DA, RUI, M.S. A Method of Determining Culture Related Users Using Computation of Correlation. (2015)
Directed by Dr. Jing Deng. 46 pp.

The provision of security on most of computer networks is based on the obtaining and exchange of a unique key between the communicating parties. It is, however, difficult to come up with a truly unique and random secret between two parties with the help from physical randomness. In this work, we focus on the problem of unique random number generation or derivation between users in online social networks. As a result of rapid development of Internet, online social networks provide a vast set of different user comments on different products and services. Such comments can inherently reflect the mindset or cultural background of those people who wrote them and it is possible to derive some unique randomness from such texts with some maneuver. We select movie reviews as the sandbox for our investigation. To manage textual content and search for certain hidden relations, the methodologies of text matching are studied. By looking the similarities of movie reviews from different users, we can refer insights into the cultural background and even predict future preferences from past comments. We present all of our findings here to aspire further investigation. We have investigated the correlation of movie reviews and studied the values of different weight assignments to the sentence and word relation. According to our results, synonym relations are the dominant positive association that impacts correlation value. We calculate correlation between review sets containing multiple reviews to avoid randomness. These correlations have then been used to evaluate and derive a unique random number. We target at a single review, and put it together with other reviews to obtain correlation values from different pairs of reviewers. Then the correlation value is binning to a 1-bit binary number. Through such a simplified

extraction, a unique random number can be generated by repeating the process of binning. Such unique random number is able to facilitate to secure information exchanges between the users. In our future work, we will explore such correlations to generate a practically usable unique secret for secret keys.

A METHOD OF DETERMINING CULTURE RELATED USERS USING COMPUTATION OF CORRELATION

by

Rui Da

A Thesis Submitted to
the Faculty of the Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2015

Approved by

____
Committee Chair

This thesis is dedicated to my parents and wife.

For their endless love, support and encouragement.

APPROVAL PAGE

This thesis written by Rui Da has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair ______________________________
Jing Deng

Committee Members ______________________________
Shan Suthaharan

____
Lixin Fu

____
Date of Acceptance by Committee

____
Date of Final Oral Examination

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Jing Deng, who constantly inspires, encourages, and guides me through the problems in my research. His wisdom and kindness have motivated me on my study and research. This work would not be successful without his support.

Besides, I would like to thank Dr. Shan Suthaharan and Dr. Lixin Fu, my other two thesis committee who take their time to offer me valuable suggestions during my thesis defense.

Finally, I would like to thank all the professors and my friends who were always supporting me and encouraging me for my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The provision of security on most of computer networks is based on the obtaining and exchange of a unique key between the communicating parties. Such unique keys usually require the equipment of random number generation. The aim of random number generation is to produce a number or a sequence of numbers which do not indicate any predictable pattern. In this way, every generated number is independent from its precedent [17]. The random numbers are classified into "true" random numbers and pseudo-random numbers. "True" random numbers are generated based upon physical phenomenon such as the noise. while pseudo-random numbers are usually generated from computer programming [10]. In our work, we aim to generate random numbers base on the correlation value between two specific movie reviews. We the use the generated unique random number to identify movie reviewers who have something in common of their culture background.

1.1 Development of Movie Industry on Internet

Movie serves as a media of information broadcast, a carrier of entertainment and also a repository of knowledge. It impacts people's life in a variety of ways. There was a survey of visitors to Notting Hill in London, the location of the famous movie "Notting Hill" by Hugh Grant and Julia Roberts [20]. The result indicated that most visitors to Notting Hill had a relatively clear image about the destination. The majority of those visitors also planned to travel to other film locations in the future. It is quite persuasive that movie does have some positive impacts on tourism. In addition, movie clips have been proved to be effective in improving education in the effective domain by Blasco et al. [5]. When discussing why they chose movie as the methodology, they stated that because movie is familiar, evocative, and nonthreatening for students. The experiment on movie clip teaching methodology indicated that it is well suited to the students who were immersed in the audiovisual culture. The students who need to learn effective and cognitive dimensions achieved significant high levels of motivation and involvement. Therefore, movie is also considered as one of the powerful resources which promotes reflective attitudes and provides learning linked to experience. Important as movie is, it becomes valuable for people to explore more about it from different aspect.

Due to the surprisingly rapid development of Internet, more and more human activities are moving from real society to this newly built virtual community. Obtaining newest information all over the world, studying profound knowledge from a large scale of fields and enjoying entertainment through various platforms are getting easier for everyone connected to Internet. Traditionally, people watch movie either in cinema or on television. However, as Internet has become the most crucial part of people's daily life, more and more movies become available through the Internet and people discuss their opinions of such movies. One of most famous platforms for online streaming is likely the Netflix. It provides not only movies but also all kinds of TV shows. The market value of Netflix was exposed to be about $1.48 billions at the year of 2006, and customers of Netflix spend around $40 on average [11]. By the time of 2009, Netflix hosted upwards of 100,000 titles on DVD and the number of subscribers is over 10 million [26]. Google play and Apple store are also ranked as the most popular online movie provider behind Netflix. Besides, there are a large number of websites which provides either online streaming or downloading. Compared to the settings in a theater, watching movie on computer, tablet or even cellphone has turned into a new norm in nowadays fast-pace life.

1.2 Importance of Movie Review

It is interesting to see that almost every website not only includes the description of movies but also provides a place for users to leave reviews or comments. Online review is proved to have significant impact on movie's box office revenue by some studies [7]. Thus, it becomes valuable to obtain more information of the consumers. The problem is how to collect or retrieve as much information as possible in an effective way. In social networks, such as Facebook and Twitter, user information is easier to collect. Detailed profile for a specific user is usually exposed for everyone, or at least part of the users, on the Internet. In addition, their activities, including webpages that they liked, games that they played or people whom they followed, all contribute to identify themselves more precisely. However, to retrieve user information on movie website is another story. On most websites, the only useful pieces of information are the reviews. It becomes significantly more difficult to retrieve the correct information for a specific user because of the limited information on user profile and lack of evident activity which implies the background of the user.

1.3 Text Matching Methodologies

To manipulate movie review for secret generation could be considered as a problem of text matching. No need to mention, search engines, leading by Google, are the most authoritative expertise for the domain of text matching. By typing fewest pieces of information, the powerful search engine is able to crawl throughout the Internet and hunt for what the exact information the users might be looking for [23]. To a search engine, relevance is not only about finding the Web page with the right words. But back to the early years, search engines did not do much more than this simplistic step. As a consequence, the search results were a limit number of value. Over the years, however, hundreds of factors have been added to determine the relevance, thus more accurate results are able to achieve. In recent years, the development of search engine also contributes the distribution of advertisement, which is much more accurate and relevant than the days when search engine just emerged. In social networks, Facebook and Twitter also have done overwhelmingly excellent job on advertising and recommendation of friends, public IDs, software and games using their tremendous user dataset. Another successful example of text matching is job searching websites [4], such as Indeed and Monster. They are able to match every detail displays on the resume, including skills and locations, and search the suitable positions or candidates for their users. With the development of technology in this field, talents have higher probability to find suitable positions and make better use of what they master. Moreover, text matching is frequently and widely used in academic. The best example is the softwares for plagiarism detection. Copying the content directly or just making few changes of words is easy to be detected by those softwares.

There are various text matching algorithms for all these methods of information manipulation and delivery. Famous as TF–IDF(term frequency–inverse document frequency) serves as the fundamental of other derived algorithms. The TF–IDF scheme is usually used to reflect the importance of a specific term in certain textual content by whitening each of the term. We will discuss TF–IDF as well as other schemes in detail in next chapter. To manage textual content and search for certain hidden relations, the methodologies of text matching from all of the previous mentioned areas can be referred to.

1.4 Network Security

The flourish of information retrieval and exchange inevitably raised a number of issues. Network security is certainly one of them. Network security is all about processing, transmitting and storing data in a secure approach so that it keeps unauthorized users away from manipulating the data illegally. There are mainly four aspects that should be concerned, i.e. secrecy, authentication, non repudiation and integrity control. Secrecy is to avoid information leaking caused by unauthorized actions, such as the spread of private information through some unprotected websites. Authentication is to issue permission to identify users so that the communication can be established with exchange of confidential information. Non repudiation is to keep signatures of users from being misused. And Integrity control is to protect online business, such as banking and online transaction [13]. Cryptography is the infrastructure for most network security issues. The mechanism of cryptography is to transform and encode the data so that it is not recognizable by unauthorized user or organizations during transmission. At the destination, the same cryptography mechanism is used to decode and convert data into a readable format. Cryptography becomes increasingly indispensable during recent years as human activities with confidential information exchange have frequently taken place on Internet.

Such cryptographic maneuvers usually require a secret key since Internet is a expansive place, every piece of information can travel to the corners of networks. It is thus necessary to build up a mechanism so that texts information can be exchanged inexplicably. The provision of security on most of computer networks is based on the generation and exchange of a unique key between the communicating parties. A key used during data transmission contains information which is known only by the sender and receiver. The modern usage of cryptographic key is divided into two categories, one is symmetric key and the other is public key. In symmetric key, the keys used for encryption and decryption are only accessible to the sender and receiver. Encryption key is set visible to both sender and receiver, but decryption key need to be obtained by computation on encryption key. As for public key, it can be inferred from the name that the key used for encryption is visible to public. However, the information used to calculate decryption key is protected and only accessible to the receiver [14].

It is however difficult to come up with a truly unique and random secret between two parties. In the field of computer science, the key usage is used as a methodology of determination for certificate restriction. Key usage defines the number of operations that are needed for a certificate. To apply a similar method to the algorithm we try to design, we use a unique random number to signal the correlation or similarity between two texts. The unique random number is generated based on a vector of correlation values between two users. By calculating the similarity, users have high correlation value in history will obtain the same unique number when they write another new review. This methodology certainly prevent the users from exchange information directly and thus protect the user privacy from leaking.

In this work, we focus on the problem of unique random number generation or derivation between users in online social networks. As a result, a user should be able to guess what the other user will be writing if given a certain number of reviews. We are trying to find a similar yet different approach to better serve the requirements for our target. The limit of information that can be retrieved increases difficulty to accomplish the result.

1.5 Document Organization

The following paper is divided into four chapters followed by the references and the appendix.

- Chapter II introduces the related work.
- Chapter III explains our schema and solution in detail.
- Chapter IV shows the simulations results to evaluate our work.
- Chapter V summarizes our work and discuss future works.

CHAPTER II

RELATED WORK

While there are many techniques related to what we are focusing on in this work, our work is highly related to text matching. Other areas such as network security and graph theory have been briefly discussed in Chapter I.

The technique of text matching is considered far more important in recent years. Thanks to the emergence of search engine and social networks, abundance of algorithms are out there. These have indeed inspired our work here. In the process of information retrieval, the traditional way is to compute the similarity between the targeted texts and given queries (in the case of queries). Several models have already been developed, including reverse document model, vector space model, and latent semantic model [28]. In this chapter, we discuss some of the algorithms.

2.1 Term Frequency–Inverse Document Frequency

Term frequency–inverse document frequency, also known as TF–IDF, is widely used for information retrieval and text mining. Term frequency is simply the times of a word that appears in a collection or corpus. And inverse document frequency is the methodology to assign each word with a weight so that the impact of commonly used words, such as "the", is diminished. TF–IDF is the product of term frequency and inverse document frequency. There are variants of both TF and IDF weights in different schemes. For TF weighting scheme, there exists binary, raw frequency, log normalization, double normalization 0.5 and double normalization K. The simplest choice is to use the raw frequency of a given word in the content. For IDF weighting scheme, there exits unary, inverse frequency, inverse frequency smooth, inverse frequency max and probabilistic inverse frequency. The inverse document frequency is used for indicating whether a word repeats frequently or rarely across all the documents. Then the TF–IDF is calculated as:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D) \tag{2.1}$$

Here the IDF is calculated as:

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2.2}$$

In Equation (2.2), N is number of documents in the corpus in total, t represents the target word, d represents the chosen document, and D is the corpus (i.e., the large structured set of texts that are used for statistical analysis, hypothesis testing, and frequency checking). The denominator represents the number of documents which contain word t. Then, IDF is calculated as the logarithm of the quotient.

TF is simply calculated as the frequency of the target word t appears in document d.

TF–IDF has high efficiency on matching the words in a specific query to documents which are relevant to the given query [19]. Although it is a relatively old scheme, its robustness and effectiveness still make it a basic factor for building other more advanced algorithms. In the year of 2007, a new algorithm developed based on TF–IDF was published [24]. Tata and Patel discussed why estimating the selectivity of cosine similarity predicates is a very difficult problem, and proposed a solution based on careful empirical observations about the distribution of the dot product of typical queries. They demonstrated that cosine similarity perform surprisingly accurate for measuring similarity between two strings. Having evaluation of the algorithm over three different dataset, it hits over 40 percent of accuracy of the actual selectivity. Their work also showed that space efficient and time efficient are both achieved by the proposed approach.

Berger et al. [2] also published a paper which examined the performance of several algorithms derived from TF–IDF on answer-finding, including adaptive TF–IDF, query expansion, statistical translation and Latent variable models. The adaptive TF–IDF adjusts IDF factors by hill-climbing for each word so that the corresponding answer is closer to its question. And in query expansion, they try to investigate and build relationship between questions and answers by generating a mapping from both the terms inside queries and that of responses. Statistical translation applies "translation model" to a query-response system. For example, words indicating locations are translated with high probability into a query word "where". At last, Latent variable models is a combination of several different models. The authors imported factored model and translation model to improve performance on question-answering systems. To achieve the result of their experiment, they imported two dataset: a collection of Usenet FAQs and a set of customer–response dialogues. All of the four algorithms they presented performed better than the traditional TF–IDF on the task of retrieving the correct answer to a question from a large candidate answer set. The techniques are diverse, and they are able to be unified by exploiting the availability of a training corpus of question and answer pairs. They also concluded the techniques based on statistical translation and aspect models performed better on bridging the "lexical chasm" between questions and answers than the TF–IDF techniques.

Ramos claimed in the paper [19] in 2003 that TF–IDF is a simple and efficient algorithm for matching a query to the desired documents. From the data he collected, results demonstrated that TF–IDF is capable of retrieving the highly relevant information for a specific query. Moreover, since TF–IDF scheme is relatively easy to program, it has the potential to become the ideal fundamental for more complicated and comprehensive algorithms. However, TF–IDF does have its limitations. That is, it does a weak job when it comes to synonyms. For example, if a user wants to find information about "priest", according to Berger and Lafferty [3], TF-IDF will skip relevant documents which are using a synonym such as "reverend". Indeed, synonyms in movie reviews would introduce a lot of issues for our research, as we will demonstrate in the rest of the work.

2.2 Latent Semantic Analysis

LSA, the abbreviation of Latent Semantic Analysis, is a theory for extracting and representing the contextual–usage meaning of words by statistical computations applied to a large corpus of text [15]. This technique only takes the raw text which is parsed into words and meaningful passages as its input [16]. It runs without any humanly defined concepts or tools related to semantics. It divides text into different parts and represents the text in a matrix. In the matrix, each row stands for a unique word and each column contains a text passage or some other context. The frequency of a unique word from a given passage or context will be recorded into the corresponding cell. In addition, cell entries are also weighted by a function, which computes the importance of the unique word as well as the degree to which the word type carries information in the domain of discourse in general. Lastly, singular value decomposition will be applied to the constructed matrix. In the article [16], LSA was proven to have the ability to achieve important components of the lexical and passage meanings by humans. However, it is also found that LSA lacks the capability to make use of some complex information with details and order.

In article [12], Hofmann developed a new approach called Probabilistic Latent Semantic Indexing. By implementing with the Expectation Maximization algorithm, the new method achieves the ability to handle synonyms and polysemous. In the experiment, the performance of PLSI was compared with the standard term matching method of term frequency, term frequency–inverse document frequency as well as Latent Semantic Indexing. Four medium size datasets, including 1033, 1400, 3204 and 1460 documents respectively, were used to conduct the experiment. Results showed that PLSI outperformed other methodologies by significant gain. PLSI takes advantage of statistic standard methods for model fitting, overfitting control and model combination. However, PLSI still has the shortcoming as LSA does, that is lack of the capability to use detailed and ordered information.

2.3 Vector Space Model

The vector space model is used to represent text documents into vectors. It is usually used in the context of information retrieval. To construct a vector space, the first step is classifying the importance of each term within the documents and choose the terms which will best represent the content of each document. Both documents and index terms build the basic vectors in a linear space. Therefore the space is determined by both the terms and the documents. The occurrence of a term in a document is stored in the vectors. The queries are also represented as basic vectors.

When comparing the query to the documents, the vectors representing the queries and documents are in turn relatively measured in the vector space. Wong and Raghaven proposed some important properties of the vector space in 1984 [27]:

a) The ability to add any two elements of the system to obtain a new element of the system
b) The ability to multiply any element by a real number
c) The vectors obey several basic algebraic rules

Based on VSM, Becker and Kuropka [1] proposed a new algorithm called Topic–based Vector Space Model (TVSM). This approach assume the dependence of terms by implementing stopword–list, stemming and thesaurus. This makes it possible to research the dependencies among those algorithms. Also, it provided a potential methodology to optimize natural language processing models.

Another article [22] proposed ConfWeight to solve the Text Categorization problems. It is based on VSM as well. The experiment was conducted by using following text collections, Reuters–21578, Ohsumed and Reuters Corpus Vol. 1. As concluded in the article, ConfWeight outperforms both TF–IDF and GainRatio by gaining a significant improvement in terms of accuracy. Additionally, the authors claimed that ConfWeight also has the ability to achieve better performance even if there is no feature selection.

The shortcoming of VSM is that it assumes independence relationship among terms, the lack of term relationship may lead to the loss of some hidden information. Although Becker and Kuropka's method may have solve the problem, term–to–term relationship still cannot be preserved.

2.4 Other Algorithms

Similarly, an algorithm for text similarity computation based on hamming distance was proposed by Zhong et al. [28]. Hamming distance is represented by the number of different characters in two strings. Here texts are transformed into corresponding vectors, and it is the vectors that are used for hamming distance calculation. Another algorithm for text similarity detection is SimHash. Introduced by Sadowski and Levin [21], the new hash function has the opposite goal to the common hash function. It produces the same or similar hash key for the texts with similar content, then the difference between two texts can be calculated by computing the distance between those two hash keys. Unfortunately, the author claimed that they only tested the method in a test set of 10000 files due to the limitation of time. Thus, its performance on large dataset is unclear.

Though the algorithms discussed above reveal different aspects to deal with text similarity problem, none has tried to use such computed results to generate unique random numbers. This is indeed the goal of this work. To reach our ultimate aim, a new approach based on cosine similarity will be introduced in next section.

CHAPTER III

SCHEME DESIGN

3.1 Overview

In this chapter, we introduce a new algorithm for text matching with the goal of unique random number generation. The algorithm is built upon the hypothesis that the correlation value of two movie reviews from two culturally related reviewers is higher than two random reviewers. Language usage can be diverse from different regions. People from different countries express themselves differently even if they share the same native language, such people from U.S.A. and England. The difference is even greater between native and non–native speakers. Boroditsky stated in [6] that one's native language plays a significant important role in shaping habitual thought. Since we are not able to obtain any information about the cultural background of a movie reviewer, the following approaches will facilitate to demonstrate our hypothesis. One of the approach is to allow two reviewers to generate a unique secret without explicit exchange of any information other than the movie reviews. An intermediate goal can be to ask them to pick the same number between 0 and $Z = 1000$. In another way, we can assume that Reviewer A and B share N number of reviews and then each of them writes a new review on a movie. They can keep the new review secret, but then they can guess what the other reviewer writes about the new movie based on their review history.

3.2 Text Processing

Different from the algorithms which have been discussed in chapter II, we intend to construct an algorithm that has the ability to obtain all the relationships among words and phrases. Before stepping into more detail about how to preserve the relationships, we first introduce two powerful tools which facilitate to text process.

ReVerb

ReVerb is a open source information extraction software that identifies and extracts binary relationships automatically from given texts. It extracts information which cannot specify the target relations in advance [9]. ReVerb first identifies the relation phrases from target text that satisfy the syntactic and lexical constraints. Base on the results, it matches a pair of noun-phrase arguments to each of the resulting relation phrase. The final extractions are assigned with a confidence score using a logistic regression classifier.

ReVerb takes a POS–tagged and NP–chunked sentence and returns a set of (x, r, y) extraction triples. POS (Part–Of–Speech) tagger is a software that reads text and assigns parts of speech to each word, such as noun, verb, adjective, etc. [25]. Given an input sentence s, ReVerb uses the following extraction algorithm:

> **Relation Extraction:** For each verb v in s, find the longest sequence of words r_v such that (1)r_v starts at v, (2)r_v satisfies the syntactic constraint, and (3)r_v satisfies the lexical constraint. If any pair of matches are adjacent or overlap in s, merge them into a single match.
>
> **Argument Extraction:** For each relation phrase r identified in relation extraction, find the nearest noun phrase x to the left of r in s such that x is not a relative pronoun, WHO-adverb, or existential "there". Find the nearest noun phrase y to the right of r in s. If such an (x, y) pair could be found, return (x, r, y) as an extraction [8].

To better elaborate how ReVerb works, we take an example of the extraction algorithm in action, consider the following input sentence:

Jim is addicted to Java, which is a programming language.

The first step identifies three relation phrases that satisfy the syntactic and lexical constraints: *is, addicted to, and is.* The first two phrases are adjacent in the sentence, so they are merged into the single relation phrase *is addicted to.* The second step searches left and right of each relation phrase to find an argument pair. For *is addicted to*, the nearest NPs are (*Jim*, *Java*). For *is*, the extractor skips over the NP *which* and chooses the argument pair (*Java*, *programming language*). Then the results are:

result1: (Jim, is addicted to, Java)
result2: (Java, is, programming language)

As shown in above example, texts are divided into short extractions in the form of 3–tuple. By analyzing the extractions from ReVerb, a positive consequence that each extraction constructs a complete and meaningful sentence is achieved. Thus, we are able to eliminate redundant information as well as preserve the meaning of the original text.

WordNet

Our goal is to determine whether two specific movie reviewers have something in common about their cultural background. Only compare the original meaning of the movie review can hardly be sufficient. Since reviewers are likely to use different semantic sentences to express their ideas, a tool which deals with synonym becomes a key component to implement our algorithm. Because every sentence from texts is the combination of different meaningful words, the system requires the information about the set of meaningful words to process natural language. This information is usually retrieved by human from traditional dictionaries, but there are more and more dictionaries designed for machine available recently. WordNet is one of them which provides a combination of traditional lexicographic information and modern computing. It is an online lexical database designed for use under program control. The speech parts nouns, verbs, adjectives and adverbs of English are classified into different sets of synonyms. The synonym sets are connected by semantic relations. The semantic relations in WordNet include two important advantages. First, they apply to common used English broadly. Second, they are familiar to most users and easy to understand, even for users who have no professional knowledge to linguistics. Table 1 illustrates all the semantic relations in WordNet and some examples for each of them [18].

Table 1. Semantic Relations in WordNet

Semantic Relation	Syntactic Category	Examples
Synonym (similar)	Noun, Verb, Adjective, Adverb	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonymy (opposite)	Adjective, Adverb (Noun, Verb)	wet, dry powerfuly, powerless friendly, unfriendly rapidly, slowly
Hyponymy (subordinate)	Noun	sugar maple, maple maple, tree tree, plant
Meronymy (part)	Noun	brim, hat gin, martini ship, fleet
Troponomy (manner)	Verb	march, walk whisper, speak
Entailment	Verb	drive, ride divorce, marry

With the powerful semantic relation sets in WordNet, our system will become practicable not only to preserve relations of words and phrases from texts but also to build new relations from the original ones.

3.3 Convert Texts into Matrices

In this section, we will elaborate the crucial part of our novel method. As stated in previous sections, the fundamental mechanism is to keep all the semantic relations from two given movie reviews by ReVerb, and introduce new semantic relations based upon original ones by exploring the hidden information using WordNet. Then, we transform texts into matrices, and illustrate all the semantic relations obtained.

The intention for transforming texts into matrices is to generate semantic relation mathematically so that correlations are computable. To achieve two comparable matrices, we need all resulting words from previous processing of both movie reviews. Assume that we have two reviews r_1 and r_2, the word sets for each review are denoted as $ws_1 = \{w_{11}, w_{12}, ..., w_{1m_1}\}$ and $ws_2 = \{w_{21}, w_{22}, ..., w_{2m_2}\}$. The set of all words from both reviews is then computed by the intersection of ws_1 and ws_2, denoted by $ws_{full} = \{w_1, w_2, ..., w_n\}$. The resulting matrix is shaped as in Table 2.

Table 2. Word Matrices Built from Two Reviews

$$\begin{array}{cc}
 & \begin{array}{cccccc} w_1 & w_2 & w_3 & w_4 & \dots & w_n \end{array} \\
\begin{array}{c} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ w_n \end{array} &
\begin{pmatrix}
n_{11} & n_{12} & n_{13} & n_{14} & \dots & n_{1n} \\
n_{21} & n_{22} & n_{23} & n_{24} & \dots & n_{2n} \\
n_{31} & n_{32} & n_{33} & n_{34} & \dots & n_{3n} \\
n_{41} & n_{42} & n_{43} & n_{44} & \dots & n_{4n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
n_{n1} & n_{n2} & n_{n3} & n_{n4} & \dots & n_{nn}
\end{pmatrix}
\end{array}$$

Unlike other algorithms as discussed in Chapter II, we assign a value, called "label", to each cell indicating the relation between two words other than the most used "frequency". We use the example extracted from ReVerb, "Reporter Maria; like; car", to elaborate how our labeling system works. And here we assume that the synonyms retrieved for "reporter" and "car" are "journalist" and "vehicle" respectively. The relations are labeled as "CN_SYNONYMS", "CN_INTRA_PHRASE", "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_PHRASE". "CN_SYNONYMS" is the relation between a word and all its synonyms produced by WordNet. "CN_INTRA_PHRASE" and "CN_INTER_PHRASE" represent the relation of words within a single phrase and the relation of words from different phrases respectively. At last, "CN_SYNONYMS_INTER_PHRASE" connects the synonyms of two words from different phrases. Thus, from the previous example, the words "reporter" and "journalist" are connected by "CN_SYNONYMS"; "reporter" and "Maria" are connected by "CN_INTRA_PHRASE"; "Maria" and "like" are connected by "CN_INTER_PHRASE"; "journalist" and "vehicle" are connected by "CN_SYNONYMS_INTER_PHRASE". Table 2 illustrates the matrix of connection labels we are building, while Table 3 illustrates how this labeling system works.

Table 3. Labeling System for Word Relations

Label	Word Relation	Example
CN_SYNONYMS	Word and its synonyms	reporter-journalist
CN_INTRA_PHRASE	Words within same phrase	reporter-Maria
CN_INTER_PHRASE	Words from different phrases	Maria-like
CN_SYNONYMS_ INTER_PHRASE	Synonyms of words from different phrases	journalist-vehicle

3.4 Correlation Computation Using Cosine Similarity

Having generated the desired matrices, we step into the next phase which is to calculate the similarity. In our work, cosine similarity is used for this specific task. Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. It is a judgment of orientation which leaves out the influence of magnitude. It is particularly important that these bounds apply for any number of dimensions. Given two vectors A and B, the cosine similarity is represented as:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{3.1}$$

There are two key factors which impact the choice of our similarity computation algorithm. One is the simplicity. Since cosine similarity takes in two vectors and calculates the inner product of them to achieve the similarity value between them, we merely need to transform the matrices into vectors to serve as the input of cosine similarity. Also, the inner product is trivial in terms of programming, which is likely to reduce the time consumption for program runtime. The other important factor is, cosine similarity abstract out the magnitude of given vectors. That is, it takes out the influence of the document length, thus only the value inside the vector matters.

In our case, we only care about half of the resulting matrix since it is symmetric by diagonal. Thus, we consider to transform the right top half of each matrix into a vector. Half of the matrix we used for transforming is shown as in Table 4. By this method, we obtain the resulting vectors in the form of $v = \{n_{12}, n_{13}, n_{23}, n_{14}, n_{24}, n_{34}, n_{15}, ..., n_{(n-1)n}\}$.

Table 4. Half of Word Matrices Built from Two Reviews

$$\begin{array}{cc}
 & \begin{array}{cccccc} w_1 & w_2 & w_3 & w_4 & \cdots & w_n \end{array} \\
\begin{array}{c} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ w_n \end{array} & \left(\begin{array}{cccccc}
 & n_{12} & n_{13} & n_{14} & \cdots & n_{1n} \\
 & & n_{23} & n_{24} & \cdots & n_{2n} \\
 & & & n_{34} & \cdots & n_{3n} \\
 & & & & \ddots & \vdots \\
 & & & & & n_{(n-1)n} \\
 & & & & &
\end{array} \right)
\end{array}$$

The algorithm is now completed by equipping all these tools and methodologies introduced in this section. We will test the functionality by using real data to analyze the applicability in Chapter IV.

CHAPTER IV

EXPERIMENTATION AND EVALUATION

4.1 Data Collection and Processing

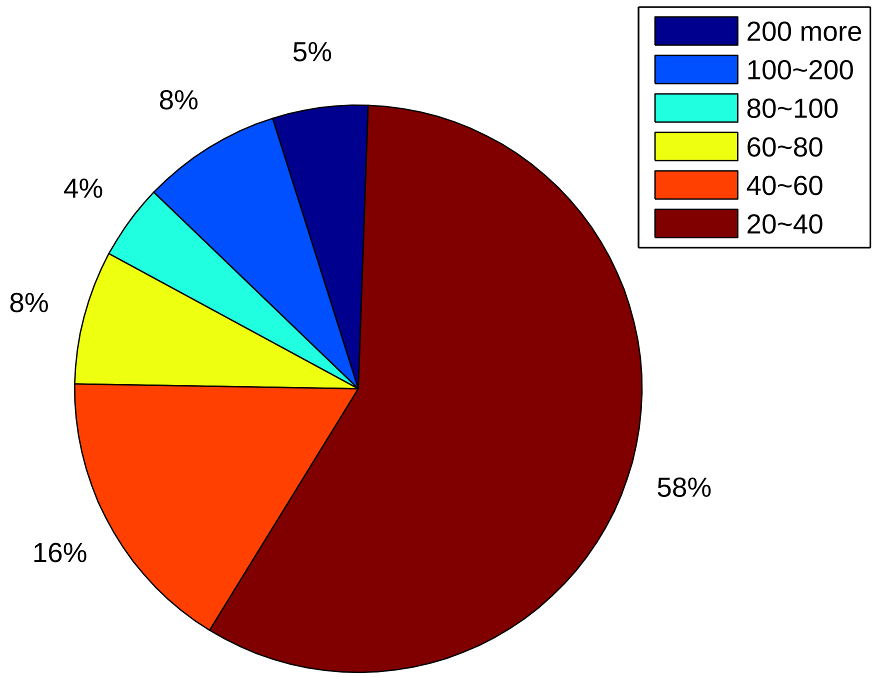

Figure 1. A Pie Graph of Reviewer Groups. The graph illustrates the number of reviewers from each group. Reviewers are divided into groups by the number of reviews they wrote. The review number of each group are shown as in the legend.

In order to ensure both accuracy and practicability of the evaluation, all the movie reviews in our dataset are retrieved and collected from websites through Internet. From the dataset, no personal information other than user ID can be obtained. Thus, data processing and manipulation are concentrated on the review content only. There are approximately 2 million reviews from 889,036 reviewers in total after we eliminated duplicate copied. However, among those reviews, only 7,201 of them wrote more than 20 reviews. An illustration of the sectional graph for those reviews we concentrate on is shown in Figure 1. As shown in the graph, most of the reviewers in our dataset have 20 to 40 reviews.

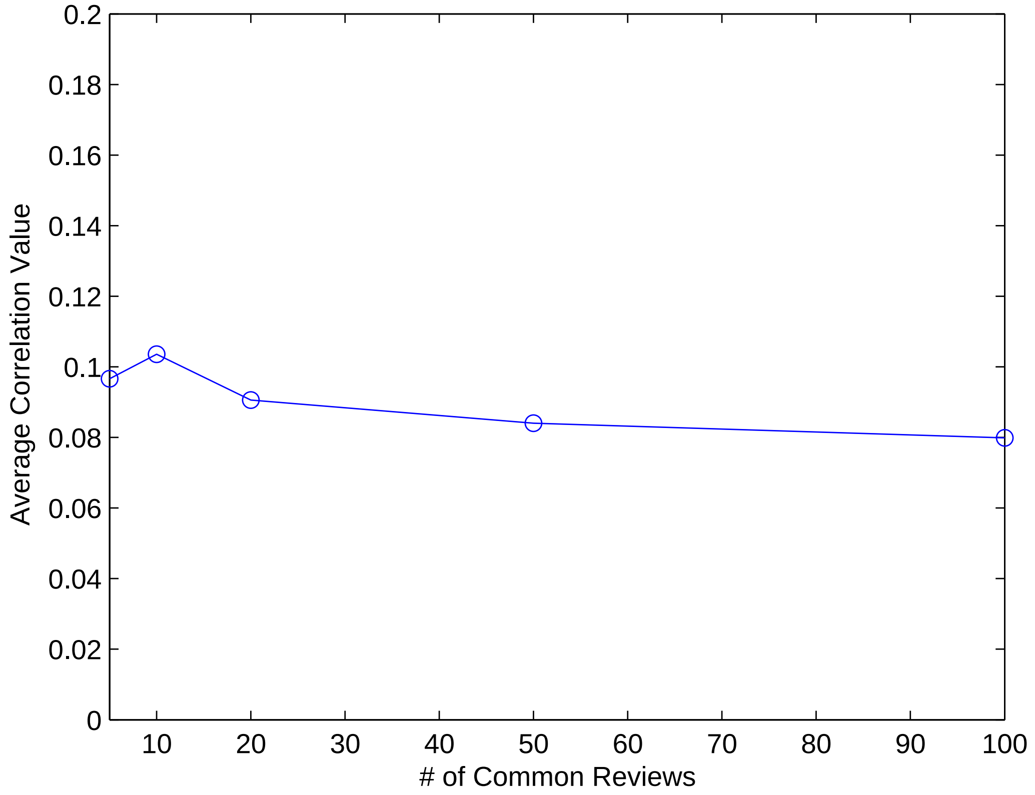

Figure 2. Comparison of Reviewer Groups Sharing Common Reviews. Results are between correlation values from 5 different pairs of reviewers. They are reviewers who have 5, 10, 20, 50 and 100 common reviews respectively. For each pair of reviewers, we compute the correlation value of each pair of common movie reviews. Then we use those values to calculate the average correlation value of each pair of reviewers.

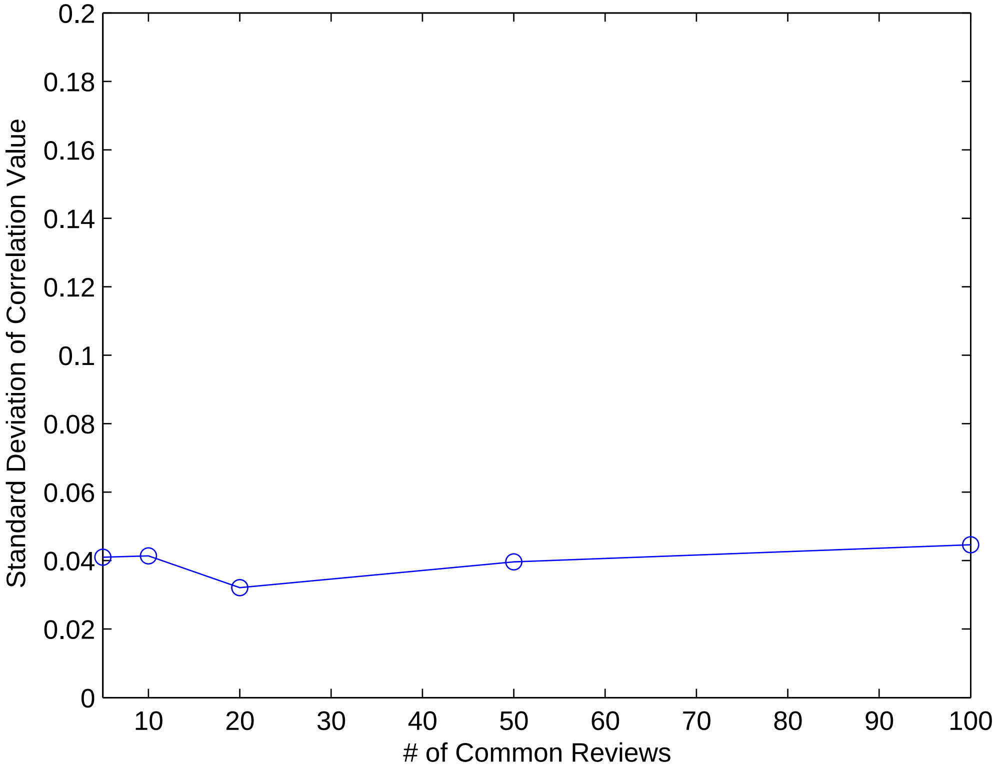

Figure 3. The Standard Deviation for Figure 2.

For the purpose of excluding incorrect result from distracting random data, we only execute our algorithm and analyze the data base on the reviewers with more than 20 reviews. We conducted some computations on different reviewers with different numbers of movie reviews. The comparisons are between different pairs of reviewers who have 5, 10, 20, 50 and 100 reviews. We compute the correlation value, i.e. the cosine similarity, of each pair of the common movie reviews, and then we compute the average and standard deviation of the correlation values. The results are shown in Figure 2 and 11. According to the result, correlation values between two reviewers who have less common reviews are more likely to fluctuate in a larger range. The elimination of reviewers have few reviews seems to have bias on the dataset. However, the actual impact is small since reviewers with fewer reviews are not likely to watch movies frequently, and thus their reviews may be written in low quality. Such movie reviews also tend to mislead our results to become incorrect.

4.2 Experiment Results Analysis

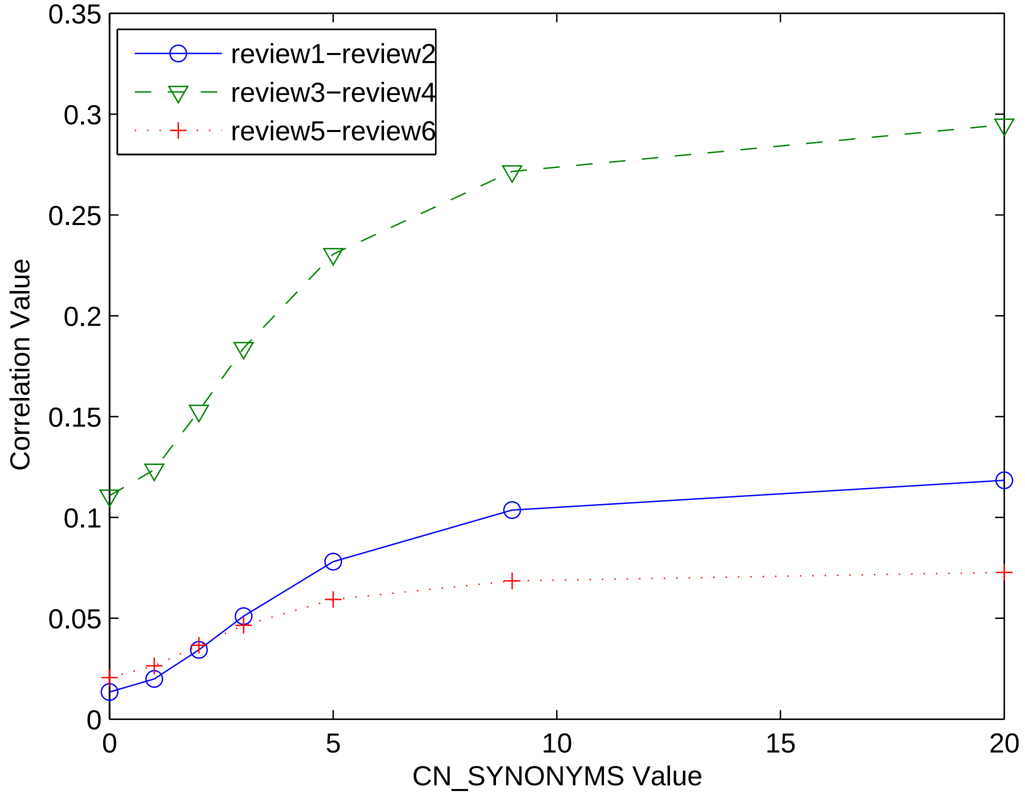

Figure 4. Correlation of Different "CN_SYNONYMS". Results are from 3 pairs of random reviews. Testing on variation of "CN_SYNONYMS" value of 0, 1, 2, 3, 5, 9 and 20. Other connection values are set to 1.

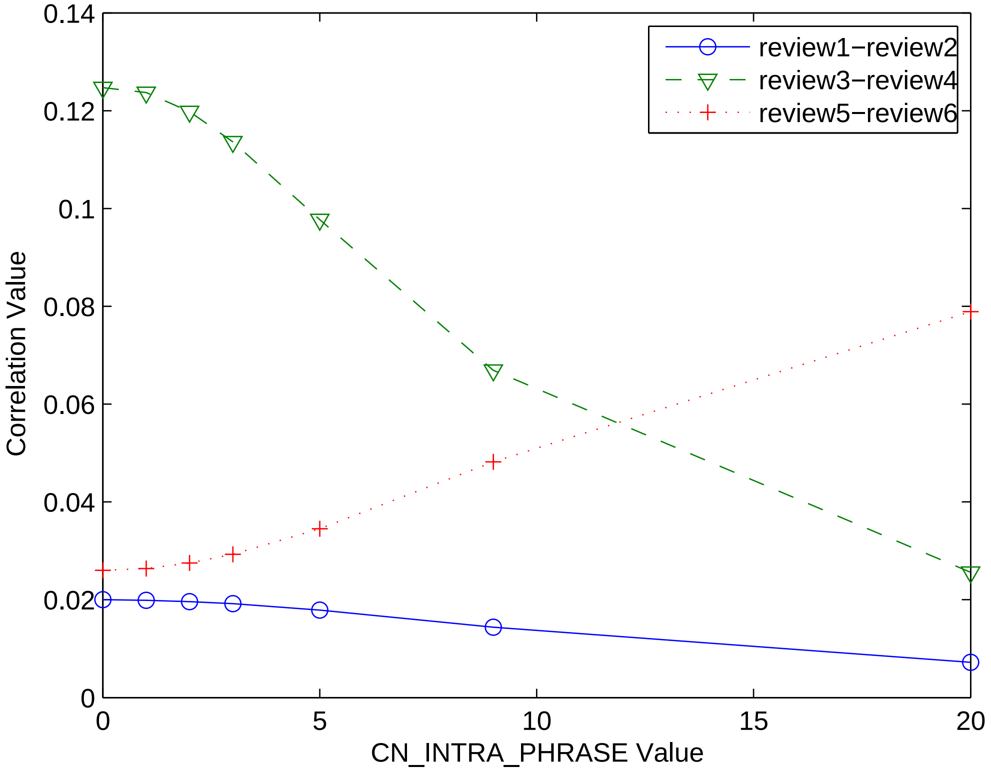

Figure 5. Correlation of Different "CN_INTRA_PHRASE". Results are from 3 pairs of random reviews. Testing on variation of "CN_INTRA_PHRASE" value of 0, 1, 2, 3, 5, 9 and 20. Other connection values are set to 1.

Then the examination on how the semantic relations impacts the correlation value is conducted. Labels as shown in Table 3 are assigned different values to build multiple combinations for test. First tested connection is "CN_SYNONYMS". We assign the value 0, 1, 2, 3, 5, 9 and 20 to it and value 1 to other connections, and compute the correlations of 3 random pairs of reviews. The results indicate a positive association between "CN_SYNONYMS" value and correlation value as illustrated in Figure 4. However, the results of "CN_INTRA_PHRASE" fail to show a regular change based upon the similar examination as in Figure 5. There are both positive and negative association between "CN_INTRA_PHRASE" value and correlation value for different pairs of movie reviews.

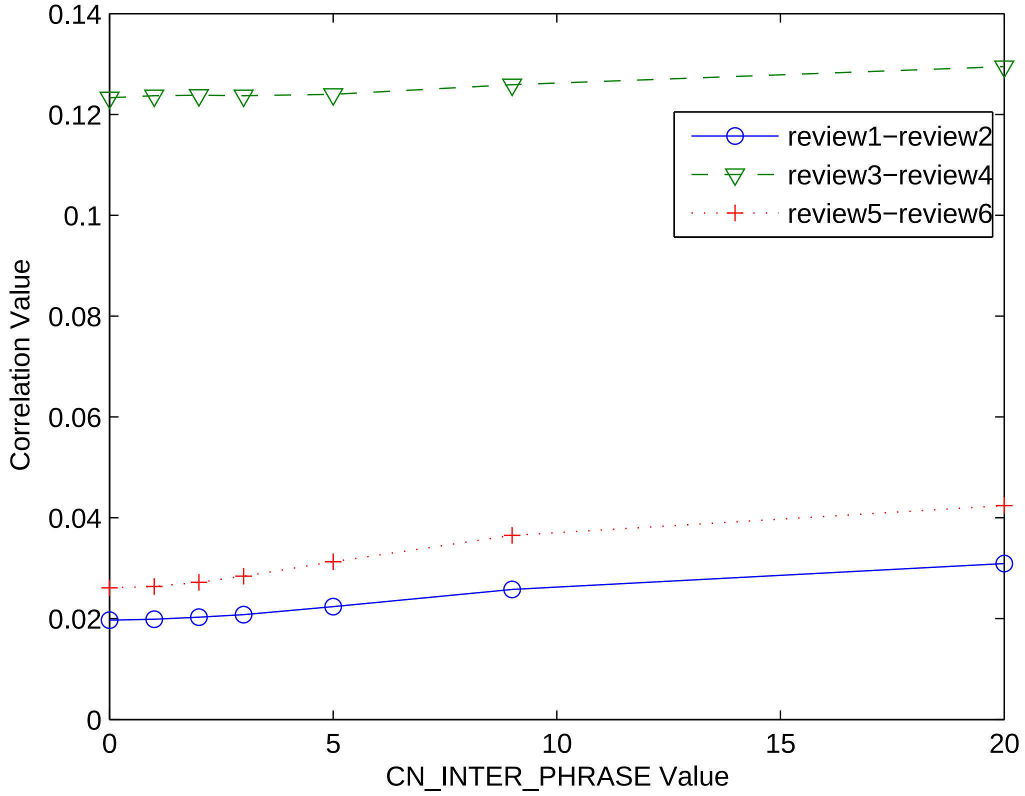

Figure 6. Correlation of Different "CN_INTER_PHRASE". Results are from 3 pairs of random reviews. Testing on variation of "CN_INTER_PHRASE" value of 0, 1, 2, 3, 5, 9 and 20. Other connection values are set to 1.

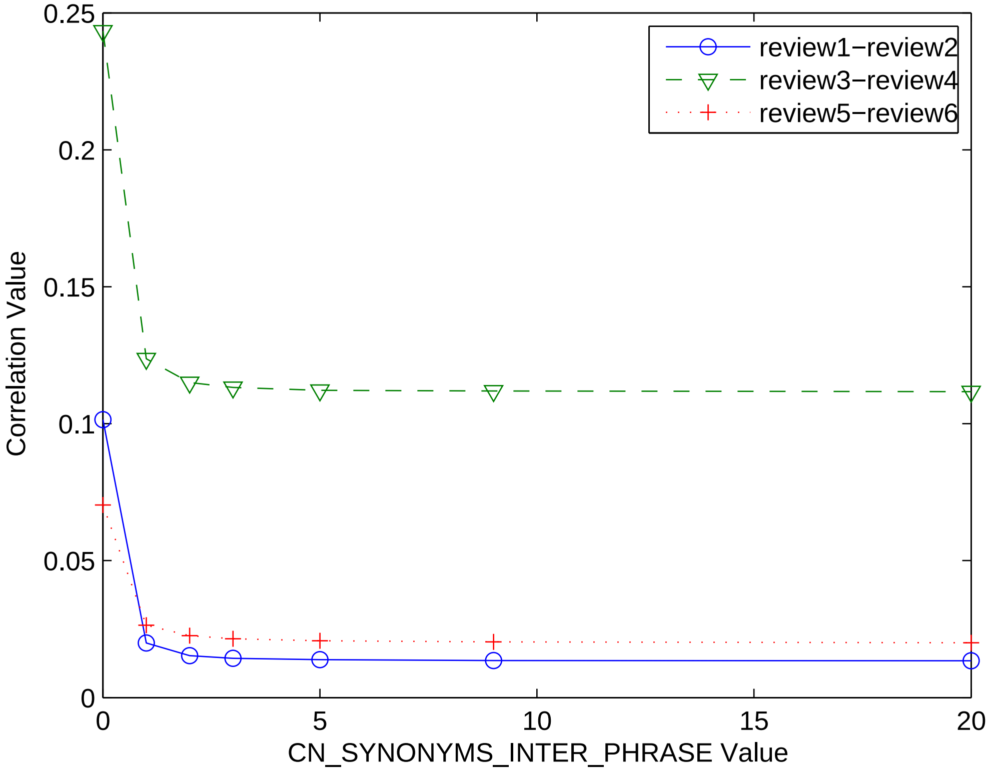

Figure 7. Correlation of Different "CN_SYNONYMS_INTER_PHRASE". Results are from 3 pairs of random reviews. Testing on variation of "CN_SYNONYMS_-INTER_PHRASE" value of 0, 1, 2, 3, 5, 9 and 20. Other connection values are set to 1.

We experiment the same combination variation on "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_PHRASE" connections as well, but the increase of connection values does not have significant impact on correlation values for both of the connections as depicted in Figure 6 and 7.

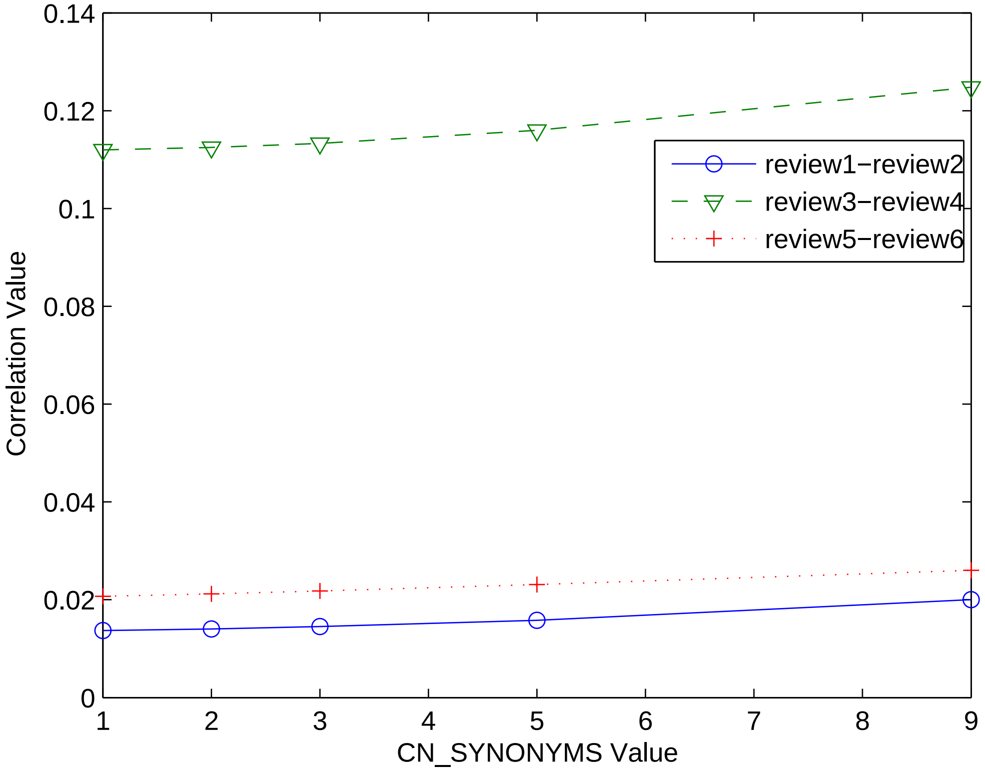

Figure 8. Correlation of Different Synonym and Inter Connections. Results are from 3 pairs of random reviews. Testing on variation of "CN_SYNONYMS" value of 1, 2, 3, 5 and 9. "CN_INTRA_PHRASE" connection is set to 1 and other connection values are set to 9.

Since our aim is to find out the hidden relation of two texts other than just term frequency, we assign a relatively larger value to the label "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_PHRASE" to investigate the impact of relations between different phrases on "CN_SYNONYMS" relation. Figure 8 shows the result from different combinations. We set both the "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_PHRASE" connections to 9, and "CN_SYNONYMS" connection increases from 1 to 9. It can be seen from the figure that when assigning a large value to the "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_-

PHRASE" connections, the impact of "CN_SYNONYMS" is minimized nearly to none. We can infer from the results that computation results would be less random by setting a relative larger connection value to "CN_INTER_PHRASE" and "CN_SYNONYMS_INTER_PHRASE" connections.

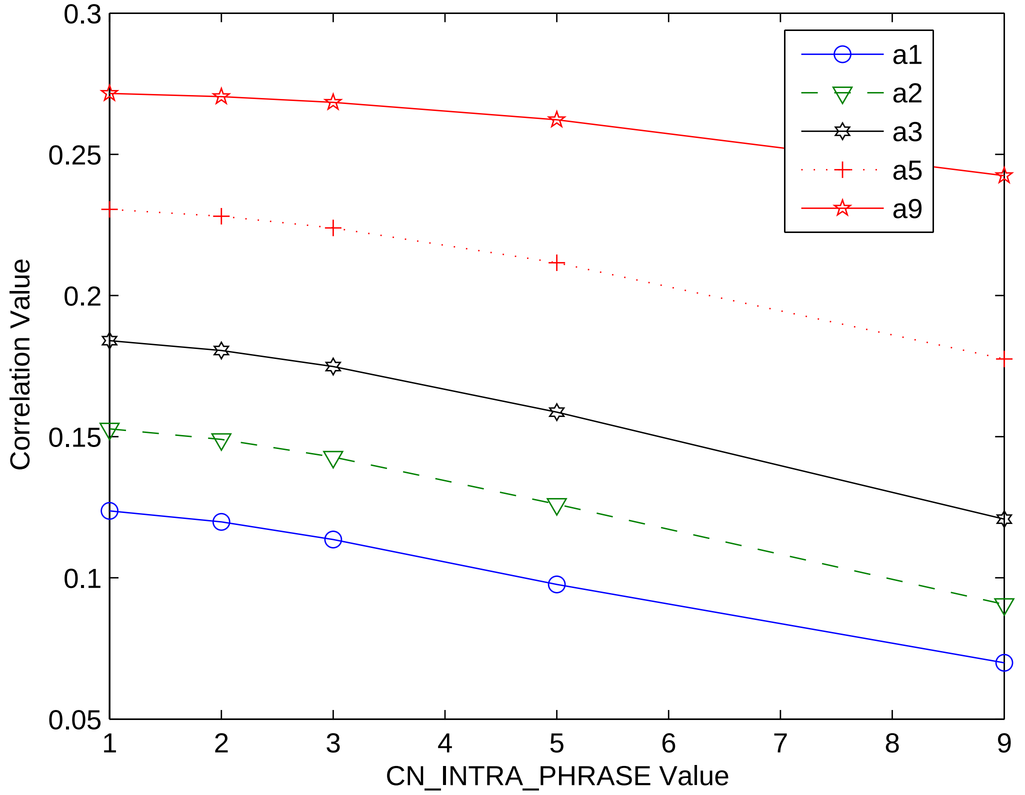

Figure 9. Correlation of Different "SYNONYMS" and "INTRA". Results are from one random pair of movie reviews. Testing on combinations of "CN_SYNONYMS" and "CN_INTRA_PHRASE" value of 1, 2, 3, 5 and 9. x-axis represents "CN_-INTRA_PHRASE" values, y-axis represents correlation values. 5 lines represent 5 different "CN_SYNONYMS" values as shown in legend.

We also experiment the relation of "CN_SYNONYMS" and "CN_INTRA_-PHRASE" by testing different combinations of label assignments as illustrated in

Figure 9. Connections are assigned with value 1, 2, 3, 5 and 9. According to the result, it is clear "CN_SYNONYMS" relation has a more significant impact on correlation value than "CN_INTRA_PHRASE" relation.

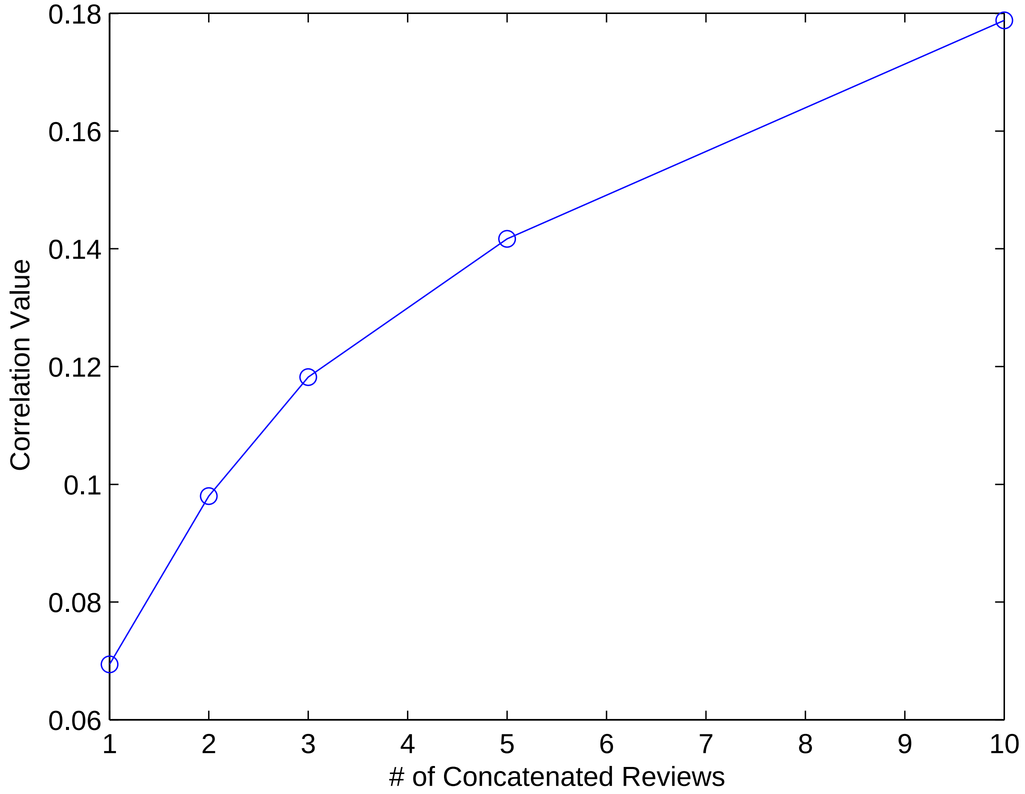

Figure 10. Correlation of Different Partitions of 60 Reviews. We compute the correlation of single review pairs as well as the correlation of multiple reviews concatenated together. In this experiment we examine the results of 2, 3, 5 and 10 reviews put together.

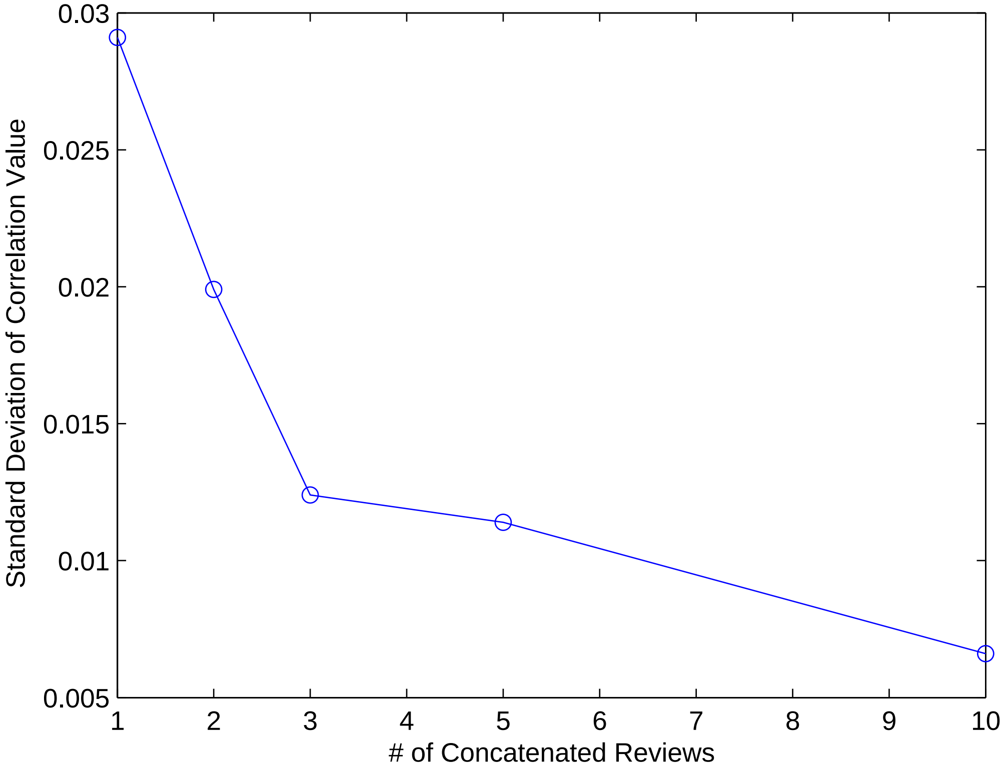

Figure 11. The Standard Deviation of Correlation in Figure 10.

Having the results from previous experiments, we put multiple movie reviews together and calculate the correlation of them. Two random reviewers with more than 60 review are selected to conduct the test. The 60 reviews are randomly selected and divided into groups of 1, 2, 3, 5, 10 reviews respectively. We calculated the average correlation value and standard deviation of each group. The results is shown in Figure 10 and 11. The results illustrate that the more movie reviews we put together, the larger correlation we can achieve, and the standard deviation of the correlation is smaller.

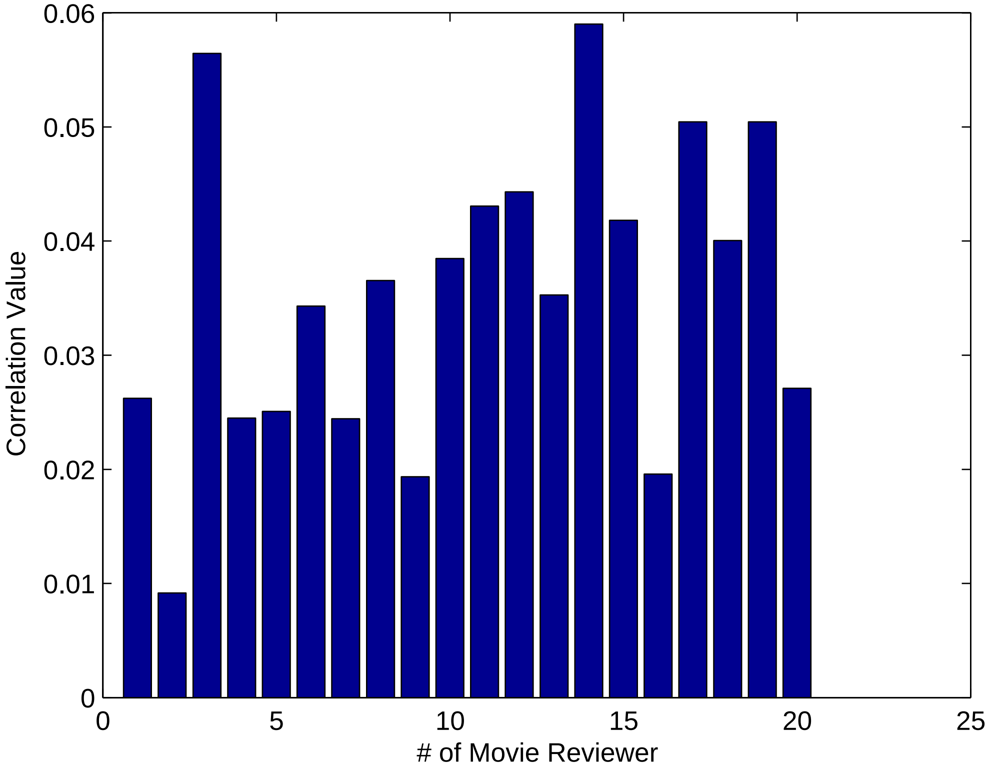

Figure 12. Correlations of 20 Pairs of Reviewers. The average correlation between a specific reviewer and 20 reviewers who have more than 10 common movie reviews. Each pair randomly picks out 20 movie reviews to calculate the average.

According to the previous results, we focus on the computation of variations of "CN_SYNONYMS" and "CN_INTER_PHRASE" connections. Now we need to analyze the data between some unique reviewers. First, we pick a specific reviewer and 20 reviewers who have more than 10 common movie reviews. Then we randomly pick 20 common reviews from the specific reviewer and each of the other 20 reviewers, and calculate the average of each pairs of reviewer. The results are shown in Figure 12. For the 20 chosen pairs of reviewers, the largest correlation is close to 0.06, and the smallest value is around 0.01. Having a list of reviewers of different correlations, we are able to make further comparison among these reviewers.

We expect that the culture factor can be inferred by the correlation value. Thus we calculate the correlation of reviews from the reviewer himself, and use that value as a benchmark because people are certainly culturally related to themselves. We use the same specific reviewer as mentioned above, say r_t ,and randomly choose 20 pairs of reviews to calculate correlation. The average correlation value is 0.039. Then we choose the reviewer, say r_1, having large correlation with r_t as listed in Figure 12, randomly pick out 20 pairs of reviews between r_t and r_1 to calculate correlation. Also, we choose the reviewer with smallest correlation, say r_2, in the same way as previous one.

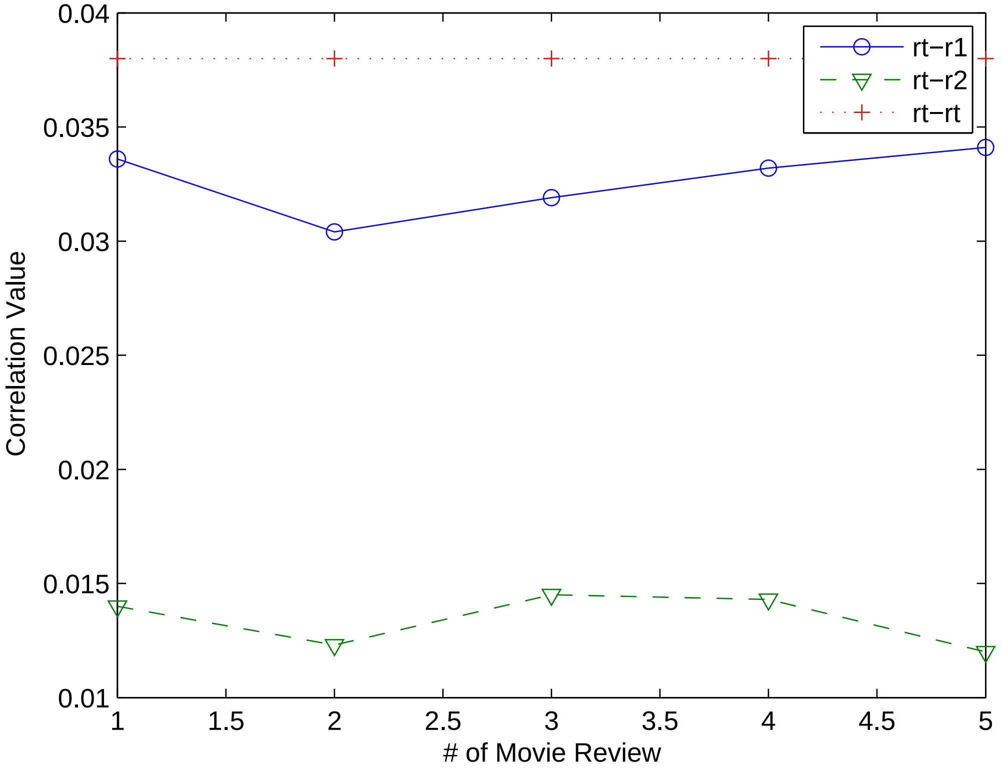

Figure 13. Average Correlation of Random Reviews. The average correlation of 20 random movie reviews between a specific reviewer r_t and two other reviewers r_1, r_2. r_1 has high correlation with r_t on common movies. r_2 has low correlation with r_t on common movies. $r_t - r_t$ represents correlation between reviews of r_t and himself. r_t-r_1 represents correlation between reviews of r_t and r_1. r_t-r_2 represents correlation between reviews of r_t and r_1.

It illustrates in Figure 13 that correlation value between r_t and r_1 is very close to correlation of r_t with himself, and is much larger than correlation between r_t and r_2. This result confirms our hypothesis positively.

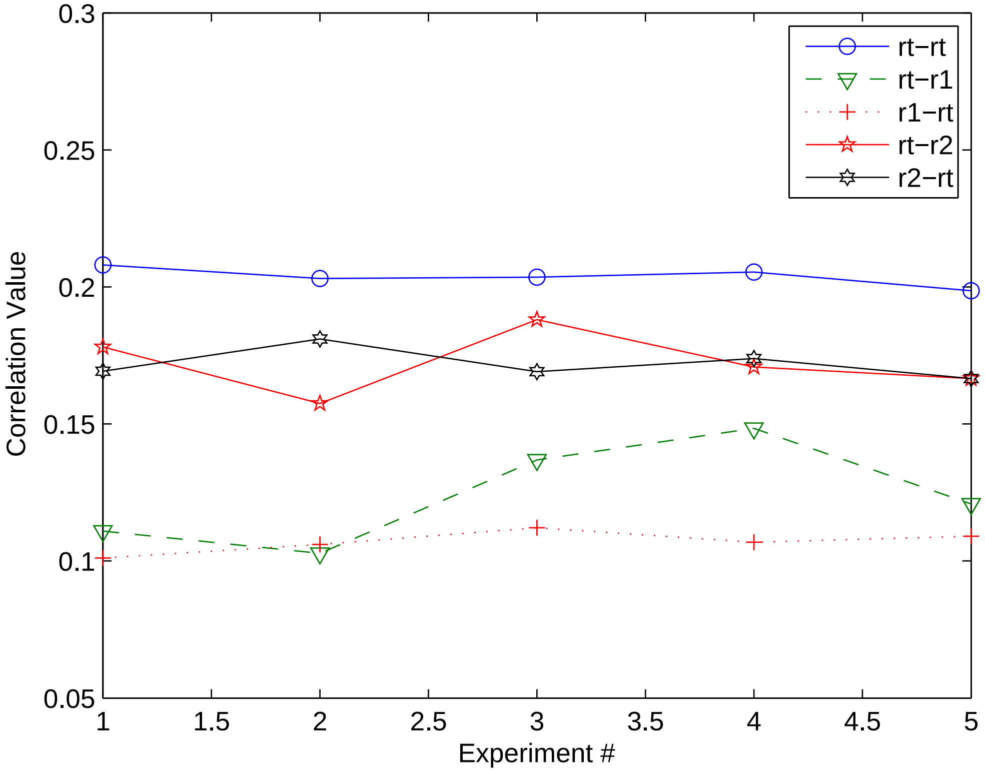

Figure 14. Correlation of 10 Combined Reviews A. We select a single review to serve as a "new" review, put it together with other 9 random reviews from the same reviewer. Then we select 10 random reviews from another reviewer and put them together. The correlation value is calculated between the two sets of reviews. In the figure we selected a random single review from reviewer r_t, r_1 and r_2 respectively, and then execute the computation. $r_t - r_1$ represent that we select a random single review from r_t and calculate correlation with reviews from r_1. The other four legend are similar. Connection labels are set to a5b1c9e1.

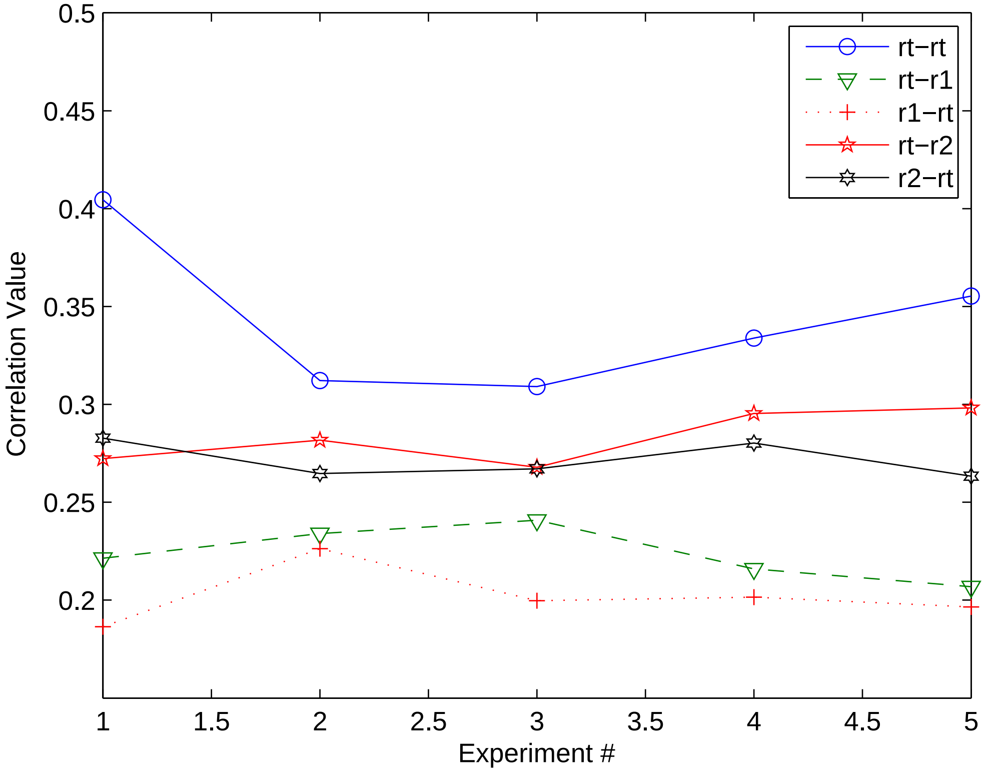

Figure 15. Correlation of 10 Combined Reviews B. We select a single review to serve as a "new" review, put it together with other 9 random reviews from the same reviewer. Then we select 10 random reviews from another reviewer and put them together. The correlation value is calculated between the two sets of reviews. In the figure we selected a random single review from reviewer r_t, r_1 and r_2 respectively, and then execute the computation. $r_t - r_1$ represent that we select a random single review from r_t and calculate correlation with reviews from r_1. The other four legend are similar. Connection labels are set to a9b1c5e1.

Since it is proved that correlation value is related to a reviewer's culture background. We will try to make connection of two culturally related reviewer without sharing information directly. The methodology is to compare a newly written review of r_m to a random amount of reviews of r_n, and calculate the average of correlation. Then compare a newly written review of r_n to a random amount of reviews of r_m, and calculate the average of correlation. We use the same reviewers r_t, r_1 and r_2

as in previous experiments. Figure 14 and 15 illustrate the results by using label assignments "a5b1c9d1" and "a9b1c5d1" respectively.

From the results, we can see the correlation of r_t and r_1 is greater than that of r_t and r_2. By having this difference among correlation values, we are able to generate 1-bit binary number by data binning. For example, in Figure 14 we can assign any correlation value greater than 0.15 to a binary number "0", and the number "1" for correlation smaller than 0.15. By calculating correlation values continuously, a unique key can be generated by repeating this process and concatenating the binary numbers.

CHAPTER V

CONCLUSIONS AND FUTURE WORKS

In this work, we have investigated the correlation of movie reviews and studied the values of different weight assignments to the sentence and word relation. From the experiment results, we conclude that "CN_SYNONYMS" is the dominant positive association that impacts correlation value, and "CN_INTRA_PHRASE" is neither a positive association nor a negative association of correlation value. When we calculate correlation between two single reviews, the results usually fluctuate in a large range. Then we put multiple reviews together and execute computation on them, the results show a significant reduction on randomness. Further exploration of reviewers on culture related factor is facilitated by the experiment results.

These correlations have then been used to evaluate and derive a unique random number. We select a specific reviewer and investigate the correlation values to other reviewers who have common movie reviews with the selected reviewer. We target a single review of those reviewers, and put it together with other reviews to obtain correlation values from different pairs of reviewers. Then the correlation value is binned to a 1-bit binary number. Through such a simplified extraction, a unique random number can be generated by repeating the process of binning. The unique random number is able to facilitate secure information exchanges between the users. In our future work, we will explore such correlations to generate a practically usable unique secret for secret keys.

REFERENCES

[1] Jörg Becker and Dominik Kuropka. Topic-based vector space model. In *Proceedings of the 6th International Conference on Business Information Systems*, pages 7–12, 2003.

[2] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199. ACM, 2000.

[3] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229. ACM, 1999.

[4] Christian Bizer, Ralf Heese, Malgorzata Mochol, Radoslaw Oldakowski, Robert Tolksdorf, and Rainer Eckstein. The impact of semantic web technologies on job recruitment processes. In *Wirtschaftsinformatik 2005*, pages 1367–1381. Springer, 2005.

[5] Pablo González Blasco, Graziela Moreto, Adriana FT Roncoletta, Marcelo R Levites, and Marco Aurelio Janaudis. Using movie clips to foster learners' reflection: improving education in the affective domain. *FAMILY MEDICINE-KANSAS CITY-*, 38(2):94, 2006.

[6] Lera Boroditsky. Does language shape thought?: Mandarin and english speakers' conceptions of time. *Cognitive psychology*, 43(1):1–22, 2001.

[7] Wenjing Duan, Bin Gu, and Andrew B Whinston. Do online reviews matter? - an empirical investigation of panel data. *Decision support systems*, 45(4):1007–1016, 2008.

[8] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[9] Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In *The Semantic Web: Semantics and Big Data*, pages 351–366. Springer, 2013.

[10] James E Gentle. *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2003.

[11] Sunil Gupta and Donald R Lehmann. Customer lifetime value and firm valuation. *Journal of Relationship Marketing*, 5(2-3):87–110, 2006.

[12] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[13] Miss Sujata S Khobragade, Mr Paritosh Sardare, Miss Bhagyashri Kumbhare, Miss Pallavi Dongre, and Mr Dipak Jha. Cryptography and network security. In *Conference Proceedings of International Conference on Advanced Computing, Communication & Networks*, pages 697–700, 2011.

[14] Manoj Kumar. Cryptography and network security. *Krishna Prakashan Media (P) Ltd*, 2007.

[15] 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. *Psychological review*, 104(2):211, 1997.

[16] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[17] George Marsaglia. Random number generation. 2003.

[18] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[19] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[20] Roger Riley, Dwayne Baker, and Carlton S Van Doren. Movie induced tourism. *Annals of tourism research*, 25(4):919–935, 1998.

[21] Caitlin Sadowski and Greg Levin. Simhash: Hash-based similarity detection. Technical report, Technical report, Google, 2007.

[22] Pascal Soucy and Guy W Mineau. Beyond tfidf weighting for text categorization in the vector space model. In *IJCAI*, volume 5, pages 1130–1135, 2005.

[23] Danny Sullivan. How search engines work. *SEARCH ENGINE WATCH, at http://www. searchenginewatch. com/webmasters/work. html (last updated June*

26, 2001)(on file with the New York University Journal of Legislation and Public Policy), 2002.

[24] Sandeep Tata and Jignesh M Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM SIGMOD Record*, 36(2):7–12, 2007.

[25] Atro Voutilainen. Part-of-speech tagging. *The Oxford handbook of computational linguistics*, pages 219–232, 2003.

[26] Wikipedia. Netflix — wikipedia, the free encyclopedia, 2015. [Online; accessed 18-April-2015].

[27] SK Michael Wong and Vijay V Raghavan. Vector space model of information retrieval: a reevaluation. In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 167–185. British Computer Society, 1984.

[28] Zhang Huanjiong Wang Guosheng Zhong Yixin. Text similarity computing based on hamming distance [j]. *Computer Engineering and Applications*, 19:007, 2001.