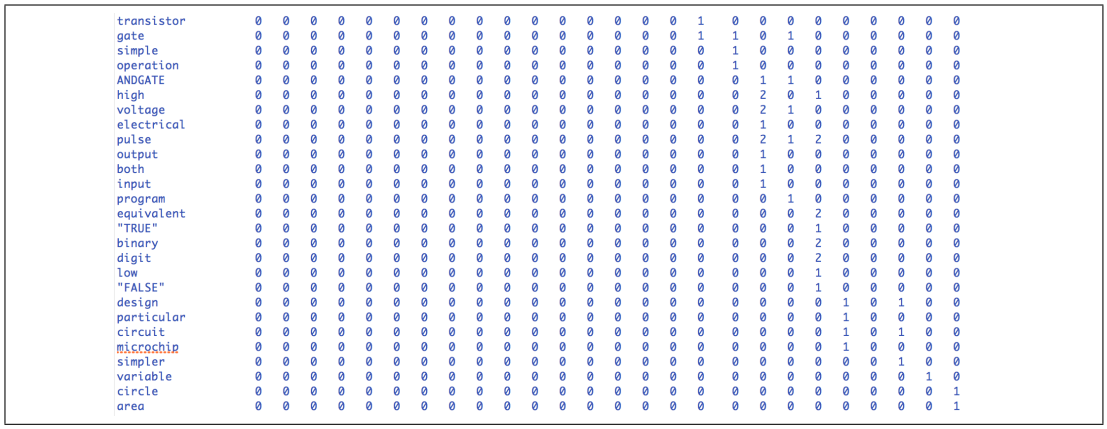


```
Below operations are applied on real data before put into LSA.

* pronoun resolution,
* anaphora resolution
* algebra and algebraic is accepted as same.
* Inflected words are introduced as a single lexeme.
* Long sentences divede into 2 sentences since data is so small to operate on

List of sentences of the text.

1. (1) Algebra provides a generalization of arithmetic by using symbols, usually letters, to represent numbers.
2. (2) For example, it is obviously
3. (28) In about 1100, the Persian mathematician Omar Khayyam wrote a treatise...
4. (51) Boolean algebra is the algebra of sets and of logic.
5. (51) It uses symbols to represent logical statements instead of words.
6. (52) Boolean algebra was formulated by the English mathematician George Boole in 1847.
7. (53) Logic had previously been largely the province of philosophers,
8. (54)but in his book, The Mathematical Analysis of Logic, Boole reduced the whole of classical, Aristotelian logic to a set of algebraic equations.
9. (56) Boole's original notation is no longer used, and
10.(57) modern Boolean algebra now uses the symbols of either set theory, or propositional calculus.
11.(59) Boolean algebra is an uninterpreted system -
12. (59) it consists of rules for manipulating symbols, but does not specify how the symbols should be interpreted.
13. (61) The symbols can be taken to represent sets and their relationships, in which case we obtain a Boolean algebra of sets.
14. (63) Alternatively, the symbols can be interpreted in terms of logical propositions, or statements, their connectives, and their truth values.
15. (65) This means that Boolean algebra has exactly the same structure as propositional calculus.
16. (67) The most important application of Boolean algebra is in digital computing.
17. (68) Computer chips are made up of transistors arranged in logic gates.
18. (69) Each gate performs a simple logical operation.
19. (69) For example, an AND gate produces a high voltage electrical pulse at the output r if and only if a high voltage pulse is received at both inputs p, q.
20. (71) The computer processes the logical propositions in its program by processing electrical pulses - in the case of the AND gate,the proposition represented is p q r.
21. (74) A high pulse is equivalent to a truth value of "true" or binary digit 1, while a low pulse is equivalent to a truth value of "false", or binary digit 0.
22. (76) The design of a particular circuit or microchip is based on a set of logical statements.
23. (78) These statements can be translated into the symbols of Boolean algebra.
24. (79) The algebraic statements can then be simplified according to the rules of the algebra, and translated into a simpler circuit design.
25. (82)An algebraic equation shows the relationship between two or more variables.
26. (83)The equation below states that the area (a) of a circle
```

Figure C.5: Real Data with 2 paragraphs.

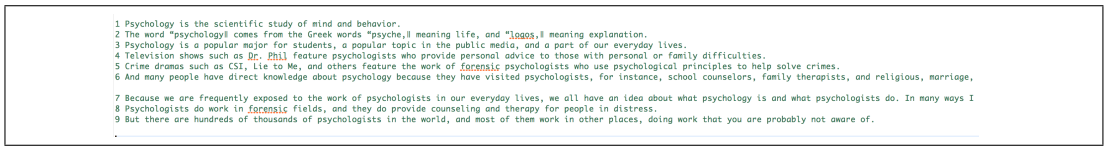
	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21	s22	s23	s24	s25	s26
boolean	0	0	0	1	1	1	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0
algebra	1	1	0	2	1	1	0	1	0	1	1	1	1	0	1	1	0	0	0	0	0	0	1	2	1	0
generalization	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
arithmetic	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
symbol	1	0	0	0	1	0	0	0	0	1	0	2	1	1	0	0	0	0	0	0	0	0	1	0	0	0
letter	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
number	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
persian	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathematician	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
khayyam	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
treatise	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
set	0	0	0	1	0	0	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0
logic	0	0	0	1	1	0	1	2	0	0	0	0	0	1	0	0	1	1	0	1	0	1	0	0	0	0
statement	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0
word	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
english	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
georgeboole	0	0	0	0	0	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
province	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
philosopher	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
book	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mathematical	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
analysis	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
whole	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
classical	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
aristo	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
equation	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
original	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
notation	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
longer	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
modern	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
theory	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
propositional	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	2	0	0	0	0	0	0
calculus	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
interpreting	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
system	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rule	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
relationship	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
case	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
term	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
connective	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
truth	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0
value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0
same	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
structure	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
most	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
important	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
application	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
digital	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
computing	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
computer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
chip	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Figure C.6: Term-Doc matrix of Real Data with 2 paragraphs.



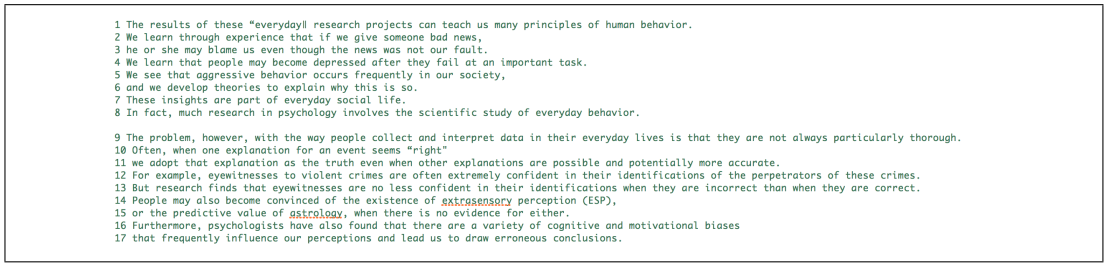
transistor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
gate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
simple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
operation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
ANDGATE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
high	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0
voltage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
electrical	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
pulse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2	0	0	0	0	0
output	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
both	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
input	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
program	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
equivalent	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
"TRUE"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
binary	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
digit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
low	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
"FALSE"	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
design	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
particular	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
circuit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
microchip	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
simpler	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
variable	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
circle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
area	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure C.7: Term-Doc matrix of Real Data with 2 paragraphs (cont.).



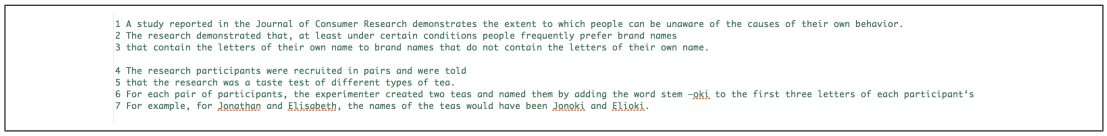
```
1 Psychology is the scientific study of mind and behavior.
2 The word "psychology‖ comes from the Greek words "psyche,‖ meaning life, and "logos,‖ meaning explanation.
3 Psychology is a popular major for students, a popular topic in the public media, and a part of our everyday lives.
4 Television shows such as Dr. Phil feature psychologists who provide personal advice to those with personal or family difficulties.
5 Crime dramas such as CSI, Lie to Me, and others feature the work of forensic psychologists who use psychological principles to help solve crimes.
6 And many people have direct knowledge about psychology because they have visited psychologists, for instance, school counselors, family therapists, and religious, marriage,

7 Because we are frequently exposed to the work of psychologists in our everyday lives, we all have an idea about what psychology is and what psychologists do. In many ways I
8 Psychologists do work in forensic fields, and they do provide counseling and therapy for people in distress.
9 But there are hundreds of thousands of psychologists in the world, and most of them work in other places, doing work that you are probably not aware of.
.
```

Figure C.8: Sample-1 of a Psychology book



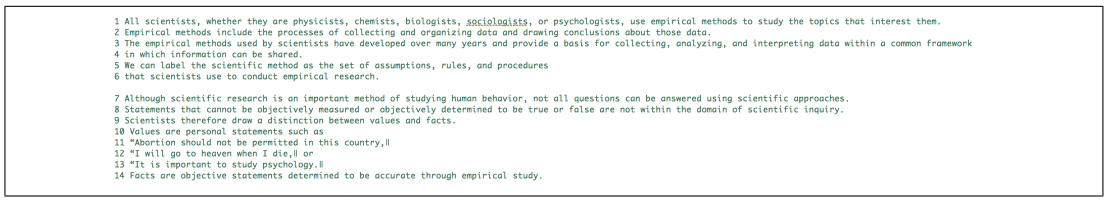
```
1 The results of these "everyday‖ research projects can teach us many principles of human behavior.
2 We learn through experience that if we give someone bad news,
3 he or she may blame us even though the news was not our fault.
4 We learn that people may become depressed after they fail at an important task.
5 We see that aggressive behavior occurs frequently in our society,
6 and we develop theories to explain why this is so.
7 These insights are part of everyday social life.
8 In fact, much research in psychology involves the scientific study of everyday behavior.

9 The problem, however, with the way people collect and interpret data in their everyday lives is that they are not always particularly thorough.
10 Often, when one explanation for an event seems "right"
11 we adopt that explanation as the truth even when other explanations are possible and potentially more accurate.
12 For example, eyewitnesses to violent crimes are often extremely confident in their identifications of the perpetrators of these crimes.
13 But research finds that eyewitnesses are no less confident in their identifications when they are incorrect than when they are correct.
14 People may also become convinced of the existence of extrasensory perception (ESP),
15 or the predictive value of astrology, when there is no evidence for either.
16 Furthermore, psychologists have also found that there are a variety of cognitive and motivational biases
17 that frequently influence our perceptions and lead us to draw erroneous conclusions.
```

Figure C.9: Sample-2 of a Psychology book



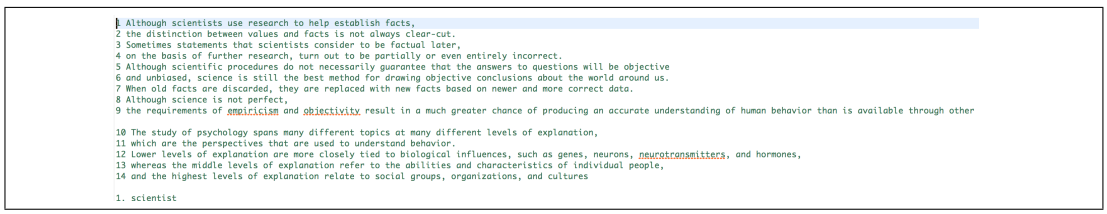
```
1 A study reported in the Journal of Consumer Research demonstrates the extent to which people can be unaware of the causes of their own behavior.
2 The research demonstrated that, at least under certain conditions people frequently prefer brand names
3 that contain the letters of their own name to brand names that do not contain the letters of their own name.

4 The research participants were recruited in pairs and were told
5 that the research was a taste test of different types of tea.
6 For each pair of participants, the experimenter created two teas and named them by adding the word stem -oki to the first three letters of each participant's
7 For example, for Jonathan and Elisabeth, the names of the teas would have been Jonoki and Elioki.
```

Figure C.10: Sample-3 of a Psychology book



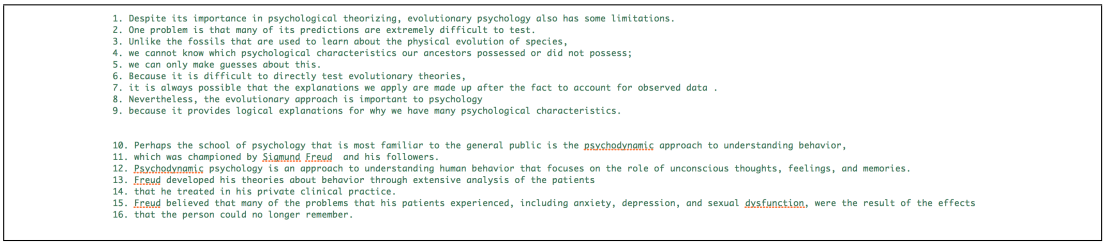
```
1 All scientists, whether they are physicists, chemists, biologists, sociologists, or psychologists, use empirical methods to study the topics that interest them.
2 Empirical methods include the processes of collecting and organizing data and drawing conclusions about those data.
3 The empirical methods used by scientists have developed over many years and provide a basis for collecting, analyzing, and interpreting data within a common framework
4 in which information can be shared.
5 We can label the scientific method as the set of assumptions, rules, and procedures
6 that scientists use to conduct empirical research.

7 Although scientific research is an important method of studying human behavior, not all questions can be answered using scientific approaches.
8 Statements that cannot be objectively measured or objectively determined to be true or false are not within the domain of scientific inquiry.
9 Scientists therefore draw a distinction between values and facts.
10 Values are personal statements such as
11 "Abortion should not be permitted in this country,‖
12 "I will go to heaven when I die,‖ or
13 "It is important to study psychology.‖
14 Facts are objective statements determined to be accurate through empirical study.
```

Figure C.11: Sample-4 of a Psychology book



```
1 Although scientists use research to help establish facts,
2 the distinction between values and facts is not always clear-cut.
3 Sometimes statements that scientists consider to be factual later,
4 on the basis of further research, turn out to be partially or even entirely incorrect.
5 Although scientific procedures do not necessarily guarantee that the answers to questions will be objective
6 and unbiased, science is still the best method for drawing objective conclusions about the world around us.
7 When old facts are discarded, they are replaced with new facts based on newer and more correct data.
8 Although science is not perfect,
9 the requirements of empiricism and objectivity result in a much greater chance of producing an accurate understanding of human behavior than is available through other

10 The study of psychology spans many different topics at many different levels of explanation,
11 which are the perspectives that are used to understand behavior.
12 Lower levels of explanation are more closely tied to biological influences, such as genes, neurons, neurotransmitters, and hormones,
13 whereas the middle levels of explanation refer to the abilities and characteristics of individual people,
14 and the highest levels of explanation relate to social groups, organizations, and cultures

1. scientist
```

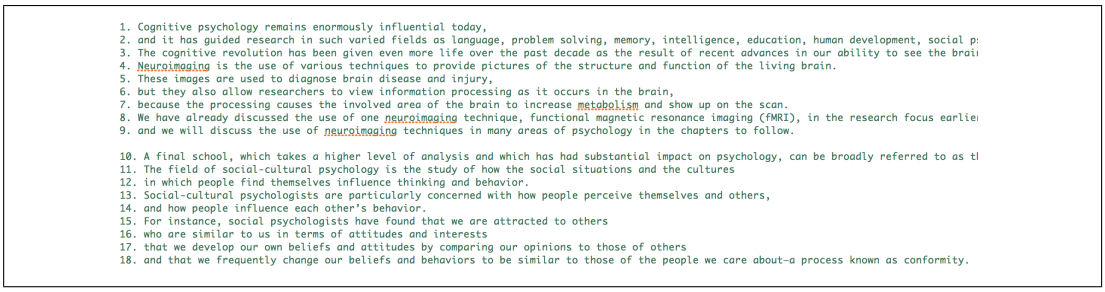
Figure C.12: Sample-5 of a Psychology book



1. Despite its importance in psychological theorizing, evolutionary psychology also has some limitations.
2. One problem is that many of its predictions are extremely difficult to test.
3. Unlike the fossils that are used to learn about the physical evolution of species,
4. we cannot know which psychological characteristics our ancestors possessed or did not possess;
5. we can only make guesses about this.
6. Because it is difficult to directly test evolutionary theories,
7. it is always possible that the explanations we apply are made up after the fact to account for observed data .
8. Nevertheless, the evolutionary approach is important to psychology
9. because it provides logical explanations for why we have many psychological characteristics.

10. Perhaps the school of psychology that is most familiar to the general public is the psychodynamic approach to understanding behavior,
11. which was championed by Sigmund Freud and his followers.
12. Psychodynamic psychology is an approach to understanding human behavior that focuses on the role of unconscious thoughts, feelings, and memories.
13. Freud developed his theories about behavior through extensive analysis of the patients
14. that he treated in his private clinical practice.
15. Freud believed that many of the problems that his patients experienced, including anxiety, depression, and sexual dysfunction, were the result of the effects
16. that the person could no longer remember.

Figure C.13: Sample-6 of a Psychology book



1. Cognitive psychology remains enormously influential today,
2. and it has guided research in such varied fields as language, problem solving, memory, intelligence, education, human development, social p
3. The cognitive revolution has been given even more life over the past decade as the result of recent advances in our ability to see the brai
4. Neuroimaging is the use of various techniques to provide pictures of the structure and function of the living brain.
5. These images are used to diagnose brain disease and injury,
6. but they also allow researchers to view information processing as it occurs in the brain,
7. because the processing causes the involved area of the brain to increase metabolism and show up on the scan.
8. We have already discussed the use of one neuroimaging technique, functional magnetic resonance imaging (FMRI), in the research focus earlie
9. and we will discuss the use of neuroimaging techniques in many areas of psychology in the chapters to follow.

10. A final school, which takes a higher level of analysis and which has had substantial impact on psychology, can be broadly referred to as t
11. The field of social-cultural psychology is the study of how the social situations and the cultures
12. in which people find themselves influence thinking and behavior.
13. Social-cultural psychologists are particularly concerned with how people perceive themselves and others,
14. and how people influence each other's behavior.
15. For instance, social psychologists have found that we are attracted to others
16. who are similar to us in terms of attitudes and interests
17. that we develop our own beliefs and attitudes by comparing our opinions to those of others
18. and that we frequently change our beliefs and behaviors to be similar to those of the people we care about–a process known as conformity.

Figure C.14: Sample-7 of a Psychology book

		Model						
Category	Dimension	Construction–integration	Structure building	Resonance	Constructionist	Event indexing	Causal network	Landscape
Features	Grammatical morphology and syntax	1	2	0	0	1	0	0
	Referential cohesion	3	3	3	1	−1	−1	2
	Situational semantics	2	0	0	1	2	0	0
	Situational cohesion	1	2	0	3	3	3	3
Processes	Dynamic activation	2	2	3	1	0	3	3
	Integration/settling	3	0	1	1	0	2	1
	Memory-based retrieval	2	0	3	1	1	0	2
	Knowledge-based inferencing	3	1	2	3	1	0	2
	Dumb activation	3	2	3	0	0	0	2
	Continuity monitoring	1	0	1	3	3	3	1
	Laying a foundation	0	3	0	0	0	0	0
	Shifting	0	3	0	0	3	0	0
	Suppression	2	3	1	0	0	0	1
Products	Situation model	3	1	0	2	3	2	2
	Propositional textbase	3	1	0	1	1	0	1
	Levels of representation	3	0	0	1	1	0	0

Figure C.15: Evaluation of comprehension models 1

		Model						
Category	Dimension	Construction–integration	Structure building	Resonance	Constructionist	Event indexing	Causal network	Landscape
	Local coherence	3	0	2	3	0	1	1
	Global coherence	3	0	2	3	0	1	1
	Sources of information	2	0	1	1	0	0	3
	Levels of comprehension	3	1	0	2	0	0	1
	Hierarchically structured representation	3	1	0	2	0	3	1
Extratextual	Goals	1	0	0	3	0	0	2
	Task	0	0	0	1	0	0	0
	Affect	1	0	0	0	0	0	0
	Standards of coherence	0	0	0	3	0	0	3
	Evaluation of comprehension	1	0	0	3	0	0	2
	Drive for explanation	0	0	−1	3	0	3	1
	Embodiment	1	0	0	0	0	0	0
	Imagery	1	1	0	0	1	0	0

Figure C.16: Evaluation of comprehension models 2

APPENDIX D

Appendix D

This appendix presents further details about the mathematical foundation of Latent Semantic Analysis (LSA). It is a good starting point for readers who are not familiar with LSA.

D.1 Preliminary information about LSA

This section presents a preliminary information about the mathematical foundation of LSA. If you know Gram-Schmidt orthogonalization process and Singular Value Decomposition (SVD), you may skip this section. This section will present a summary of the references given below. For further reading, please refer to the bibliography.

1. Strang, Gilbert. "Introduction to linear algebra." Cambridge Publication (2003).
2. Baker, Kirk. "Singular value decomposition tutorial." The Ohio State University (2005).

D.1.1 Points, Vector, Space, Dimension and Coordinates

Point: List of numbers which specifies a position in a space

Coordinate: An ordered list of numbers

Space: A vector space V over a field F. Elements of V are vectors. Elements of F are scalars. A vector space has two operations; vector addition and vector multiplication.

Dimension: Length of Coordinate, the ordered list of numbers

Vector: Element of a space

Linear Combination: $cv + dw = \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}3\\4\end{bmatrix}$ is the combination over a 2 dimensional vector space with $c = d = 1$

D.1.2 Vector Operations

D.1.2.1 Vector Addition

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, v + w = \begin{bmatrix} v_1 + w_1 \\ v_2 + w_2 \end{bmatrix}$$

for example,

$$v = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, w = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, v + w = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

D.1.2.2 Scalar Multiplication

$$2v = \begin{bmatrix} 2v_1 \\ 2v_2 \end{bmatrix}, -w = \begin{bmatrix} -w_1 \\ -w_2 \end{bmatrix}, v + w = \begin{bmatrix} v_1 + w_1 \\ v_2 + w_2 \end{bmatrix}$$

for example,

$$v = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, w = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, v + w = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

D.1.2.3 Linear Combination

DEFINITION: The sum of cv and dw is a linear combination of v and w.

There are four special linear combinations: sum, difference, zero, scalar multiplication

1v+1w sum of vectors

1v-1w sum of vectors

0v+0w zero vector

cv+0w vector cv in the direction of v

D.1.2.4 Pictures of All combinations of a Vector

1. All combinations of cu fill a line
2. All combinations of $cu + dv$ fill a plane
3. All combinations of $cu + dv + ew$ fill a three-dimensional space

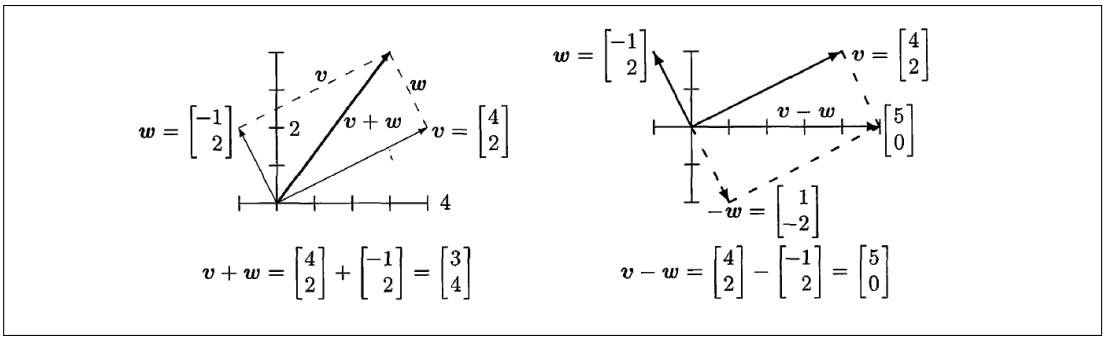


Figure D.1: Linear Combinations of Vectors

D.1.3 Vector Terminology

D.1.3.1 Vector Length

$v = (v_1, v_2, v_3, ..., v_n)$

$||v|| = \sqrt{S_{i=1}^{10} v_i}$

For example, if $v = [1, 2, 3, 4, 5]$, then $||v|| = \sqrt{1^2 + 2^2 + 3^2 + 4^2 + 5^2} = \sqrt{55} = 7.41$

D.1.3.2 Scalar Multiplication

if $v = [v_1, v_2, v_3, ..., v_n]$ and d is a scalar, then $dv = [dv_1, dv_2, dv_3, ..., dv_n]$.

D.1.3.3 Inner Product

$(\vec{x}, \vec{y}) = \vec{x}.\vec{y} = S_{i=1}^{n} x_i y_i$

For example, $\vec{x} = (1, 2, 3), \vec{y} = (4, 0, 1)$ then inner product of ($\vec{x}$ and $\vec{y}$) is

$\vec{x}.\vec{y} = 1 * 4 + 2 * 0 + 3 * 1 = 7$

D.1.3.4 Orthogonality

2 vectors are orthogonal if their inner product is equal to zero. For example, $\vec{v} = [1, 0]$ and $\vec{w} = [0, 1]$ are orthogonal because their inner product is zero.

$\vec{v}.\vec{w} = 1 * 0 + 0 * 1 = 0$

D.1.3.5 Normal Vectors

Normal vector is a vector whose length is 1. Any vector whose length ¿1 can be initialized to a unit vector by dividing its each component by its length..

For example, $\vec{v} = 3, 4$ has the following normalized vector

$||v|| = \sqrt{3^2 + 4^2} = 5$

Then, the normal vector of $\vec{v}$ is $\vec{v} = 3/5, 4/5$

D.1.3.6 Orthonormal Vectors

Vectors with a unit length that are orthogonal are called *orthonormal*. For example,

$\vec{u} = [2/5, 1/5, -2/5, 4/5]$

and

$\vec{v} = [3/\sqrt{65}, -6/\sqrt{65}, 4/\sqrt{65}, 2/\sqrt{65}]$

are orthonormal because

$\vec{u} = \sqrt{(2/5)^2 + (1/5)^2 + (-2/5)^2 + (4/5)^2} = 1$

$\vec{v} = \sqrt{(3/\sqrt{65})^2 + (-6/\sqrt{65})^2 + (4/\sqrt{65})^2 + (2/\sqrt{65})^2} = 1$

$\vec{u}.\vec{v} = \frac{6}{5\sqrt{65}} - \frac{6}{5\sqrt{65}} - \frac{8}{5\sqrt{65}} + \frac{8}{5\sqrt{65}} = 0$

D.1.3.7 Gram-Schmidt Orthonormalization Process

Gram-Schmidt Orthonormalization Process is a method to convert a set of vectors to its orthonormal vectors. Here are the steps of this process.

1. Convert the first vector to its orthonormal vector
2. Rewrite the remaining vectors in terms of themselves minus multiplication of already orthonormalized vectors.

For example, to convert the column vectors of

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 2 & 3 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

into orthonormal column vectors

$$A = \begin{bmatrix} \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{6} & \frac{2}{3} \\ 0 & \frac{2\sqrt{2}}{3} & \frac{-1}{3} \\ \frac{\sqrt{6}}{3} & 0 & 0 \\ \frac{\sqrt{6}}{6} & \frac{-\sqrt{2}}{6} & \frac{-2}{3} \end{bmatrix}$$

first normalize $\vec{v} = [1, 0, 2, 1] : \vec{v} = [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$. Then, normalize the second vector.

$\vec{w_2} = \vec{v_2} - \vec{u_1}.\vec{v_2} * \vec{u_1} = [2, 2, 3, 1] - [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}].[2, 2, 3, 1] * [\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}]$

$\vec{w_2} = [1/2, 2, 0, -1/2]$

Normalize $\vec{w_2}$ to obtain

$\vec{w_2} = [\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3}, 0, \frac{-\sqrt{2}}{6}]$

The following rule

$\vec{w_k} = \vec{v_k} - S\, i = 1k - 1\vec{u_i}.\vec{k} * \vec{u_i}$

can be applied to obtain the $\vec{w_3}$

D.1.4 Matrix Terminology

D.1.4.1 Square Matrix

A matrix is said to be square if its length of columns is equal to its length of the rows. For example, $A = \begin{bmatrix} 1 & 3 & 4 \\ 1 & 6 & 9 \\ 0 & 3 & 4 \end{bmatrix}$

D.1.4.2 Transpose Matrix

Transpose of a matrix A_{ij} is created by converting its columns into rows. The transpose of matrix A is A^T.

$$A = \begin{bmatrix} 1 & 3 & 4 \\ 1 & 6 & 9 \\ 0 & 3 & 4 \end{bmatrix}, A^T = \begin{bmatrix} 1 & 1 & 0 \\ 3 & 6 & 3 \\ 4 & 9 & 4 \end{bmatrix}$$

D.1.4.3 Matrix Multiplication

Matrix multiplication is different from the inner products of matrix pairs. Matrix multiplication is possible when the column number of the first matrix is equal to the row number of the second matrix.

The coordinates of AB are determined by taking the inner product of each row in A and each column in B. That is, if $A_1, ..., A_m$ are the row vectors of matrix A, and $B^1, ..., B^s$ are the column vectors of B, then ab_{ik} of AB equals $A_i.B^k$. For example,

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 16 \\ 0 & 26 \end{bmatrix}$$

D.1.4.4 Identity Matrix

Identity matrix is a matrix whose diagonal values are 1 and other values are 0. When identity matrix is multiplied by matrix A, $AI = A$.

Here is an example.

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix}$$

D.1.4.5 Orthogonal Matrix

Matrix A is orthogonal if $AA^T = I$, For example,

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & -4/5 \\ 0 & 4/5 & 3/5 \end{bmatrix}$$

is orthogonal because

$$AA^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & -4/5 \\ 0 & 4/5 & 3/5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & 4/5 \\ 0 & -4/5 & 3/5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

D.1.4.6 Determinants

Determinant is a function which reduces a square matrix to a scalar value. It is denoted as $|A|$ or $det(A)$. Here are some examples

$$A = [5], det(A) = 6\ A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, det(A) = ad - bc\ A = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}, det(A) = 1 * 5 - 3 * 2 =$$

-1

Finding the determinant of a n-square matrix where $n > 2$ is done as follows:

1. Disable the first row and column; if the remaining n-square matrix is a 2-square matrix, then add the determinant of the 2-square matrix as a scalar and calculate the determinant of the substituted matrix.
2. If disabling rows and column does not produce a 2-square matrix, go deeper and apply step 1.

Here is an example.

$$\begin{vmatrix} -1 & 4 & 3 \\ 2 & 6 & 4 \\ 3 & -2 & 8 \end{vmatrix} = (-1)\begin{vmatrix} 6 & 4 \\ -2 & 8 \end{vmatrix} - (4)\begin{vmatrix} 2 & 4 \\ 3 & 8 \end{vmatrix} + (3)\begin{vmatrix} 2 & 6 \\ 3 & -2 \end{vmatrix} - 1(6 * 8 - 4 * (-2)) - 4(2 * 8 - 3 * 4) + 3(2 * (-2) - 6 * 3) = -138$$

D.1.4.7 Eigenvectors and Eigenvalues

Eigenvector is a non-zero vector which satisfies the equation equation.

$$A\vec{v} = \lambda\vec{v}$$

,

where A is a square matrix,λ is a scalar and $\vec{v}$ is an eigenvector. Elements of eigenvectors are called Eigenvalues. Eigenvectors are also known as *characteristic vectors* or *latent vectors*. Please note that A is a matrix whereas *lambda* is a scalar which means that we have to convert the matrix A to its scalar values by calculating its determinant.

Let's find the eigenvectors of matrix below .

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

we have the following equation:

$$A\vec{v} = \lambda\vec{v} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \lambda\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$
$$\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

To satisfy the equation, we have to convert the first matrix to its scalar value by finding its determinant and the determinant value has to be zero if we want a non-zero matrix of $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

$(2-\lambda)*(2-\lambda)-(1*1)=0$
$\lambda^2-4\lambda+3=0$
$(\lambda-3)*(\lambda-1)=0$

We have two values of λ, $\lambda_1=3, \lambda_2=1$. Let's add the λ values to the equation we obtained before:

$$\begin{bmatrix}2-\lambda & 1\\ 1 & 2-\lambda\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=0$$
$$\begin{bmatrix}2-3 & 1\\ 1 & 2-3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=0$$
$$\begin{bmatrix}-1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=0$$

$-x_1+x_2=0, x_1-x_2=0, x_1=x_2$, for $\lambda=3$, we have $x_1=x_2$ which means $eigenvector_1=[1,1]$. Applying the same steps for $\lambda=1$, we will have $eigenvector_2=[1,-1]$

D.1.4.8 Singular Value Decomposition (SVD)

SVD is a process in which a set of vectors and are reduced to a new set of vectors where they have one variant of the original vectors. From this point of view, SVD can be considered a reduction process. The second approach is to evaluate SVD as a process of revealing the latent variables of the original vectors. In this case, the matrix of A, $A=USV^T$ is decomposed into three matrices which have reduced variables. At the time of reconstructing the original matrix A from the reduced matrices, the most variant variables, namely the latent relations of vectors, are sufficient to recover the original matrix. Assume that there is the original matrix A and its decomposed version is: $A=USV^T$. When we reduce certain unimportant variables from the original version, we will obtain a reduced version which is A, $A'=USV^T$. According to the second approach, when the length of difference of A and A' is reduced to zero, the most variant variable of the original matrix is revealed. Since there is a trade off when determining $|A-A'|=0$, we will have several versions of A'. The version which satisfies the condition of the least difference of A and A' is the matrix which stores the most variant variables, namely the latent information among the uncorrelated variables of the original matrix. Latent Semantic Analysis (LSA) highly relies on the process of SVD.

D.1.4.9 A Real Example for SVD

Let's have a matrix of $A=\begin{bmatrix}3 & 1 & 1\\ -1 & 3 & 1\end{bmatrix}$ and apply the SVD process on this matrix.

To convert A into its SVD equivalent $A=USV^T$, the following steps will be applied.

1. Find $X=AA^T$

2. Find eigenvectors of X
3. Normalize the eigenvectors of X
4. Apply the Gram-Schmidt Orthonormalization process to the normalized eigenvectors of X to obtain U
5. Find $Y = A^T A$
6. Find eigenvectors of Y
7. Normalize the eigenvectors of Y
8. Apply the Gram-Schmidt Orthonormalization process to the normalized eigenvectors of Y to obtain V^T

The first 4 steps are similar to the last 4 steps.

$$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$$

$$X = AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

Let's find the eigenvectors of X.

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{vmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{vmatrix} = 0$$

$$(11-\lambda)^2 - 1 * 1 = 0$$

$$(\lambda - 10)(\lambda - 12) = 0$$

$$(\lambda_1 = 10, \lambda_1 = 12$$

Let's add the eigenvalues back to the original equations.

$$\begin{bmatrix} 11-10 & 1 \\ 1 & 11-10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$(11-10)x_1 + x_2 = 0, x_1 = -x_2$$

Now we have the eigenvector of [1,-1] for $\lambda = 10$. For $\lambda = 12$, we will have the eigenvector of [1,1] which is represented below

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Now apply Gram-Schmidt Orthonormalization.

Normalize $\vec{v_1}$

.

$\vec{u_1} = \frac{\vec{v_1}}{|\vec{v_1}|} = \frac{[1,1]}{\sqrt{1^2+1^2}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$

Calculate the second vector.

$\vec{w_2} = \vec{v_2} - \vec{u_1}.\vec{v_2} * \vec{u_1}$
$[1,-1] - [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}].[1,-1] * [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] = [1,-1]$

Normalize [1,-1] to get $[[\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]]$

Now we have the U as follows:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

The same operation will be applied to obtain V^T. V is based on A^TA.

$$Y = A^TA = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$$

Find the eigenvalues of Y.

$$\begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

which will result in the following equation.

$$\begin{vmatrix} 10-\lambda & 0 & 2 \\ 0 & 10-\lambda & 4 \\ 2 & 4 & 2-\lambda \end{vmatrix} = 0$$

This is equal to the equation below.

$$(10-\lambda)\begin{vmatrix} (10-\lambda) & 4 \\ 4 & (2-\lambda) \end{vmatrix} + 2\begin{vmatrix} 0(& 10-\lambda) \\ 2 & 4 \end{vmatrix} = (10-\lambda)[(10-\lambda)(2-\lambda)-16] + 2[0 - (20-2\lambda)]$$
$$\lambda(\lambda-10)(\lambda-12) = 0$$

We obtain the following eigenvalues for Y: $\lambda_1 = 0, \lambda_2 = 10, \lambda_3 = 12$ f r Y. when we solve the following equation for three values of λ

$$\begin{vmatrix} 10-\lambda & 0 & 2 \\ 0 & 10-\lambda & 4 \\ 2 & 4 & 2-\lambda \end{vmatrix} = 0$$

we get $\vec{v_1} = [1,2,1]$ for $\lambda = 12$, $\vec{v_1} = [2,-1,0]$ for $\lambda = 10$, $\vec{v_1} = [1,2,-5]$ for $\lambda = 0$.

As a result we have the following matrix of eigenvectors.

$$V = \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & 2 \\ 1 & 0 & -5 \end{bmatrix}$$

When we apply Gram-Schmidt Orthonormalization to V, we will obtain the following matrix.

$$V = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{30}} \\ \frac{2}{\sqrt{6}} & \frac{-1}{\sqrt{5}} & \frac{2}{\sqrt{30}} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

To finish the decomposition of A as $A = USV^T$, we need the S matrix. The S matrix is obtained by taking the square roots of the eigenvalues in the eigenvector matrix. Since we have the two eigenvalues namely $\lambda_1 = 12, \lambda_2 = 10$, we have two values for our diagonal matrix. The number of columns and rows are determined to fit in the multiplication rule for U and V^T. Therefore the S matrix is as follows:

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$
.

Now, we have all the decomposed matrices of A. Then, calculate again to recover the matrix A.

$$A_{mn} = U_{mm}S_{mn}V^T_{nn} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

$$\begin{bmatrix} \sqrt{6} & \sqrt{5} & 0 \\ \sqrt{6} & -\sqrt{5} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

D.1.4.10 Example of Reduced SVD

A reduced SVD is an SVD technique which reduces the decomposed matrices of the original matrix A. Let's have $A_{td} = U_{tm}S_{mm}V^T_{md}$ and a reduced version as $A'_{td} = U_{tk}S_{kk}V^T_{kd}$ where $k < m$ and $|A_{td} - A'_{tk}| = min(|A_{td} - A'_{kd}|), 0 < k \leq m$.

A reduced SVD is the heart of the Latent Semantic Analysis. When a term-document matrix is decomposed as $A = USV^T$, the latent correlations between the row-tensor and the column-tensor are revealed as linearly independent components. Since these

components are numerical values, they can be used to observe and measure a latent relation for similarity or document retrieval purposes. When some of the revealed independent components are ignored and the original matrix is recovered through the multiplication of USV^T, an approximation to the original matrix is obtained.

Here is a reduced SVD version of the matrix.

$$A = \begin{bmatrix} 2 & 0 & 8 & 6 & 0 \\ 1 & 6 & 0 & 1 & 7 \\ 5 & 0 & 7 & 4 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 10 & 0 & 0 & 7 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 104 & 8 & 90 & 108 & 0 \\ 8 & 87 & 9 & 12 & 109 \\ 90 & 9 & 90 & 111 & 0 \\ 108 & 12 & 111 & 138 & 0 \\ 0 & 109 & 0 & 0 & 129 \end{bmatrix}$$

Lambda values of AA^T are $\lambda_1 = 321.07, \lambda_2 = 230.17, \lambda_3 = 12.70, \lambda_4 = 3.94, \lambda_5 = 0.12$. These lambda values are used to compute U as follows.

$$U = \begin{bmatrix} -0.54 & 0.07 & 0.82 & -0.11 & 0.12 \\ -0.10 & -0.59 & -0.11 & -0.79 & -0.06 \\ -0.53 & 0.06 & -0.21 & 0.12 & -0.81 \\ -0.65 & 0.07 & -0.51 & 0.06 & 0.56 \\ -0.06 & -0.80 & 0.09 & 0.59 & 0.04 \end{bmatrix}$$

$$A^TA = \begin{bmatrix} 79 & 6 & 107 & 68 & 7 \\ 6 & 136 & 0 & 6 & 112 \\ 107 & 0 & 177 & 116 & 0 \\ 68 & 6 & 116 & 78 & 7 \\ 7 & 112 & 0 & 7 & 98 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.46 & 0.02 & -0.87 & -0.00 & 0.17 \\ -0.07 & -0.76 & 0.06 & 0.60 & 0.23 \\ -0.74 & 0.10 & 0.28 & 0.22 & -0.56 \\ -0.48 & 0.03 & 0.40 & -0.33 & 0.70 \\ -0.07 & -0.64 & -0.04 & -0.69 & -0.32 \end{bmatrix}$$

Here is the S matrix where the least two dimensions are reduced. According to this reduction, the column vectors of U and the row vectors of V^T are also reduced. The singular values of S are also sorted in a descending order.

$$S = \begin{bmatrix} 17.92 & 0 & 0 \\ 0 & 15.17 & 0 \\ 0 & 0 & 3.56 \end{bmatrix}$$

$$A' = \begin{bmatrix} -0.54 & 0.07 & 0.82 \\ -0.10 & -0.59 & -0.11 \\ -0.53 & 0.06 & -0.21 \\ -0.65 & 0.07 & -0.51 \\ -0.06 & -0.80 & 0.09 \end{bmatrix} \begin{bmatrix} 17.92 & 0 & 0 \\ 0 & 15.17 & 0 \\ 0 & 0 & 3.56 \end{bmatrix} \begin{bmatrix} -0.46 & 0.02 & -0.87 & -0.00 & 0.17 \\ -0.07 & -0.76 & 0.06 & 0.60 & 0.23 \\ -0.74 & 0.10 & 0.28 & 0.22 & -0.56 \end{bmatrix}$$

$$= \begin{bmatrix} 2.29 & -0.66 & 9.33 & 1.25 & -3.09 \\ 1.77 & 6.76 & 0.90 & -5.50 & -2.13 \\ 4.86 & -0.96 & 8.01 & 0.38 & -0.97 \\ 6.62 & -1.23 & 9.58 & 0.24 & -0.71 \\ 1.14 & 9.19 & 0.33 & -7.19 & -3.13 \end{bmatrix}$$

It is observed that the a_{ij} values of A' has an approximation to the a_{ij} values of A.

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ:

Fen Bilimleri Enstitusu ☐

Sosyal Bilimler Enstitüsü ☐

UygulamalıMatematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı . . . . . .

Adı . . . . . .

Bölümü . . . . . .

TEZİN ADI

AUTOMATED COHERENCE DETECTION WITH TERM-DISTANCE PATH EXTRACTION OF THE CO-OCCURRENCE MATRIX OF A DOCUMENT

TEZİN TÜRÜ.....: Yuksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın.

2. Tezimin tamamı yalnızca Ortadoğu Teknik Üniversitesi kullanıcılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi yada elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.)

3. Tezim 1 yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi yada elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.)

Yazarın imzası ..... Tarih .....