# MINING SEMANTIC RELATIONSHIPS BETWEEN CONCEPTS ACROSS DOCUMENTS USING WIKIPEDIA KNOWLEDGE

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Peng Yan

In Partial Fulfillment
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

September 2013

Fargo, North Dakota

North Dakota State University
Graduate School

**Title**

Mining Semantic Relationships between Concepts across Documents using Wikipedia Knowledge

**By**

Peng Yan

The Supervisory Committee certifies that this ***disquisition*** complies with

North Dakota State University's regulations and meets the accepted standards

for the degree of

**DOCTOR OF PHILOSOPHY**

SUPERVISORY COMMITTEE:

Wei Jin

Chair

Brian M. Slator

Juan Li

Ying Huang

Approved:

| 11/01/2013 | Brian M. Slator |
| --- | --- |
| Date | Department Chair |

# ABSTRACT

The ongoing astounding growth of text data has created an enormous need for fast and efficient Text Mining algorithms. However, the sparsity and high dimensionality of text data present great challenges for representing the semantics of natural language text. Traditional approaches for document representation are mostly based on the Vector Space (VSM) Model which takes a document as an unordered collection of words and only document-level statistical information is recorded (e.g., document frequency, inverse document frequency). Due to the lack of capturing semantics in texts, for certain tasks, especially fine-grained information discovery applications, such as mining relationships between concepts, VSM demonstrates its inherent limitations because of its rationale for computing relatedness between words only based on the statistical information collected from documents themselves. In this dissertation, we present a new framework that attempts to address the above problems by utilizing background knowledge to provide a better semantic representation of any text. This is accomplished through leveraging Wikipedia, the world's currently largest human built encyclopedia. Meanwhile, this integration also sufficiently complements the existing information contained in text corpus and facilitates the construction of a more comprehensive representation and retrieval framework.

Specifically, we present 1) Semantic Path Chaining (*SPC*), a new text mining model that automatically discovers semantic relationships between concepts across multiple documents (which the traditional search paradigm such as search engines cannot help much) and effectively integrates various evidence sources from Wikipedia; 2) the kernel methods that provide a more appropriate estimation of semantic relatedness between concepts and better utilize Wikipedia background knowledge in our defined query contexts; 3) Concept Association Graph (*CAG*), a

graph-based mining prototype system interfaced directly to Wikipedia, enables fast and customizable concept relationship search using Wikipedia resources.

The effectiveness of the proposed techniques has been evaluated on different data sets. The experimental results demonstrate the search performance has been significantly enhanced in terms of accuracy and coverage compared with several baseline models. In particular, some existing state-of-the-art related work such as Srinivasan's closed text mining algorithm, Explicit Semantic Analysis (ESA) [19] and the RelFinder system [26, 27, 41] has been used as the comparison models.

# ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Wei Jin, who offered me an opportunity to come to the U.S. to pursue my PH.D study, introduced me to the research area of text mining, and gave excellent guidance and enormous support to my research. It was Dr. Jin who helped me out of numerous research difficulties, enlightened me with constructive advice, and guided me to move forward along the right track during my research. I have been always feeling that it is my great luck to have such a nice advisor who is strict towards my research in school but treats me as a friend in normal times. Without her guidance, I would never be able to complete my PH.D work.

I am also very grateful to Dr. Brian Slator, who brought me to the research field of immersive virtual environments, and has been supporting my study over the past 17 months. Because of Dr. Slator, I was able to continue my study, and had a chance to apply one of the models proposed in this dissertation into the real world software industry. I was honored to be a member in his development team and work with so many talented engineers

I wish to express my love and gratitude to my beloved brother, Fei Yan, who works hard in China and takes almost all of the responsibilities for my parents that I should have taken, so that I can focus on my study. If it had not been for supporting my life in the U.S., he would have had a car that he had craved for two years. No matter how far we are away from each other, I want to let him know my heart has always been with him.

Also, I want to thank my parents, who pushed the longing of wanting me to go back and stay with them down into the dim recesses of their minds, understood and supported me in all moments of my life.

I would also like to thank my wife, Shuhui Ren, who sacrificed her personal career and life in China, and came here to take care of me. We lived a hard life together due to the financial embarrassment, but she had never complained to me and always unconditionally supported me for the past two years. I would never forget those numerous late nights she accompanied me either in school or at home for catching up with approaching conference deadlines. My current achievement is at too much cost of her interests.

Deepest gratitude is also due to members of my committee, Dr. Juan Li, Dr. Ying Huang without whose assistance, this study would not have been completed.

# DEDICATION

I dedicate this dissertation to my brother, my parents and my wife.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1. INTRODUCTION

Text is the most traditional method for information recording and knowledge representation. Text mining focuses on mining high-quality information from mass text. However, great challenges have been posed for many text mining tasks because of the increasing sheer volume of text data and the difficulty of capturing valuable knowledge hidden in them. Therefore efficient and high-quality text mining algorithms are demanded and effective document representation and accurate semantic relatedness estimation become increasingly crucial.

Mining semantic relationships/associations between concepts from text is important for inferring new knowledge and detecting new trends. More commonly, text documents are represented as a Bag of Words (BOW) and semantic relatedness between words is measured by statistical information gathered from the corpus such as term frequency (TF), inverse document frequency (IDF), and the widely used cosine similarity weighting scheme, referred to as Vector Space Model (VSM) [30, 33, 71]. Clearly, this is a considerable oversimplification of the problem because a lot of the semantics in a document is lost when just replacing its text with a set of words, such as the order of terms and the frontiers between sentences or paragraphs. While entities could be treated as terms and represented by index representation, the correlation between entities is lost. Due to the lack of capturing semantics in texts, for certain tasks, especially fine-grained information discovery applications, such as mining relationships between concepts, VSM demonstrates its inherent limitations. In this dissertation, we present a new framework that attempts to address the above problems by utilizing background knowledge to provide a better semantic representation of any text and a more appropriate estimation of semantic relatedness between terms. This is accomplished through leveraging Wikipedia, the

world's currently largest human built encyclopedia. Meanwhile, this integration also sufficiently complements the existing information contained in text corpus and facilitates the construction of a more comprehensive representation and retrieval framework.

### 1.1. Problem Description and Motivation

Specifically, we present Semantic Path Chaining (*SPC*), a new text mining model that automatically discovers semantic relationships between concepts across multiple documents and effectively integrates various evidence sources from Wikipedia. The proposal of this query scenario is based on the observation that few works (tools) are link aware when searching text documents over the Internet or general document collections and integration of information from multiple interrelated units may be more useful in answering users' information needs, especially when there is very little data/information available about entities of interest, and thus their relationships are not clear and need exploration. A traditional search involving, for example, two person names will attempt to find documents mentioning both of these individuals. In the event that there are no pages containing both names, no documents are returned or just documents with one of the names ranked by relevancy. Even if two or more interrelated documents contain both names, the existing search tools cannot integrate information into one relevant and meaningful answer. The goal of this research is to explore automated solutions to sift through these extensive document collections and automatically discover these significant but may be unapparent links. In addition, we propose to leverage Wikipedia, the largest encyclopedia in existence, to augment text representation and complement existing knowledge in document collections. Under this context, any texts can be represented as a weighted mixture of a predetermined set of natural concepts from Wikipedia, which are provided by humans themselves and can be easily explained.

Compared with other existing solutions for addressing similar tasks with no or little background knowledge taken into account [8, 13, 30, 31, 33, 65, 70, 71, 75], this proposed approach has shown the significant advantage in providing more comprehensive and accurate solutions through the use of vast amounts of highly organized human knowledge encoded in Wikipedia. We believe this integration will have impact far beyond the proposed context and benefit a wide range of natural language processing applications in need of large scale world knowledge.

Formally, a Semantic Path Chaining (*SPC*) problem is defined as follows. Given a query, involving concepts A and B, it has the following meaning: find the most plausible relationships between concept A and concept B assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. Furthermore, if no relationships are identified in the existing document collection, is there a connection between A and B that can be discovered from the Wikipedia knowledge base? The query output takes the form of chains of entities, as in $A \rightarrow C1 \rightarrow C2 \rightarrow ,\ldots, \rightarrow B$ in this example, each relating to and connecting to other concepts in the chain that partially answer the user's information need. Figure 1.1 shows an example for the relationship query involving "*Bin Ladin*" and "*Omar Abdel Rahman*". The identified connection between them is *Bin Ladin* → *Al Qaeda* → *Abdullah Yusuf Azzam* → *Omar Abdel Rahman*. In this example, discovered links may tell a story that "*Bin Ladin*" who inspired the September 11 attacks founded "*Al Qaeda*"; "*Abdullah Yusuf Azzam*", an Islamic scholar and theologian, was also one of the founders of "*Al Qaeda*"; He was the professor of "*Omar Abdel Rahman*" who "*built a strong rapport with bin Laden during the Soviet war in Afghanistan*". Note that this chain cannot be discovered by solely relying on the existing text corpus (i.e., the publicly available 9/11 commission report, the dataset we used for this research). All the related links discovered can only be acquired from analyzing Wikipedia knowledge.

Paragraph 1:
***Osama bin Mohammed bin Awad bin Laden*** was the founder of ***al-Qaeda***, the Sunni militant Islamist organization that claimed responsibility for the September 11 attacks on the United States, along with numerous other mass-casualty attacks against civilian and military targets.

Paragraph 2:
In 1989, after the Soviets pulled out of Afghanistan, ***Azzam*** and his deputy ***Osama bin Laden*** decided to keep their movement permanent and founded the ***Al Qaeda***.
……
However, it was reported that ***Bin Laden*** and ***Azzam*** also had a major dispute on where ***Al Qaeda*** should focus their operations
……
***Azzam*** is thought to had influence on jihadists such as ***al-Qaeda*** with the third stage of his "four-stage process of jihad"

Paragraph 3:
Although ***Abdel-Rahman*** was not convicted of conspiracy in the Sadat assassination, he was expelled from Egypt following his acquittal. He made his way to Afghanistan in the mid-1980s where he contacted his former professor, ***Abdullah Azzam***, co-founder of Maktab al-Khadamat (MAK) along with ***Osama bin Laden***.
……
***Rahman*** built a strong rapport with ***bin Laden*** during the Soviet war in Afghanistan and following ***Azzam***'s murder in 1989 ***Rahman*** assumed control of the international jihadists arm of MAK/***Al Qaeda***.

Figure 1.1. Evidence for the Associations Discovered between "*Bin Ladin*" and "*Omar Abdel Rahman*" Using Wikipedia Articles: "*Osama bin Laden*", "*Abdullah Yusuf Azzam*" and "*Omar Abdel-Rahman*"

## 1.2. Research Contributions

The contributions of this work can be summarized as follows:

1) A new Wiki-enabled cross-document knowledge discovery framework has been proposed and implemented which effectively complements the existing information contained in the document collection and provides a more comprehensive knowledge representation and mining framework supporting various query scenarios. In

particular, over 5,000,000 Wikipedia articles and more than 700,000 Wikipedia categories are considered.

2) Effective noise filtering techniques specifically tailored to meet our needs are provided. A series of heuristic strategies to remove noisy evidence from multiple Wikipedia articles has been designed to increase the reliability of the overall knowledge encoded.
3) A solution for alleviating semantic loss of the Vector Space Model (VSM) has been presented through incorporating background knowledge from Wikipedia. A better estimation of semantic relatedness is provided by combining various evidences from Wikipedia such as article content, associated categories, and anchor texts. Various kernel methods are designed to achieve this goal.
4) The proposed approach also tackles the limitations of traditional Bag-of-Words representation which only considers the terms appearing in the text, thus failing to identify highly related terms that may not co-occur literally with entities of interest in the text corpus. Note that through considering relevant terms spanning all Wikipedia articles, the space of concepts and relationships considered now in our solution is not limited to those present in the document collection. A high-dimensional space of natural concepts derived from Wikipedia can be incorporated.
5) While the problem addressed in this dissertation focuses on relationship discovery, the proposed semantic relatedness estimation model based on various kernel methods can be easily adapted and deployed in solving other important problems such as classification and clustering.

6) A distributed solution based on the Map-Reduce framework has been developed which shows its potential in efficiently processing large scale Wikipedia data. The "Map" and "Reduce" functions suited to our task have been designed and integrated.

### 1.3. System Overview

The architecture of the proposed system is illustrated in Figure 1.2. There are 3 modules in the system.

1) Semantic Path Chaining (*SPC*) Module: this module receives two topics of interest as input and automatically generates concept chains using the *SPC* method proposed in this dissertation. There are four sub-modules described as follows
   - Semantic Profile Generation Module: this module receives two topics of interest, conducts independent search against the documents, and builds semantic profiles containing topic relevant concepts.
   - Wikipedia Knowledge Preparation Module: this module goes through the profiles built from the first module, and selects relevant Wikipedia articles and categories from the Wikipedia database.
   - Concept Weighting Module: this module employs two different weighting schemes proposed in this dissertation to calculate the semantic relatedness between concepts.
   - Feature Enrichment Module: this module enriches the discovered relationships using concepts derived from Wikipedia.

2) Concept Association Graph Module: this module receives two topics of interest along with user specified constraints and builds a Concept Association Graph (*CAG*) for discovering relationship chains. There are five sub-modules shown below:

- Query Construction Module: this module receives two search topics along with user specified constraints (e.g., the similarity threshold used for concept pruning), encapsulates them within one query.
- Wikipedia Knowledge Preparation Module: this module is similar to the counterpart module in the *SPC* module except that here it selects topic-relevant anchor texts instead of categories from Wikipedia as potential nodes for *CAG* construction.
- Knowledge Representation Module: responsible for representing topic relevant knowledge received from the Knowledge Preparation module.
- Graph Search Module: responsible for detecting potential relationships between the two given topics on top of the constructed *CAG*.
- Result Visualization Module: this module visualizes the entire discovery process in real time and gives a graphical view of the discovered relationships.

3) Concept Chain Persistence Module: responsible for caching the returned search results for efficient future retrieval.

The entire relationship discovery process is triggered by receiving two topics of interest input by the user, and completed automatically by returning possible concept chains connecting the given topics.

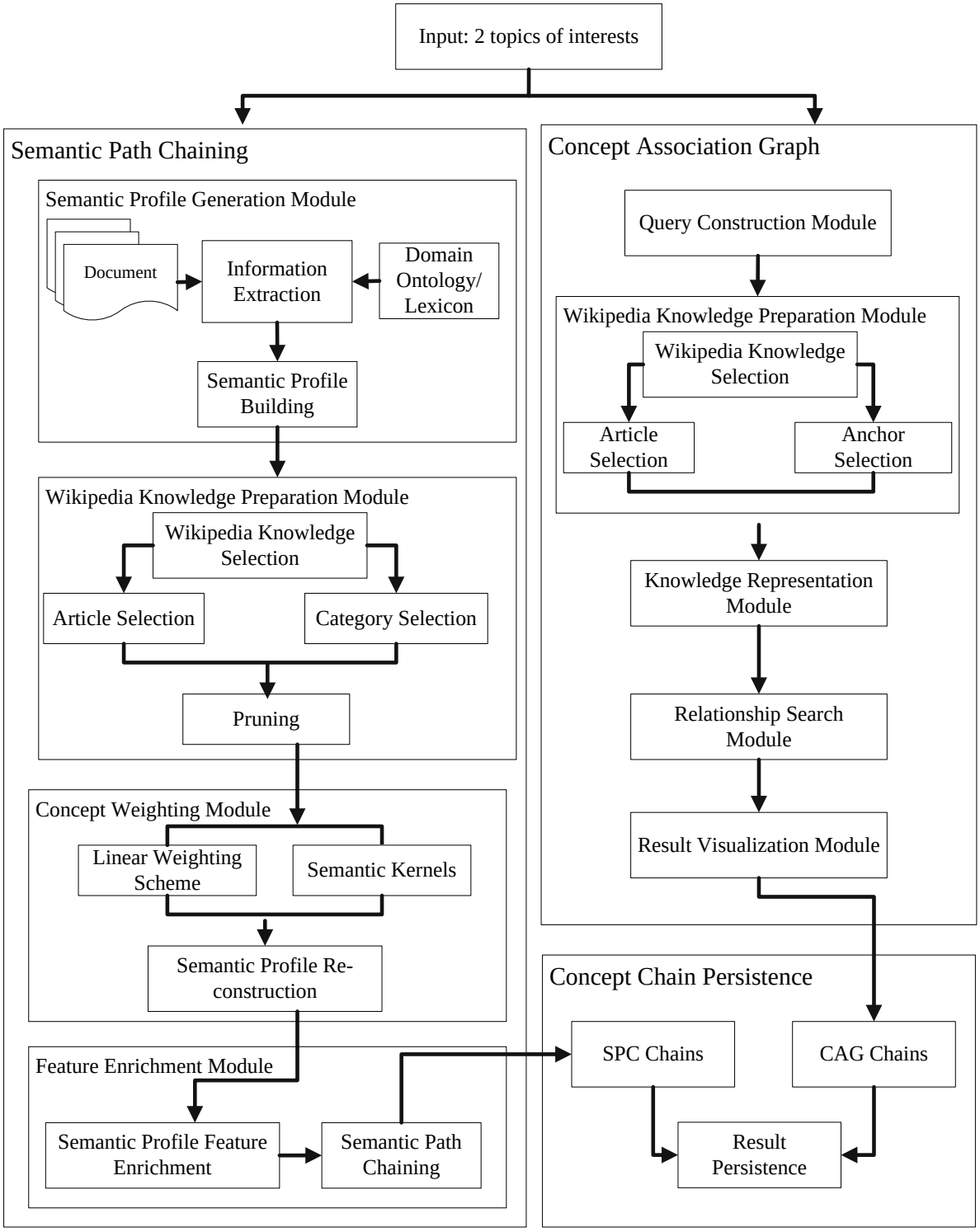


Figure 1.2. System Architecture

## 1.4. Organization of the Dissertation

The remainder of the dissertation is organized as follows. Chapter 2 describes related work. Chapter 3 introduces the Semantic Path Chaining (*SPC*) model for discovering semantic relationships between two given topics. Chapter 4 discusses the kernel methods in detail. Chapter 5 discusses the Concept Association Graph (*CAG*) mining model. Experimental results are presented and analyzed in Chapter 6. Chapter 7 concludes this dissertation and describes future directions.

# CHAPTER 2. LITERATURE REVIEW

In this chapter, we introduce background knowledge in text mining. We start with the Vector Space Model, one of the most widely used models in information retrieval systems which represents text documents as vectors of identifiers.

## 2.1. Vector Space Model

The Vector Space Model (VSM) (also called Bag of Words (BOW) model), developed by Salton and his group at Cornell [60], is an algebraic model for document or query representation in a n-dimensional vector where each dimension corresponds to a term appearing in the document or query.

### 2.1.1. TFIDF Weighting Scheme

The VSM makes two assumptions to define the importance of each term:

1) The frequency of a word in a document is related to the importance of this word in the document.
2) The importance of a word in a document is independent from the importance of other words.

According to the assumptions above, the order of terms in a document is trivial. The importance of each term is represented by a weight, as known as term weight and can be calculated in different weighting schemes. One of the best known schemes is called TFIDF weighting scheme containing two components. Supposed a document $d_j$ in a collection is defined as follows:

$$d_j = (t_{1,j}, t_{2,j}, ..., t_{n,j}) \quad (2.1)$$

Where $t_{i,j}$ ($i=\{1,2,\ldots,n\}$) is a term occurring in the document $d_j$.

The two components for the TFIDF weighting scheme are defined as below:

a) Term Frequency ($tf_{i,j}$): the number of occurrences of term $i$ in document $j$.

b) Inverted Document Frequency ($idf_{i,j}$): $idf_{i,j} = \log(|D|/|d' \in D \mid t_{i,j} \in d'|)$ where $D$ is the total number of documents in the document set. The Inverted Document Frequency is introduced to attenuating the effect of terms that occur too often in the collection to be meaningful for relevance measurement, since in many cases, rare terms are more informative than frequent terms and thus should be given larger weights.

Using the TFIDF weighting scheme, a document $d_j$ in a collection is represented by a vector spanned by $s$ number of concepts in the collection:

$$d_j = [tf_{1,j} \cdot idf_{1,j}, tf_{2,j} \cdot idf_{2,j}, ..., tf_{s,j} \cdot idf_{s,j}] \tag{2.2}$$

Then the similarity between documents $d_i$ and $d_j$ can be computed by comparing the deviation of angles between their corresponding document vectors. In practice, the cosine value of the angle between the two vectors is usually used to represent the similarity:

$$sim(d_i, d_j) = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{|\vec{d}_i||\vec{d}_j|} = \frac{\sum_{m=1}^{s} d_{i,m} d_{j,m}}{\sqrt{\sum_{m=1}^{s} d_{i,m}^2}\sqrt{\sum_{m=1}^{s} d_{j,m}^2}} \tag{2.3}$$

Where $d_{i,m}$ is the TFIDF value of term $m$ in document $d_i$, and $d_{j,m}$ is the TFIDF value of term $m$ in document $d_j$.

Besides computing the similarity between two documents, the VSM is also widely used for measuring the relevance of a document for a given query. Suppose a query is defined as below:

$$q = (t_1, t_2, ..., t_n) \tag{2.4}$$

Where $t_i$ is a term occurring in the query $q$, and the weight of each term in the query is defined as follows:

$$weight(t_i) = tf(t_i) \cdot idf(t_i) \quad (2.5)$$

Where $tf(t_i) = \begin{cases} 1 & \text{if } t_i \text{ occurs in the query} \\ 0 & \text{otherwise} \end{cases}$ and $idf(t_i) = \begin{cases} idf(t_i) \text{ in the given documents} & \text{if } t_i \text{ occurs in the query} \\ 0 & \text{otherwise} \end{cases}$.

Then the similarity between a document and a query can also be measured through the cosine value between their corresponding term vectors. Using this similarity, all documents in a collection can be ranked according to their relevance to the query.

### 2.1.2. Limitations of Vector Space Model

Although applied in many research fields such as document classification and clustering, knowledge discovery in text, the VSM has major inherent limitations in the following aspects.

a) Omission of the semantic content of words: there is no semantic interpretation of words in VSM. Documents are represented by a bag of words where the semantic content of the words is omitted.

b) Intuitive Weighting Scheme: the weighting scheme used in VSM is intuitive since the semantic relatedness between words is only measured based on the statistical information collected from documents themselves.

## 2.2. VSM-based Approaches

### 2.2.1. Cross-Document Coreference

Cross-document information analysis aims at breaking the document boundary through analyzing information from multiple text resources. Coreference analysis refers to the process of determining whether or not two mentions of entities refer to the same person [37]. Cross-

document coreference analysis goes one step further to consider whether the mentions of a name in multiple text sources are the same. Bagga and Baldwin [5] developed a cross-document coreference resolution algorithm based on the VSM to perform disambiguation for people having the same name.

For example, in Figure 2.1, a simple cross-document coreference analysis task attempts to decide whether the three mentions of "*Perry*" in *Document* 1 and *Document* 2 refer to the same person.

**Document 1:**
***John Perry***, of Weston Golf Club, announced his resignation yesterday. ***He*** was the President of the Massachusetts Golf Association. During his two years in office, ***Perry*** guided the MGA into a closer relationship with the Women's Golf Association of Massachusetts.

**Document 2:**
Oliver "Biff" Kelly of Weymouth succeeds ***John Perry*** as president of the Massachusetts Golf Association. "We will have continued growth in the future," said Kelly, who will serve for two years. "There's been a lot of changes and there will be continued changes as we head into the year 2000."

Figure 2.1. Cross-Document Coreference Example [Bagga et al, 1998]

The algorithm [5] is composed of three steps:

1) Receive coreference related documents and generate coreference chains (e.g. in Figure 2.1, a typical coreference chain in Document 1 is: *John Perry→He→Perry*).
2) For a particular entity of interest (e.g. *John Perry*), extract all sentences containing the noun phrases which form the coreference chain (e.g. "*John Perry→He→Perry*") in each document as a document summary, e.g. the summary for *Document* 1 in Figure 2.1 is composed of three sentences respectively:

- Sentence 1: ***John Perry****, of Weston Golf Club, announced his resignation yesterday*.
- Sentence 2: ***He*** *was the President of the Massachusetts Golf Association*.
- Sentence 3: *During his two years in office,* ***Perry*** *guided the MGA into a closer relationship with the Women's Golf Association of Massachusetts*.

3) Compute the similarity between two document summaries using the VSM. Two summaries are considered to be regarding the same entity if their similarity is above a certain threshold.

However, this approach computes the similarity between two entities of interest only based on their corresponding summaries. It is very likely to view two entities as the same if their summaries have most of the same words but the corresponding documents differ a lot. Also, their approach was only evaluated on a small corpus of documents and did not demonstrate the effectiveness on large scale corpora. Actually, Chung and James have shown that this approach does not work well when translated to a substantially larger corpus of documents [21]. [21] developed a much larger and more ambiguous corpus called "*Person-x Corpus*", and used a variation of the VSM to provide improved results. However, their proposed approach is still a VSM-based approach and inevitably inherits the limitations of VSM.

### 2.2.2. Open and Closed Discovery Algorithm

Built within the discovery framework established by Swanson and Smalheiser [71], Srinivasan proposed the open and closed text mining algorithm [67] to automatically discover interesting concepts from MEDLINE.

The open discovery algorithm starts from a source topic and searches intermediate links to reach several destination topics. As shown on the left of Figure 2.2, given a topic A of interest, there may be an interesting indirect relationship between A and C1, A and C2, etc., via the linking B terms. The algorithm first limits its search within linking terms e.g. B1, B2, etc. and then starts from each intermediate term and looks for topics represented by terms C1, C2, etc. In comparison to an A to B to C search, the closed discovery algorithm illustrated on the right of Figure 2.2 takes as input two topics of interest (e.g. topic A and C), and then starts bidirectional search at the same time from both A and C, looking for novel and meaningful interlinking B terms. Swason actually confirmed his hypothesis that there might be an association between Raynaud's disease (topic A) and fish oils [69] (topic C) using the similar way to the closed discovery algorithm [67]. Their work drew attention from other researchers to use this approach such as [74] a few years after Swason's discovery.

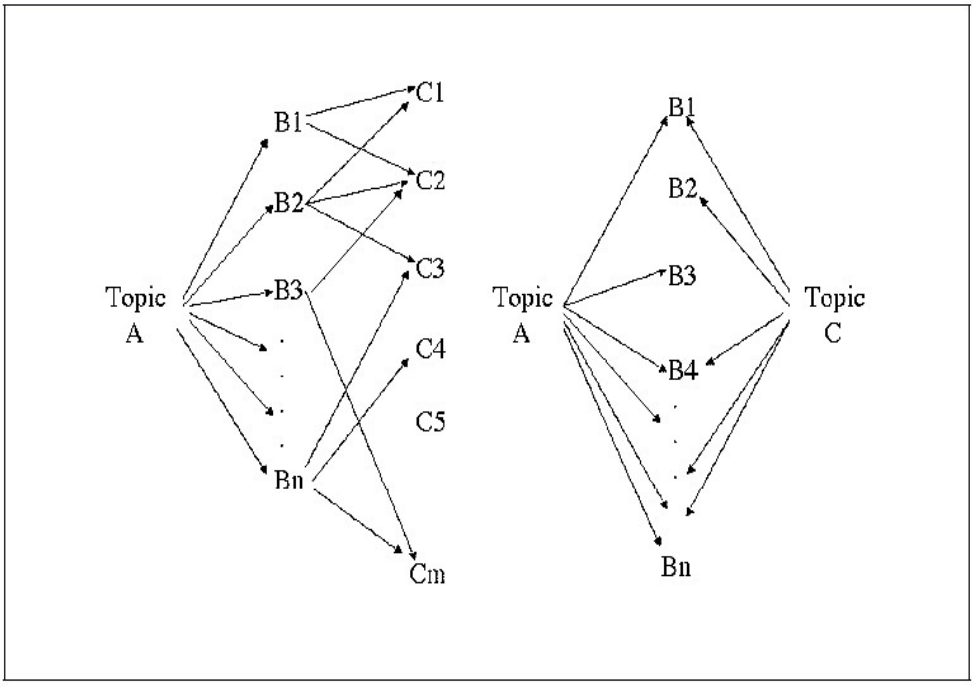


Figure 2.2. Indirect Concept Links [Srinivasan 2004]

The *SPC* model proposed in this dissertation is motivated by the closed discovery algorithm. We extend this algorithm by considering multiple levels of intermediate profiles, and use sentence-level co-occurrence as a way to calculate the semantic relatedness between concepts. The most important improvement against the closed discovery algorithm is that we successfully integrate background knowledge derived from Wikipedia into the knowledge discovery process, and thus we are able to provide a better modeling of concept representation and semantic relatedness estimation.

### 2.2.3. Concept Chain Queries

Concept Chain Queries (CCQ), defined in [30, 34], was based on the hypotheses that the wealth of recorded knowledge is greater than the sum of its parts [11], and designed to discover novel links between two topics of interest (e.g., two person names) across multiple documents CCQ can capture salient information [44] among documents. It goes beyond what traditional IR models handle [4, 30, 34]. A concept chain query involving concept A and concept B intends to find the best path linking topic A to topic C. The paths found stand for potential conceptual connections between them. Typically, the uncovered links involving concepts A and B have the following meaning: find the most plausible relationship between concept A and concept B assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. However, the techniques proposed in [30, 33] were all built under the assumption of VSM representation without background knowledge incorporated, and thus demonstrating their inherent limitations. For example, *Ziyad Khaleel*, also known as *Khalil Ziyad*, was a Palestinian-American al-Qaeda member, based in the United States, being identified as a "*procurement agent"* for *Bin Ladin*'s terroristic organization. Clearly he has a close relationship

with *Bin Ladin*. Nevertheless, he will not be taken into consideration if his name does not appear in the document collection where the concept chain queries are performed. In other words, any concepts will be considered irrelevant to the topic of interest, unless they both occur in the text literally. While in our *SPC* model, various information resources from Wikipedia is incorporated to complement the existing knowledge contained in the documents, and the relationships discovered do not necessarily appear in the documents literally. Therefore, the search results are enriched with relevant Wikipedia concepts where concept closeness is measured by referring to various evidence sources from Wikipedia.

## 2.3. Web Oriented Approaches for Measuring Semantic Relatedness

### 2.3.1. Web Page Counts Oriented Approach

Serving as an integral part of information retrieval and natural language processing, semantic similarity between words has gained increasing attention over the past years. Since similarity between entities may change during the march of time, making use of up-to-date resources to assist in similarity computing tasks is promising such as the web resources [35, 43, 56]. Bollegala [7]developed an automatic method for semantic similarity calculation using returned page counts and text snippets generated by a Web search engine. Specifically, they modified four popular co-occurrence measures to compute similarity using page counts (page-count-based similarities), and developed a pattern extraction algorithm to extract lexico-syntactic patterns that indicate various aspects of semantic similarity, such as the "*is a*" semantic relationship. The process of extracting patterns from text snippets is illustrated in Figure 2.3.

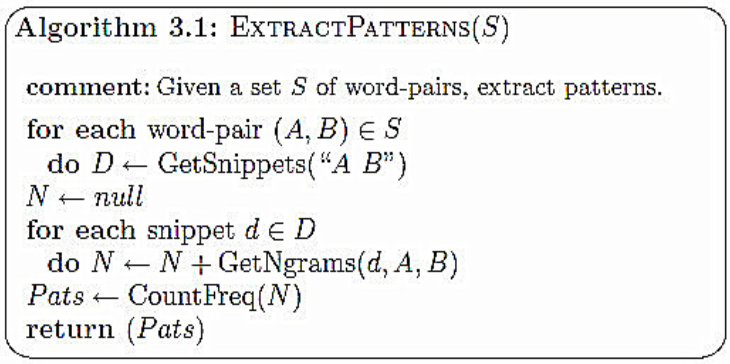


Figure 2.3. Extract Patterns from Snippets [Bollegara, 2007]

Suppose *S* is a set containing synonymous word-pairs, *A* and *B* are two terms forming a word pair, the algorithm first gets all text snippets for the query *A* and *B* returned by search engines, and then extracts all *n*-grams for *n* = 2,3,4,5. Next those *n*-grams containing exactly one *A* and one *B* are selected as candidate patterns for representing the semantic relationship between *A* and *B*. At last the frequency of each candidate pattern is calculated to represent the goodness of the pattern. To form a final weighting scheme for measuring semantic similarity between words, they created a feature vector by merging the pattern frequency and the four page-count-based similarities, then used this vector to calculate a final similarity between two words.

### 2.3.2. Related Terms Oriented Approach

Semantic relatedness indicates the degree to which words are associated via any type (such as synonymy, meronymy, hyponymy, hypernymy, functional, associative and other types) of semantic relationships [59], while semantic similarity restrict the relations between words to hyponymy/hypernymy. Schütze et al created a thesaurus from the local document collection to perform semantic information retrieval [61]. Reznik [57] provides the widely used example of *car* and *gasoline* to illustrate the difference between relatedness and similarity. Much work has been done to measure the semantic relatedness between words in different ways. Since words

that are very different in semantic meaning may imply a certain relation when applied in a specific functional context such as the *cars* and *gasoline* example (*cars* use *gasoline*), one method is to compute the semantic relatedness between words using the words related to them. Salahli [59] proposed a method that calculated semantic relatedness between terms using a set of determiners (special words that are highly related to terms of interest). Suppose $D_1 = \{d_{11}, d_{12}, d_{13}..., d_{1n}\}$ and $D_2 = \{d_{21}, d_{22}, d_{23}..., d_{2m}\}$ are two sets containing determiner/related words to $W_1$ and $W_2$, they first combined the two sets to form a set of determiner words: $D = \{d_1, d_2, d_3, ..., d_k\}$, then calculated the semantic relatedness between $W_1$ and each determiner word $d_i$ as follows:

$$rel(d_i, W_1) = \text{freq}(d_i, W_1) / \text{maxfreq}_1 \tag{2.6}$$

Where $\text{freq}(d_i, W_1)$ is the number of pages where $d_i$ and $W_1$ co-occur, and $\text{maxfreq}_1 = \max\{rel(d_1, W_1), rel(d_2, W_1), ..., rel(d_k, W_1)\}$). Then they calculated the relatedness between $W_1$ and $W_2$ as follows:

$$rel(W_1, W_2) = (\sum_{i=1}^{k} (\frac{\alpha_i R_i}{1 + R_i}) + syn) / (1 + syn) \tag{2.7}$$

Where $R_i$, $\alpha_i$ and $syn$ are defined as:

$$R_i = \frac{\min\{rel(d_i, W_1), rel(d_i, W_2)\}}{\max\{rel(d_i, W_1), rel(d_i, W_2)\}} \tag{2.8}$$

$$\alpha_i = \begin{cases} 2 & \text{d}_\text{i} \text{ occurs in both } \text{D}_1 \text{ and } \text{D}_2 \\ 1 & \text{otherwise} \end{cases} \tag{2.9}$$

$$syn = \begin{cases} 1 & \text{W}_1 \text{ and } \text{W}_2 \text{ are synonymy or nearly synonymy} \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

Resnik [57] claimed that both WordNet and Wikipedia were used as information resources to implement the proposed approach. However, a major problem for this approach is the difficulty of choosing determiner words, and they did not give much information about how they extracted intended information from WordNet and Wikipedia, which part of data was used in their approach, and how they used the extracted information to generate determiner words. Also, the evaluation of the proposed approach was not sufficient, since only a small number of word pairs were shown in the results. Therefore, in this dissertation, we do not classify this approach as WordNet or Wikipedia-based approach (the author also claimed their approach belongs to the web page counting-based measurement method [59]).The proposed models in our work has successfully addressed the above mentioned problem by automatically going through the Wikipedia space to search for topic-related concepts and computing the semantic relatedness between the relevant concepts and the input topic.

## 2.4. WordNet-based Approaches

### 2.4.1. Introduction to WordNet

WordNet is a lexical database for the English language [46]. The main contributions of WordNet to data mining tasks are: first, it provides short and general definitions for named entities; second, it records the various semantic relations between the constructed synonym sets. The goal of WordNet is twofold: to combine dictionaries and thesaurus so they are more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

As of December 2006, the database of WordNet 3.0 had incorporated 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs. WordNet distinguishes between nouns, verbs, adjectives and adverbs, and groups them into sets of synonyms called

synsets. One synset might be connected to other synset via a number of semantic relations. Figure 2.4 shows the relations constructed in WordNet. Note that hypernymy exists inside both nouns and verbs. For example, for nouns, *X*, *Y*, *Y* is a hypernym of *X* if every *X* is a (kind of) *Y*; and for verbs, *X*, *Y*, the verb *Y* is a hypernym of the verb *X* if the activity *X* is a (kind of) *Y*, e.g. *to perceive* is a hypernym of *to listen*.

- Nouns
  - *hypernyms*: *Y* is a hypernym of *X* if every *X* is a (kind of) *Y* (*canine* is a hypernym of *dog*)
  - *hyponyms*: *Y* is a hyponym of *X* if every *Y* is a (kind of) *X* (*dog* is a hyponym of *canine*)
  - *coordinate terms*: *Y* is a coordinate term of *X* if *X* and *Y* share a hypernym (*wolf* is a coordinate term of *dog*, and *dog* is a coordinate term of *wolf*)
  - *holonym*: *Y* is a holonym of *X* if *X* is a part of *Y* (*building* is a holonym of *window*)
  - *meronym*: *Y* is a meronym of *X* if *Y* is a part of *X* (*window* is a meronym of *building*)
- Verbs
  - *hypernym*: the verb *Y* is a hypernym of the verb *X* if the activity *X* is a (kind of) *Y* (*to perceive* is an hypernym of *to listen*)
  - *troponym*: the verb *Y* is a troponym of the verb *X* if the activity *Y* is doing *X* in some manner (*to lisp* is a troponym of *to talk*)
  - *entailment*: the verb *Y* is entailed by *X* if by doing *X* you must be doing *Y* (*to sleep* is entailed by *to snore*)
  - *coordinate terms*: those verbs sharing a common hypernym (*to lisp* and *to yell*)
- Adjectives
  - *related nouns*
  - *similar to*
  - *participle of verb*
- Adverbs
  - *root adjectives*

Figure 2.4. Relations Defined in WordNet [Wikipedia, February 2013]

WordNet has been used for a number of different purposes in information systems such as text retrieval [22], document clustering [10, 36] and document categorization [8, 58]. In the following section, we will discuss several approaches using WordNet for improving knowledge representation.

### 2.4.2. WordNet for Knowledge Representation

Related work attempting to overcome the limitations of VSM and integrate background knowledge into text representation has also been reported in categorization and knowledge discovery applications. Hotho et al. [28] exploited WordNet to improve the VSM text

representation for document clustering. Specifically, they enriched VSM-based term vectors with concepts derived from the core ontology in WordNet. The employment of ontology has two benefits: first, synonyms are resolved; second, non-related terms can be related to each other if they belong to the same parent. For example, the relationship between "*beef*" and "*pork*" can be revealed by incorporating the concept "*meat*" generated from WordNet into the VSM representation, since both "*beef*" and "*pork*" are sub-concepts of "*meat*" according to the relations defined in WordNet. Three different strategies were proposed for document representation. Suppose a term vector using the VSM is represented as $\vec{t}_d$, and $\vec{c}_d$ is the concept vector containing the terms' corresponding concepts derived from WordNet, the three strategies are defined as below:

- Strategy 1: add all concepts in $\vec{c}_d$ to the term vector $\vec{t}_d$, i.e. $\vec{t}_d$ is replaced by the concatenation of $\vec{t}_d$ and $\vec{c}_d$. After the concatenation, a term in previous $\vec{t}_d$ that also appeared in Wordnet as a member of the synset would be accounted for at least twice in the new vector representation.
- Strategy 2: replace terms by concepts. If a term in $\vec{t}_d$ also appears in $\vec{c}_d$, it is expelled by the corresponding concept in $\vec{c}_d$.
- Strategy 3: concept vector only. This strategy discards all terms in $\vec{t}_d$, and uses only $\vec{c}_d$ to represent a document.

They also developed different strategies for word sense disambiguation, as well as hypernym processing. Then the similarity between two documents $d_1$ and $d_2$ is computed using their corresponding term vectors $\vec{t}_1$ and $\vec{t}_2$ through the cosine of the angle between $\vec{t}_1$ and $\vec{t}_2$ :

$$\cos(\measuredangle(\vec{t}_1, \vec{t}_2)) = \frac{\vec{t}_1 \cdot \vec{t}_2}{|\vec{t}_1| \cdot |\vec{t}_2|} \quad (2.11)$$

The evaluation was performed on the Reuters-Corpus to show that clustering with background knowledge incorporated outperformed clustering without background knowledge.

Martin [45] also developed a method for transforming the noun-related portions of WordNet into a lexical ontology to enhance knowledge representation. Scott and Matwin [62] proposed a new representation of text based on WordNet hypernyms. These WordNet-based techniques have shown their advantages of improving the traditional VSM-based representation to some degree but suffered from the relatively limited information coverage by Wordnet.

## 2.5. DBpedia-based Approaches

### 2.5.1. Introduction to DBpedia

DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web [3].DBpedia allows you to ask sophisticated queries against datasets derived from Wikipedia and to link other datasets on the Web to Wikipedia data [3].The DBpedia project focuses on the task of converting Wikipedia content into structured knowledge, such that Semantic Web techniques can be employed against it: asking sophisticated queries against Wikipedia, linking it to other datasets on the Web, or creating new applications or mashups [3]. The information extraction and publication process of DBPedia is illustrated in Figure 2.5.

There have been a significant amount of work and applications of utilizing DBpedia to explore the knowledge from Wikipedia such as the tool named “Faceted Wikipedia Search” and the tool named “RelFinder”[41].

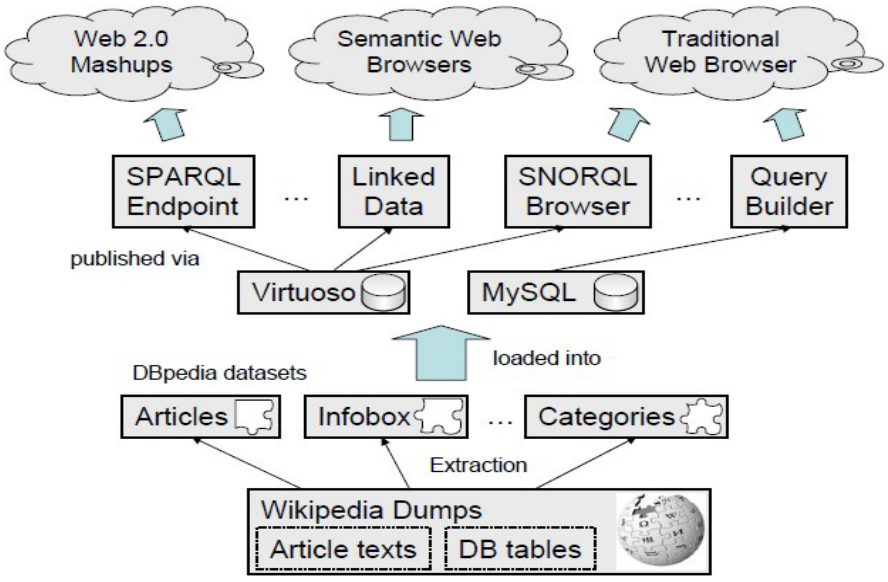


Figure 2.5. Overview of the DBpedia Components [Auer et al, 2007]

### 2.5.2. Faceted Wikipedia Search

Faceted Wikipedia Search is an alternative search interface for Wikipedia, which facilitates infobox data in order to enable users to ask complex questions against Wikipedia knowledge [25]. It was designed in concern of the incapability of searching infobox data using Wikipedia's search engine. The motivation was the infobox data in each Wikipedia article provides the most relevant facts and is well structured in the form of attribute-value pairs that are easy to be accessed for general information retrieval needs. Faceted Wikipedia Search allows users to ask complex questions, like "Which rivers flow into the Rhine and are longer than 50kilometers?" or "Which skyscrapers in China have more than 50 floors and were constructed before the year 2000?"against Wikipedia knowledge [25]. The ability of answering such questions requires structured information to be extracted from Wikipedia. Figure 2.6 gives the user interface of the Faceted Wikipedia Search. The core of this tool is the faceted search paradigm. Basically, each entity is divided into a number of subsets where each subset is defined

by an additional restriction on a property [25] referred to as a facet. Thus each facet is actually depicting an aspect of the given entity. When customizing the search for a specific information need, users can declare multiple facets needed regarding an entity to construct a complex query. As shown in Figure 2.6, Area 1 enables users to perform free text search, while customized search is provided in Area 2 and 3 which allow users to define the facets and filters respectively. Formally, the facet information is structured in the form of a set of tuples $< f, d, v >$ where each tuple has the following meaning: a document *d* has a value *v* in the facet *f*. Given a query *q* with each facet and value specified, the search algorithm returns a set of documents $D_q$ that meet the input facets and values.

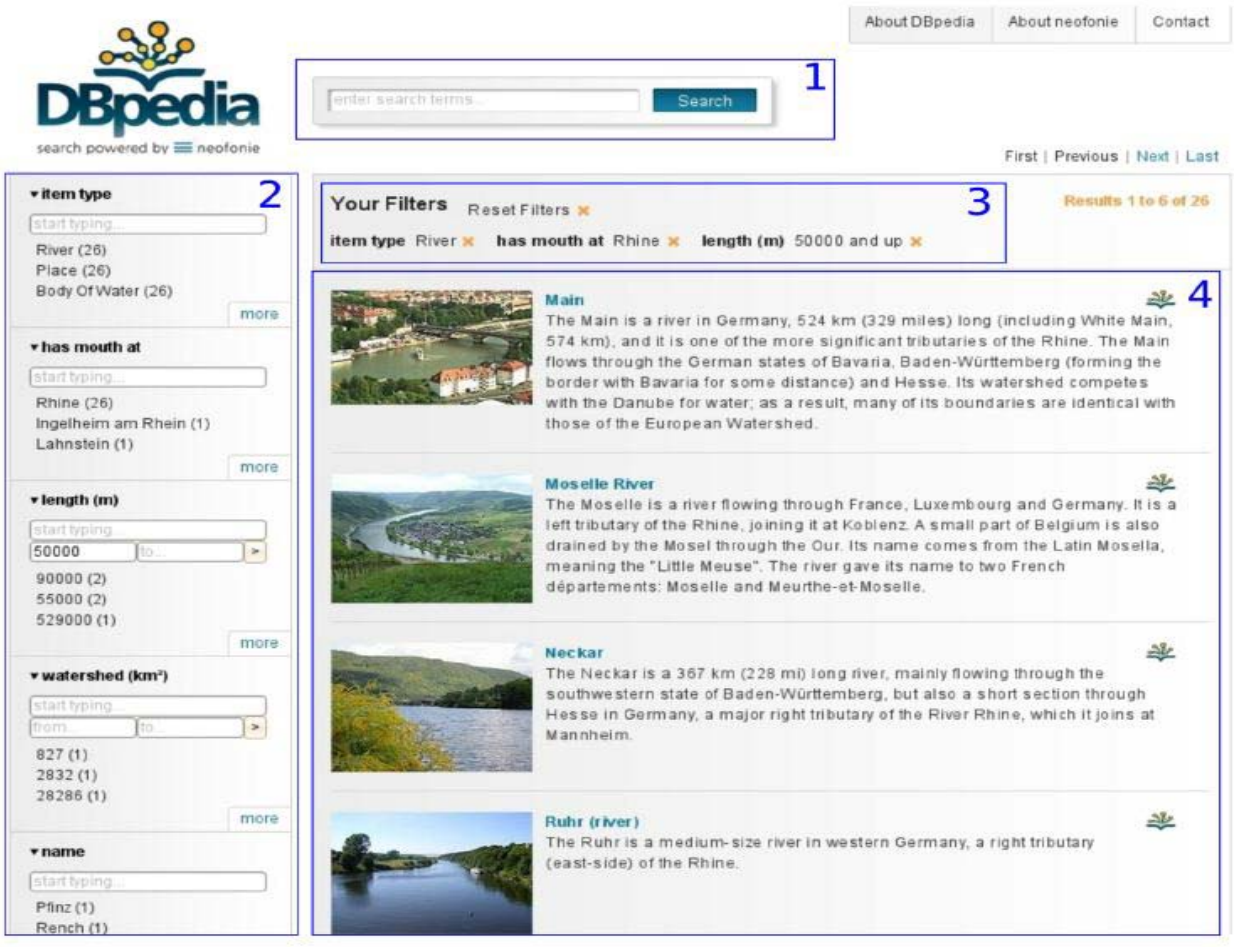


Figure 2.6. Screen Shot of the Faceted Wikipedia Search User Interface. [Hahnet al, 2010]

The major limitation of this tool is that it only takes advantage of infobox data from Wikipedia. Other rich relations such as the categorical information, the article content, the anchor texts are not considered at all. On the contrary, our models in this dissertation incorporate various information resources from Wikipedia and reasonably combine to form a comprehensive knowledge discovery framework.

### 2.5.3. RelFinder: DBpedia Relationship Finder

RelFinder is a tool for exploring connections between objects in a Semantic Web knowledge base [41]. The goal of the DBpedia Relationship Finder is to provide a user interface to explore the huge DBpedia data set [3] by providing a means to find connections between different objects [41]. It presents an approach that extracts a graph covering relationships between two objects of interest [25] and emphasizes the human aspect of relationship discovery by offering sophisticated interaction support [27]. Given two objects like two person names, RelFinder aims at finding the ontological connections between them and display the uncovered connections in a user-friendly way by adopting many existing algorithms in the background. The core two algorithms of RelFinder are illustrated in Figure 2.7 and Figure 2.8. Figure 2.7 gives the algorithm of decomposing the DBpedia infobox graph. Basically, the algorithm in Figure 2.7: 1) takes as input a RDF table containing a set of triples, 2) goes through each object in the RDF table, 3) uses a breadth first strategy to find all objects connected to the current object, and 4) puts the current object and all its connected objects into one component. Figure 2.8 shows the algorithm of finding the relationships between two objects. It starts with two given objects, computes their minimum and maximum distances, and performs SQL queries to obtain connections between them that satisfy the user specified requirements.

**Algorithm 1**: RDF Graph Decomposition.

**Input**: an RDF statements table (a set of triples)
**Output**: objects separated in components stored in a component table

```
1  create necessary database tables;
2  filter triples in the statements table and copy them in a table T;
3  initialise an empty queue Q;
4  clusterId = 0;
5  while T is not empty do
6      pick first object O from T and add it at the end of Q;
7      write O to component table;
8      while Q is not empty do
9          find all objects obj, which are object or subject of a triple in T, which contains O
           as subject or object;
10         forall O' ∈ obj do
11             if O' ∉ Q then
12                 add O' at the end of Q;
13                 add O' to component table;
14             delete triples in T containing O';
15         set O to first object in Q;
16     increment clusterId;
```

Figure 2.7. RDF Graph Decomposition [Lehmann et al, 2007]

**Algorithm 2**: Workings of the DBpedia Relationship Finder.

**Input**: first object $O_1$, second object $O_2$, maximum distance $d_{max}$, maximum number of results $n$, ignore list of objects and predicates

```
1  if query has been saved then
2      load result from cache;
3  else
4      if O1 and O2 are in the same component then
5          compute minimum distance min and maximum distance max according to
           components table;
6          compute preview connection and display it;
7          set d = min;
8          set m = 0;
9          while d < d_max and m < n do
10             formulate SQL query for obtaining at most (n − m) connections between O1
               and O2 of length d without objects and properties in the ignore list:
11             if connections exist then
12                 display connections;
13                 m = m + number of found connections;
14             increment d;
15             if d − d_max then
                   Output: no connections within the specified maximum distance exist
16     else
           Output: no connection exists, objects in different components
```

Figure 2.8. Workings of the DBpedia Relationship Finder [Lehmann et al, 2007]

However, in comparison to the graph-based approach proposed in this dissertation for concept relationship discovery, RelFinder has the following major limitations:

- It does not provide any solution for measuring the semantic relatedness between the discovered objects.
- There is no measurement of the goodness of the generated concept chains.
- It only focuses on ontological information discovery.
- It has the restricted input format and limited discovery results as demonstrated by performing queries using its provided representative query pairs. For example, if the user wants to find the potential relationships between concepts A and B, and there is no exact matching objects in the RDF database, the user needs to choose the corresponding objects for A and B on their own. In case the user does not have good knowledge of A and B, it would be difficult to choose the correct matching objects from the RDF database, while our proposed graph-based approach is able to automatically discover the relevant concepts to A and B from the space of Wikipedia.
- RelFinder is more like a finder instead of a miner, because it is actually performing SQL queries against the RDF database.

## 2.6. Wikipedia-based Approaches

### 2.6.1. Introduction to Wikipedia

Wikipedia is the largest human built encyclopedia in the world. It has over 5,000,000 articles by April 05, 2011, and is maintained by over 100,000 contributors from all over the world. As of February 2013, there are editions of Wikipedia in 285 languages. Knowledge in Wikipedia ranges from psychology, math, physics to social science and humanities.

As an open source project, the entire content of Wikipedia is easily obtainable. All the information from Wikipedia is available in the form of database dumps that are released periodically, from several days to several weeks apart. In this dissertation, we employ only part of its whole information for our data mining tasks. The main content exploited in this work is illustrated in Figure 2.9.

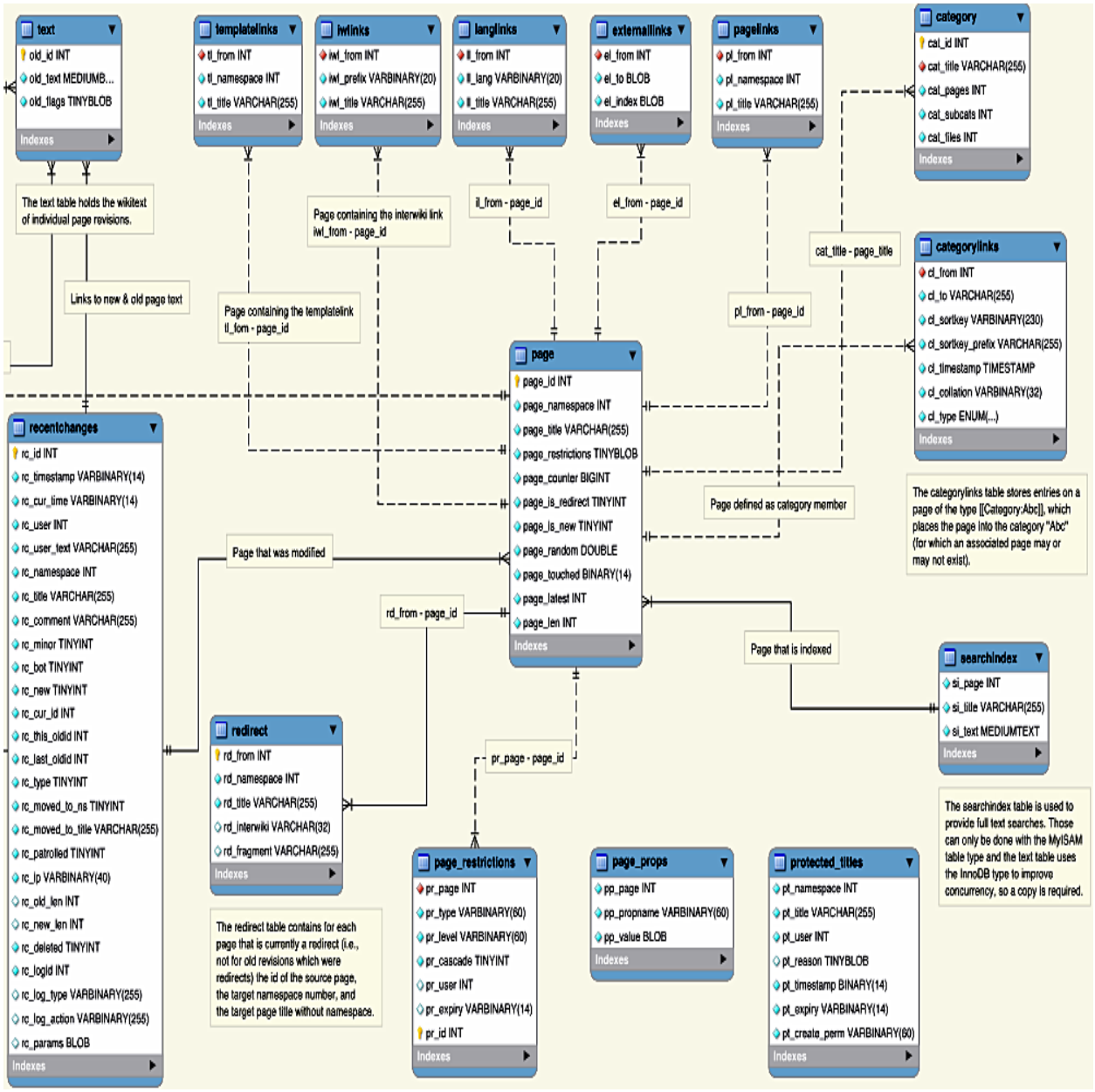


Figure 2.9. Wikipedia Database Schema

### 2.6.2. Wikipedia Article Content-based Approaches

As the world's largest knowledge base to date, Wikipedia has gained increasing popularity in serving various data mining and information retrieval tasks. Gurevych et al used Wikipedia to integrate semantic relatedness into the information retrieval process [24], and Müller et al [49] used Wikipedia in domain-specific information retrieval. Gabrilovich et al [18] applied machine learning techniques to Wikipedia and proposed a new method to enrich document representation from this huge knowledge repository. Specifically, they built a feature generator to identify most relevant Wikipedia articles for each document, and then used concepts corresponding to these articles to create new features. The feature generator acts similar to a text classifier: it receives a text fragment, and maps it to the most relevant Wikipedia articles [18]. For example, feeding "*Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge*" as input to the feature generator yields the following relevant concepts: ENCYCLOPEDIA, WIKIPEDIA, ENTERPRISE CONTENT MANAGEMENT, BOTTLENECK, PERFORMANCE PROBLEM, and HERMENEUTICS. These generated Wikipedia concepts are then candidate concepts for feature enrichment.

As claimed in [18], one of the advantages using Wikipedia over Open Directory Project (ODP) is the articles in Wikipedia are much cleaner than typical Web pages, and mostly qualify as standard written English. However, without proper feature selection strategies employed, there will still be a large amount of noise concepts introduced by the feature generator. To address this issue, [18] developed a set of simple heuristics for removing possible noise concepts through discarding articles that have fewer than 100 non-stop words or fewer than 5 incoming and outgoing links. Furthermore, disambiguation pages and those related to chronology are also

discarded. The experimental evaluation showed great improvements across a diverse collection of datasets.

The graph structure of various information resources has been used for semantic relatedness estimation such as [6, 23, 42]. Compared with Latent Semantic Analysis (LSA) [15, 38], Explicit Semantic Analysis (ESA), introduced by Gabrilovich et al. [19], is a method to represent the meaning of nature language texts using Wikipedia. Compared with [39], ESA deals with documents that are aligned with encyclopedia articles [19]. Basically, it maps a given text or a term to a conceptual vector space which is spanned by all Wikipedia articles. The conceptual vector built using ESA consists of real values indicating the Wikipedia-derived articles' association strengths to the given text or term.

Figure 2.10 shows an example of building the vector for a given text document using ESA. The effectiveness of ESA was demonstrated by comparing with two competitive baseline algorithms [40]. However, the original ESA method is subject to the noise concepts introduced, especially when dealing with multi-word phrases. As shown in Figure 2.10, the Wikipedia article "*Alyssa_Ashcroft*" is actually a survival horror video game and does nothing with the given counterterrorism document. But it was finally built into the conceptual vector and ranked in a high position according to the similarity computing model employed in ESA.

There has been interesting work utilizing ESA in a cross-lingual information retrieval setting [55, 64] to allow retrieval across languages. In that effort the authors performed article selection to filter out those irrelevant Wikipedia articles (concepts). However, we observe the selection process resulted in the loss of many dimensions in the following mapping process, whereas in our proposed approach, the process of article selection is postponed until two

semantic profiles have been merged so that the semantic loss could be possibly reduced to the minimum.

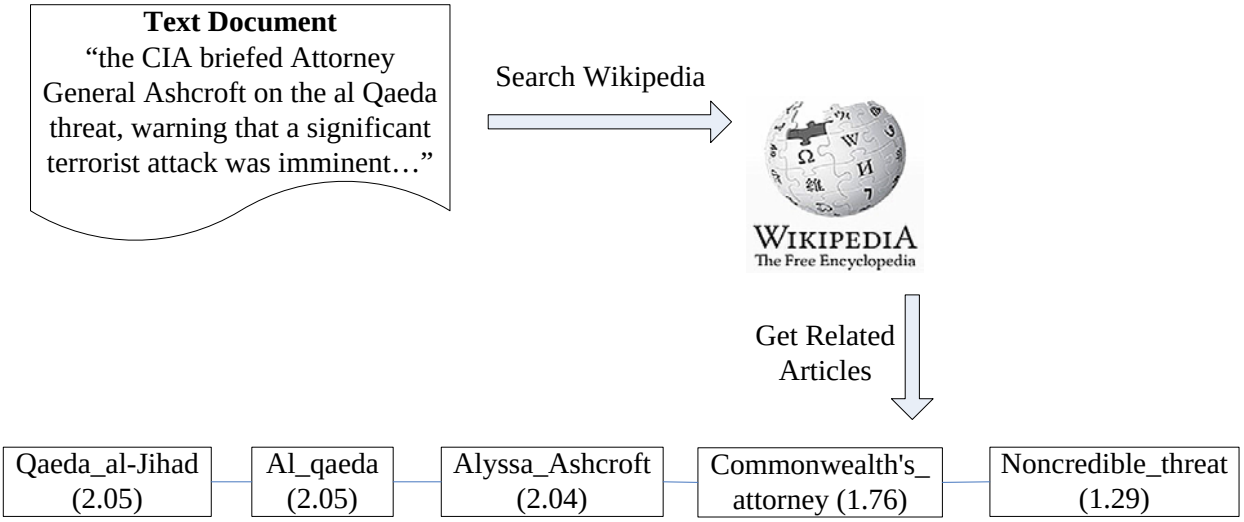


Figure 2.10. Building ESA-based Vector for a Text

### 2.6.3. Wikipedia Link Structure-based Approach

To improve semantic relatedness using Wikipedia, Milne [48] proposed a Wikipedia Link Vector Model (WLVM) which takes advantage of the link structure and titles of Wikipedia articles while ignoring their textual content. To identify the articles that might discuss the terms of interest, they did a simple string comparison between each article title and the term, and select those articles whose titles match the term. Then similar to the VSM, they used link counts weighted by the probability of each occurring link to construct vectors containing relevant Wikipedia concepts, and then computed the cosine value between two vectors to represent the semantic relatedness between two terms of interest. Suppose $t$ is the total number of articles within Wikipedia, then the weighted value $w$ for the link $a \rightarrow b$ is:

$$w(a \rightarrow b) = |a \rightarrow b| \times \log\left(\sum_{x=1}^{t} \frac{t}{|x \rightarrow b|}\right) \quad (2.12)$$

In other words, the weight of a link within a source document is its number of occurrences in the source document, multiplied by the inverse probability of any link to the target document [48]. Therefore, a link $a \rightarrow b$ is considered less significant for judging the semantic relatedness between articles if many other articles also link to the same target *b*.

Suppose there are *n* links $\{l_i \mid i = 1,...,n\}$ within article *x* and *y*, then the vector built for article *x* and *y* are as below:

$$x = (w(x \rightarrow l_1), w(x \rightarrow l_2), ..., w(x \rightarrow l_n)) \quad (2.13)$$

$$y = (w(y \rightarrow l_1), w(y \rightarrow l_2), ..., w(y \rightarrow l_n)) \quad (2.14)$$

The semantic relatedness between articles *x* and *y* is given by the angle between their corresponding vectors. Then the actual relatedness between two given terms of interest is the lowest angle found between any pair of relevant articles.

However, only the hyperlink structure of Wikipedia and article titles were extracted to compute the semantic relatedness between query terms, without any analysis of the textual contents of Wikipedia articles. And experiments have shown that solely relying on the hyperlink structure of Wikipedia and article titles makes this approach fall well behind Explicit Semantic Analysis (ESA) [19] and only outperform some of the measures provided by [68].

### 2.6.4. Wikipedia as a Thesaurus

Milne et al built a domain-specific thesaurus of agriculture using Wikipedia, and showed the thesauri derived using their techniques capitalized on existing public efforts and tended to reflect contemporary language usage better than their costly, painstakingly constructed manual counterparts [47]. Specifically, their thesaurus was built by identifying synonymy and polysemy as well as hierarchical relations as shown in Figure 2.11 from Wikipedia. However, the thesaurus

constructed in [47] only utilized a limited part of the information of Wikipedia and was specially designed for agriculture.

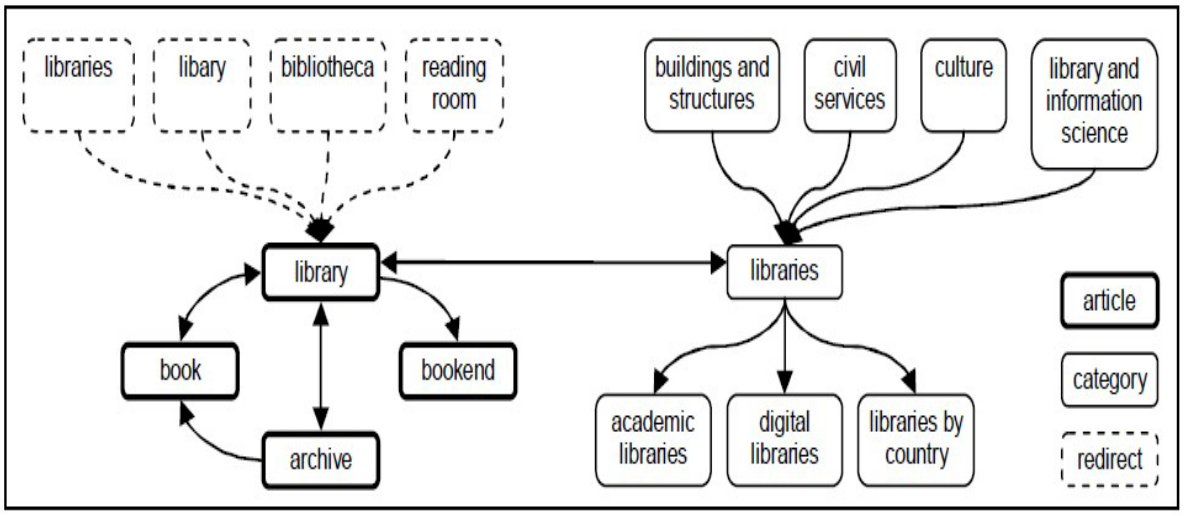


Figure 2.11. Example Structures from Wikipedia [Milne, 2006]

To overcome the limitations existing in [47] and introduce background knowledge for text classification, Wang et al [73] proposed a method to build a general thesaurus from Wikipedia. They viewed each topic of a Wikipedia article as a concept and made use of various relations provided by Wikipedia such as synonymy, polysemy, hyponymy to help measure the semantic relatedness between concepts. Their semantic relatedness estimation model took advantage of two resources provided by Wikipedia: articles and categories. The article based measure was building two concept vectors using the VSM and then calculating the cosine similarity using the two vectors. With the categorical information derived from Wikipedia, they came up with another two measures: the out-linked category-based measure and the distance-based measure. Out-linked categories of an article are the categories that out-linked articles of the original article belong to [73]. For each Wikipedia article, they built a category vector

composed of all out-linked categories for it. Supposed $\vec{c}_1$ and $\vec{c}_2$ are the two category vectors for article $a_1$ and $a_2$, the similarity between $a_1$ and $a_2$ is then defined as follows:

$$S_{olc} = \frac{\vec{c}_1 \cdot \vec{c}_2}{\vec{c}_1 \times \vec{c}_2} \tag{2.15}$$

With this weighting scheme, two articles sharing more out-linked categories in Wikipedia will have higher similarity. In addition to the out-linked categories, they also defined a distance-based measure using the hierarchical categorization structure obtained from Wikipedia. Suppose article $a_1$ belongs to category $c_1$ and article $a_2$ belongs to category $c_2$, they measured the similarity between $a_1$ and $a_2$ by calculating the shortest path of $c_1$ and $c_2$ on the categorization graph:

$$. Dis_{category}(a_1, a_2) = \frac{length(c_1, c_2)}{D} \tag{2.16}$$

Where $length(c_1, c_2)$ is the number of nodes along the shortest path between $c_1$ and $c_2$, and $D$ is the maximum depth of the categorization graph. The final weighting scheme for computing the semantic relatedness between two concepts is linearly combining the three semantic relatedness measures, i.e. the article-based measure and the two category-based measures. Based on [73], Wang et al. [72] embedded background knowledge derived from Wikipedia into a semantic kernel to enrich document representation for text classification. It is able to capture the semantic closeness of synonyms, and perform word sense disambiguation for polysemous terms. However, their method is based on a thesaurus built from Wikipedia and constructing the thesaurus requires a considerable amount of effort, and too much human intervention was involved in their process. Our *SPC* model proposed in this dissertation also adopts the similar strategy by linearly combining different similarities. The major problem of the linear combination of different

similarities obtained from different Wiki resources is that it involves too much human intervention in determining the best parameter setting. That is part of the motivations that we propose to use the proximity matrix to overcome this limitation using the kernel methods.

## 2.7. Summary

The main drawback of the VSM-based approaches is that the semantic relationships between topics cannot be found if those interlinking concepts do not co-occur in the text literally. Even for those which do co-occur with topics in texts, their semantic relatedness to topics cannot be captured and measured in a semantic way since only statistical information collected from the text corpus is taken into consideration. Such limitations have motivated an increasing number of works proposed to incorporate background knowledge as an effective aid to complement the existing knowledge contained in the text corpus. The background knowledge can be obtained in different ways, e.g. from the Web or human built knowledge bases such as WordNet, DBpedia and Wikipedia. The performance of the Web oriented approaches highly relies on the generated outputs from search engines and has not reached the satisfying level, such as the incapability of performing effective cross-document knowledge discovery. The WordNet-based techniques have shown their advantages of improving the traditional VSM-based document representation to some degree but suffered from the relatively limited information coverage and the painful maintenance. DBpedia-based approaches can overcome such limitations in comparison to WordNet since DBpedia automatically evolves as Wikipedia changes. But in terms of data quality and formalization, WordNet outperforms DBpedia. Wikipedia has gained more and more popularity in the research field of data mining. But most work of utilizing Wikipedia has been

focusing on classification and clustering. We may be among the first to propose to explore the usage of Wikipedia for cross-document knowledge discovery in this dissertation.

# CHAPTER 3. SEMANTIC PATH CHAINING

Various models for connecting topics in documents have been defined over the past years such as [9, 51, 76]. However, most focus on special problems, e.g. [6, 16, 20, 29] for community detection. This chapter presents a comprehensive text mining model for mining semantic associations between concepts across multiple text units through incorporating the extensive knowledge derived from Wikipedia. Our algorithm is motivated by Concept Chain Queries (CCQ) [30, 33] which is a VSM-based mining model. A concept chain query involving concept A and concept B intends to find the best path linking concept A to concept B. The paths found stand for potential conceptual connections between them. Figure 3.1 shows an example of the query pair "*Nashiri :: Nairobi attack*". Since "*Nashiri*" co-occurs with "*Jihad Mohammad Ali al Makki*" in the same sentence in Document 1, and "*Nairobi attack*" co-occurs with "*Jihad Mohammad Ali al Makki*" in the same sentence in Document 2, "*Nashiri*" and "*Nairobi attack*" can be linked through the concept "*Jihad Mohammad Ali al Makki*". However, CCQ is built under the assumption of the VSM-based representation without background knowledge introduced, and thus demonstrating the inherent limitations. For example, the detected links are limited to the associations occurring in the document collection where the query is performed; the semantic relatedness computing method is mainly based on statistical information collected from the corpus and no background knowledge has been taken into account.

To alleviate all such limitations, we propose Semantic Path Chaining (*SPC*), a new model for uncovering semantic paths between concepts with a focus on taking background knowledge into consideration. The approach proposed here is based on the method proposed by Srinivasan's closed text mining algorithm [67] in the biomedical domain, but we extend it to handle a more complicated query scenario where multiple-stage semantic paths are desired and also attempt to

incorporate Wikipedia knowledge to enrich document representation. Motivated by the Explicit Semantic Analysis (ESA) technique introduced by Gabrilovich et al. [19], which is able to use the space of Wikipedia articles to measure the semantic relatedness between fragments of natural language text, we develop a hybrid approach and weighting scheme that combines the advantages of ESA and content-based statistical analysis. Another distinct difference from the original ESA method is that Gabrilovich et al. only focused on document-level textual analysis through mapping a given text fragment or term to a conceptual vector space spanned by all Wikipedia articles, whereas here we extend this technique by considering other valuable evidences from Wikipedia such as categories associated with each Wiki concept to further improve the semantic relatedness estimation between concepts.

**Document 1:**
***Nashiri*** and *his cousin*, *Jihad Mohammad*, returned to Afghanistan, probably in 1997, Nashiri again encountered Bin Ladin, still recruiting for "the coming battle with the United States." Nashiri joined al Qaeda and later was recognized as the chief of al Qaeda operations in and around the Arabian Peninsula.
**Document 2:**
In late 1998, al Qaeda decided mounting an attack against a U.S. vessel and *Jihad Mohammad*, also known as *Azzam*, was a suicide bomber for the ***Nairobi attack***.

Figure 3.1. A Concept Chain Example for the Query "*Nashiri :: Nairobi attack*"

### 3.1. Semantic Paths Discovery from Documents

*SPC* is attempting to mine semantic paths between two concepts (e.g., two person names) across documents incorporating Wikipedia knowledge. We propose to use the features extracted from text corpus, as well as the relationships discovered from Wikipedia to construct semantic paths which stand for potential conceptual connections between them. Given a query involving topics A and B, we try to find (i) if there is a direct connection (association) between them (e.g.

co-occurrence in sentences), or (ii) if they can be connected by several intermediate concepts (paths) in the documents, or (iii) if they can be connected by any concepts not appearing in the documents at all. In the second case, the answer may not be contained in any particular individual document, but may be the result of relating the content of a small set of documents. And in the third case, an outside knowledge base is required for detecting those topic related concepts from outside documents. Note that all the connecting concepts detected between A and B must be semantically related to them, i.e. the semantic relatedness estimation must address the omission of the semantic content of concepts.

### 3.1.1. Ontology Mapping and Semantic Profile Representation

To detect semantic relationships between topics of interest, we first represent each topic as a semantic profile which is essentially a set of highly related concepts to the given topic in the corpus. To further differentiate between the concepts, semantic type (ontological information) is employed in profile generation. Table 3.1 illustrates part of semantic type - concept mappings.

Table 3.1. Semantic Type - Concept Mapping

| **Semantic Type** | **Instances** |
|---|---|
| Human Action | attack, killing, covert action, international terrorism |
| Leader | Vice president, chief, governor |
| Country | Iraq, Afghanistan, Pakistan, Kuwait |
| Diplomatic Building | consulate, pentagon, UAE Embassy |
| Government | Bush administration, white house, national security council |
| Person | Deputy national security adviser, chairman, executive director |

Thus each profile is defined as a vector composed of a number of semantic types.

$$profile(T_i) = \{ST_1, ST_2, ..., ST_n\} \quad (3.1)$$

Where $ST_i$ ($i=\{1,2,\ldots,n\}$) represents a semantic type to which the concepts appearing in the topic-related text snippets belong. We used a sentence as window size to measure the relevance of appearing concepts to the topic term. Under this representation each semantic type is again referred to as an additional level of vector composed of a number of terms that belong to this semantic type.

$$ST_i = \{w_{i,1}m_1, w_{i,2}m_2, ..., w_{i,n}m_n\} \quad (3.2)$$

Where $m_j$ represents a concept belonging to semantic type $ST_i$, and $w_{i,j}$ represents its weight under the context of $ST_i$ and sentence level closeness. When generating the profile we replace each semantic type in equation 3.1 with equation 3.2. In equation 3.2, to compute the weight of each concept, we employ a variation of the TFIDF weighting scheme and then normalize the weights:

$$w_{i,j} = s_{i,j} / highest(s_{i,l}) \quad (3.3)$$

Where $l = 1, 2, ..., r$ and there are totally $r$ concepts for $ST_i$, $s_{i,j} = df_{i,j} * Log(N / df_j)$, where $N$ is the number of sentences in the collection, $df_j$ is the number of sentences concept $m_j$ occurs, and $df_{i,j}$ is the number of sentences in which topic $T$ and concept $m_j$ co-occur and $m_j$ belongs to semantic type $ST_i$. By using the above three formulae we can build the corresponding profile representing any given topic.

To summarize, the procedure of building semantic profiles for a given topic *T* of interest is composed of the following four steps:

1) Concept Extraction: extract all potential concepts from the document collection which co-occur with the topic *T* in the sentence level.
2) Semantic Type Employment: each concept will be associated with and grouped under one or more semantic types (e.g., Human Action, Country, Person) which it belongs to.
3) Weight Calculation: for each concept, a variation of the TFIDF scheme is used to calculate its weight
4) Weight Normalization: within each semantic type, the concept weights are further normalized by the highest concept weight observed for the semantic type, and then ranked according to the normalized weights.

### 3.1.2. Chaining Semantic Paths

In this step, we search potential conceptual connections in different levels, and use them to construct semantic paths linking two given topics (concepts). It is inspired by the closed text mining algorithm [67], which in turn is based on the discovery framework established by Swanson et al. [71] in the biomedical domain. Suppose A and C are two given topics of interest, the algorithm of generating semantic paths connecting A to C from the text corpus is composed of the following sequential steps:

1) Conduct independent searches for A and C. Build the A and C profiles. Call these profiles AP and CP respectively.
2) Compute a B profile (BP) composed of terms in common between AP and CP. The weight of a concept in BP is the sum of its weights in AP and CP. This is the first level of intermediate potential concepts.

3) Expand the concept chain using the created BP profile together with the topics to build additional levels of intermediate concept lists which (i) connect the topics to each concept in BP profile in the sentence level within each semantic type, and (ii) normalize and rank them.

### 3.2. Semantic Relatedness Measurement with Wikipedia Articles

To utilize Wikipedia knowledge to complement the existing information in the document collection, we adapt the Explicit Semantic Analysis (ESA) technique proposed by Gabrilovich et al. [19] as our underlying content-based measure for analyzing Wikipedia articles relevant to the given topics of interest.

#### 3.2.1. Document Representation with ESA

Under the ESA method, each article in Wikipedia is treated as a concept, and each document is represented by an interpretation vector containing related Wikipedia concepts (articles) to the document.

$$\phi(d) =< as(d, a_1), ..., as(d, a_n) > \tag{3.4}$$

Where $as(d, a_i)$ represents the association strength between document $d$ and Wikipedia article $a_i$. Suppose $d$ is spanned by all words appearing in it, i.e. $d =< w_1, w_2, ..., w_j >$, the association strength $as(d, a_i)$ is computed as follows:

$$as(d, a_i) = \sum_{w_j \in d} tf_d(w_j) tf \cdot idf_{a_i}(w_j) \tag{3.5}$$

Where $tf_d(w_j)$ is the frequency of word $w_j$ in document $d$, and $tf \bullet idf_{a_i}(w_j)$ is the $tf \bullet idf$ value of word $w_j$ in Wikipedia article $a_i$. As a result, the vector for a document is represented by a list of real values indicating the association strength of a given document with respect to Wikipedia articles. By using efficient indexing strategies such as single-pass in memory indexing, the computational cost of building these vectors for a given term (or text fragments containing multiple terms) can be reduced to within 200-300 ms.

### 3.2.2. Noise Cleaning with Heuristics

As discussed above, the original ESA method [19] is subject to the noise concepts introduced, especially when dealing with multi-word phrases. For example, when the input is *Angelina Jolie*, the generated interpretation vector will contain a fair amount of noise concepts such as *Eudocia Angelina*, who was the queen consort of Stephen II Nemanjić of Serbia from 1196 to 1198. This Wikipedia concept (article) is selected and ranked high in the interpretation vector because the term *Angelina* occurs many times in the article "*Eudocia Angelina*", but obviously this article is irrelevant to the given topic *Angelina Jolie*. In order to make the interpretation vector more precise and relevant to the topic, a sequence of heuristics is devised to clean the vector as shown in Figure 3.2.

More specifically, a modified Levenshtein Distance algorithm is devised to measure the relevance of the given topic to each Wikipedia concept generated in the interpretation vector with a single word as a unit for allowable edit operations, which allows the adapted algorithm to be used to compute the similarity between any two text snippets. If the topic contains only one word, then the number of its occurrences in the corresponding Wikipedia article is used for judgement. If it occurs more than three times, this article is viewed as relevant to the given topic

and kept in the interpretation vector. If the topic contains multiple words, we view each word as a character and employ our adapted version of the Levenshtein distance algorithm to evaluate the relevance of the topic to the article text. If their Levenshtein distance is under the defined threshold, the article is viewed as relevant. Otherwise, it will be removed from the interpretation vector.

Input: a topic T of interest
an interpretation vector V representing the topic T

Output: a cleaned Wikipedia-based concept vector V' representing the topic T

1. If T is a single word topic, then count the number of occurrences of T in the article texts represented by each concept $v_i$ in V, respectively. If T occurs more than 3 times, then keep $v_i$ in V, otherwise, remove $v_i$ from V.

2. If T is a multi-word topic, then the adapted Levenshtein distance algorithm applies to measure the relevance of each Wikipedia concept (article) $v_i$ in V to topic T.

2.1. If NumOfWords(T) $\leq$ 2, then extract all text snippets $TS_j$ within the window size NumOfWords(T)+1 from the article text of $v_i$. If there exists a j such that LevenshteinDistance(T, $TS_j$) <= 1, then keep $v_i$ in V, otherwise, remove $v_i$ from V.

2.2. If NumOfWords(T) > 2, then extract all text snippets $TS_j$ within the window size NumOfWords(T)+2 from the article text of $v_i$. If there exists j such that LevenshteinDistance(T, $TS_j$) $\leq$ 2, then keep $v_i$ in V, otherwise, remove $v_i$ from V.

Figure 3.2. Interpretation Vector Cleaning Procedure

Table 3.2 illustrates the effect of removing irrelevant concepts from the interpretation vector.

Table 3.2. Interpretation Vector Cleaning Results

| **Query** | Machine Learning | |
|---|---|---|
| **Top 10 Wikipedia Concepts** | | |
| **#** | **Before Cleaning** | **After Cleaning** |
| **1** | Machine Learning | Machine Learning |
| **2** | Anti-Hebbian Learning | Machine Learning Algorithm |
| **3** | Project-based Learning | Statistical Learning |
| **4** | Student-centered Learning | Learning Algorithms |
| **5** | Machine Learning Algorithm | Learning to Learn |
| **6** | Post-Turing Machine | Cumulative Learning |
| **7** | Breton-Pretot Machine | Transductive Learning |
| **8** | Z MACHINE | State Machine |
| **9** | Machine EP | Learning Theory (disambiguation) |
| **10** | Turing-Post Machine | Reinforcement Learning |

### 3.2.3. Computing Semantic Relatedness

After the cleaning step, we are able to use the resulting interpretation vectors for computing similarities between any two concepts. In our context of mining associations between two topics, say A and C, we compute the Cosine similarity between the interpretation vectors of topic A and each concept $v_i$ (as shown in Figure 3.2) in the intermediate BP profile, as well as between topic C and each concept $v_i$, and take the average of two Cosine similarities as the overall similarity for each concept $v_i$ in BP profile.

Formally, given two topics of interest *A* and *B*, they can be represented by two interpretation vectors spanning related Wikipedia concepts respectively as follows:

$$V(A) = \{as(A, a_1), as(A, a_2), ..., as(A, a_n)\} \quad (3.6)$$

$$V(B) = \{as(B, a_1), as(B, a_2), ..., as(B, a_n)\} \quad (3.7)$$

Where $as(A/B, a_i)$ is the association strength between topic *A*/*B* and Wikipedia article $a_i$, and *n* is the total number of articles in Wikipedia. The semantic relatedness between topic *A* and *B* can be calculated using the angle of the two interpretation vectors.

$$sim(A, B) = \cos(\theta(V(A), V(B))) \quad (3.8)$$

We integrate the adapted Explicit Semantic Analysis method into the Concept Chain Queries by applying it in the process of computing semantic relatedness when generating semantic profiles. Suppose we are given two topics of interest: *George Bush* and *Bin Ladin*, we first extract all concepts that co-occur with both *George Bush* and *Bin Ladin* to construct an intermediate profile $profile(B) = \{t_1, t_2, ..., t_m\}$. Then, we use the adapted ESA method to compute the semantic relatedness between each concept in the intermediate profile $t_i$ and the two topics (*George Bush* and *Bin Ladin*), and take the average of the two values as the final weight for $t_i$ as below:

$$w(t_i) = \frac{sim(\text{George Bush}, t_i) + sim(\text{Bin Ladin}, t_i)}{2} \quad (3.9)$$

### 3.3. Semantic Relatedness Measurement with Wikipedia Categories

Human edited categories associated with each Wikipedia concept (article), another valuable resource provided by Wikipedia, have also been integrated to better serve the knowledge discovery task. The goal of the category system in Wikipedia is to provide

navigational links to all Wikipedia articles in a hierarchy of categories which readers can browse and quickly find sets of articles on the topics that are defined by those characteristics. Figure 3.3 illustrates part of the Wikipedia category system. Each article in Wikipedia belongs to one or more categories. We found that if two articles share more categories, they are more likely strongly related to each other.

Based on the assumption that those concepts (articles) sharing similar categories may be closer to each other in terms of semantic relatedness, a Wikipedia category interpretation vector has been built for each desired Wikipedia concept and the semantic relatedness between two concepts of interest is determined by the percentage of common categories shared by the two corresponding category interpretation vectors.

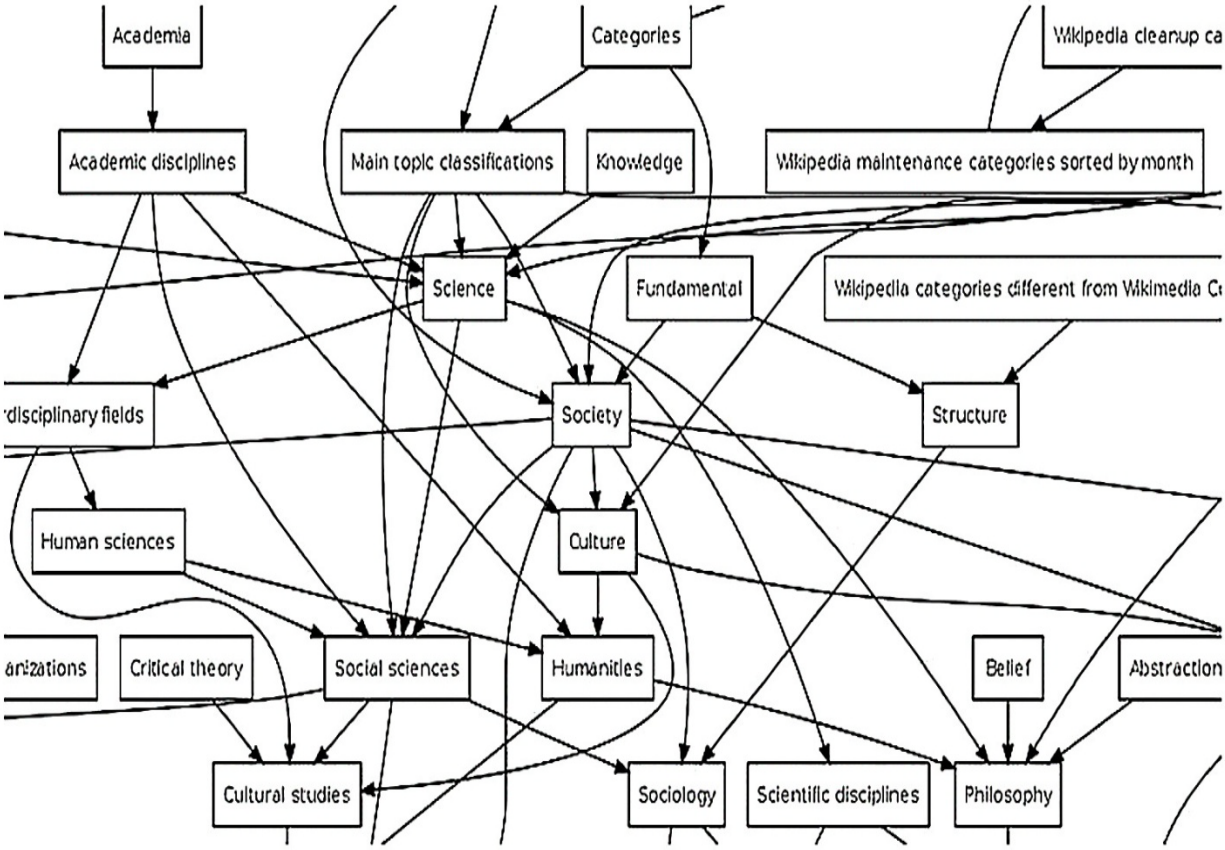


Figure 3.3. Part of Wikipedia Category System

Formally, suppose the interpretation vector for article $a_i$ is $V_i =< p_1, p_2, ..., p_m >$, where $p_i$ in $V_i$ represents a Wikipedia page (or article) that is relevant to $a_i$, then article $a_i$ can be further represented as a *Category Space Vector* (*CSV*) as follows spanning the Wikipedia category space.

$$CSV(a_i) =<< w_{i,1,1}c_{1,1}, w_{i,2,1}c_{2,1}, ... >, ..., < w_{i,1,m}c_{1,m}, w_{i,2,m}c_{2,m}, ... >> \quad (3.10)$$

Where $c_{x,y}$ represents category $c_x$ that $p_y$ in $V_i$ belongs to, and $w_{i,x,y}$ is the weight of $c_{x,y}$. To calculate $w_{i,x,y}$, we count the number of sub-vectors within $CSV(a_i)$ in which $c_{x,y}$ appears, and then normalize it:

$$w_{i,x,y} = \frac{w_{i,x,y}}{highest(w_{i,d,y})} \quad (3.11)$$

Where $d = 1, 2, ..., r$ and there are totally $r$ categories in Wikipedia. The semantic relatedness between two Wikipedia concepts (articles) can then be computed by the Cosine similarity between their corresponding *CSVs*.

Figure 3.4 shows the categories built for three concepts: "*Distributed Computing*," "*Cloud Computing*" and "*Software Engineering*." The produced semantic relatedness between "*Distributed Computing*" and "*Cloud Computing*" is 0.715, 0.094 between "*Distributed Computing*" and "*Software Engineering*", and 0.151 between "*Cloud Computing*" and "*Software Engineering*". This is consistent with our understanding that "*Distributed Computing*" and "*Cloud Computing*" are more semantically related, while both of them are less related to "*Software Engineering*".

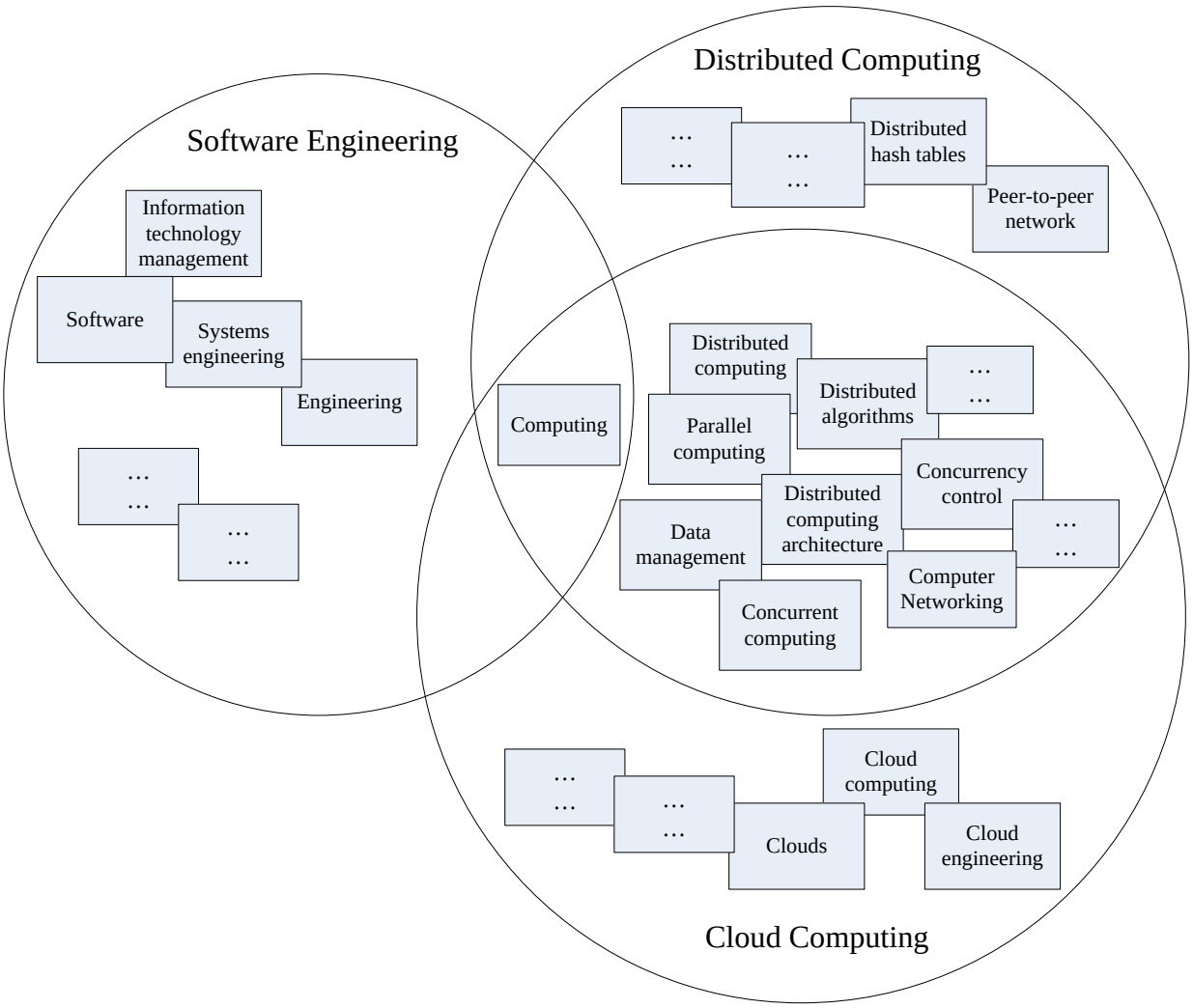


Figure 3.4. Category Overlaps of the Concepts in the Interpretation Vectors of "*Distributed Computing*", "*Cloud Computing*" and "*Software Engineering*"

### 3.4. Final Weighting Scheme

A final ranking for each concept generated in the intermediate profiles is calculated by linearly combining its TFIDF-based similarity, content-based similarity and category-based similarity together as below:

$$S_{overall} = \lambda_1 \cdot S_{TFIDF} + \lambda_2 \cdot S_{wiki-article-content} + (1-\lambda_1-\lambda_2) \cdot S_{wiki-category} \quad (3.12)$$

Where $\lambda_1$ and $\lambda_2$ are two tuning parameters that can be adjusted based on the preference on the two similarity schemes in the experiments. $S_{TFIDF}$ refers to the similarity computed using the VSM, and $S_{wiki-article-content}$ and $S_{wiki-category}$ refer to the semantic relatedness computed using the adapted ESA method and Wikipedia categories respectively.

### 3.5. The New Model of Mining Semantic Relationships

After defining the semantic relatedness measures between concepts, we are presenting now the new solution for building semantic paths between concepts. Suppose A and C are two given topics of interest, with Wikipedia knowledge incorporated in our model, we are able to leverage Wiki concepts to enrich the relationships (i.e., not limited to those occurring in the document collection literally). Thus the generated links would be an integration of relationships identified from the text corpus plus from Wikipedia knowledge. The process can be summarized as the following major steps and is further illustrated in Figure 3.5.

1) Build ESA-based interpretation vectors for A and C. Employ the cleaning procedure illustrated in Figure 3.2 to remove noise concepts in the generated interpretation vectors. The concepts that survived after cleaning are ordered according to their association strength as described in Section 3.2. , and will be serving as potentially novel connections between topics A and C.
2) Enrich the generated BP profile with newly identified Wiki concepts (represented by the corresponding Wikipedia article titles) by merging the cleaned interpretation vectors for topics A and C. The weight of each newly identified Wiki concept in BP is the sum of its association strengths in the cleaned interpretation vectors for topics A and C.

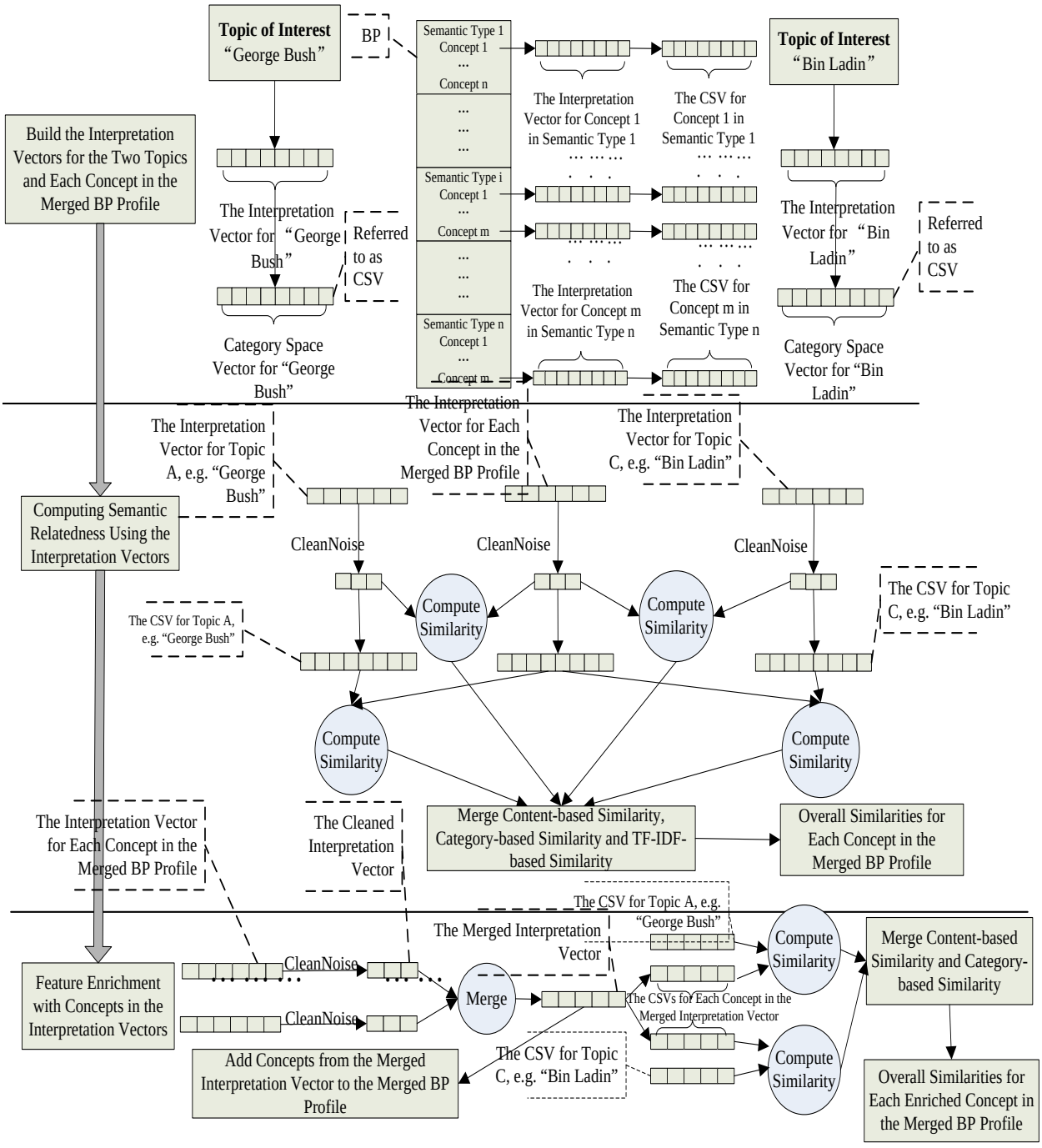


Figure 3.5. The New Model of Mining Semantic Relationships

3) Go through the same procedures as in the above two steps to enrich DP and EP profiles that contain the intermediate concepts connecting the topics to each concept in BP profile.

4) The BP profile is further enriched by considering relevant Wiki categories that the newly identified Wiki concepts (articles) belong to. The weight of each newly identified Wiki category in BP is the same as that of the corresponding Wiki concept.
5) Go through the same procedure as in Step 4 to enrich DP and EP profiles with the newly identified relevant Wiki categories.

### 3.6. Summary

Our focus in this chapter can be considered as having three dimensions: 1) we concentrate on cross-document relationship discovery where the traditional search paradigm such as search engines cannot help much; 2) we attempt to address the omission of word semantics of the Vector Space Model (VSM) by providing a better modeling of knowledge representation and semantic relatedness estimation; 3) we enrich our search space by extending to a high-dimensional space of natural concepts derived from Wikipedia that may not co-occur literally with entities of interest in the text corpus. Using the *SPC* model, we are able to represent the relationships between any two concepts by considering both the statistical information from the documents and the background knowledge from Wikipedia. Basically, the *SPC* model performs bidirectional search looking for intermediate links between two given topics of interest e.g. topics A and C. The resulting links are always obtained by considering both sides (i.e. topics A and C) so that they are highly relevant to both of the given topics. However, due to the combinatorial explosion problem of enumerating all intermediate links, we currently consider the lengths of the generated concept chains at most 4. Even though, it is still able to discover a fair amount of chains covering various scenarios, and the results have unearthed a great number of interesting relationships for each pair of topics. In case the user is more interested in finding

chains without being limited to length of 4, we propose another mining model which is graph-based and able to perform more flexible search in the following section.

# CHAPTER 4. KERNEL METHODS

This chapter presents the kernel methods for solving concept chain queries [78]. There have been a fair amount of approaches proposed to address the VSM limitations. In this work, we propose different approaches by building semantic kernels in concern of the omission of the semantic meanings of words in VSM. The basic idea of the kernel methods is to embed the data in a new suitable feature space (with more information integrated), such that solving the problem in the new space is easier (e.g. linear). To be exact, the new space here stands for the space that incorporates Wikipedia knowledge, and the kernel represents the semantic relationships between two concepts/topics uncovered in this new space.

## 4.1. Semantic Kernels for Concept Relationship Queries

Concept relationship queries against a set of documents aim at discovering the underlying semantic relationships between concepts from the given documents. Using the VSM, given a dictionary containing *N* number of concepts, a topic *T* of interest can be represented using a weighted vector as shown below:

$$\phi : T \mapsto \phi(T) = (tf(t_1, T), tf(t_2, T), ..., tf(t_N, T)) \in \mathbb{R}^N \quad (4.1)$$

Where $tf(t_i, T)$ represents the frequency of the concept $t_i$ and the topic *T* co-occurring in the same sentence. Based on this representation, we define a semantic kernel to represent the relationships between two topics $T_1$ and $T_2$ as follows:

$$k(T_1, T_2) = \phi(T_1)\phi(T_2)^T$$

$$= \begin{pmatrix} tf(t_1, T_1) & tf(t_2, T_1) & . & . & tf(t_N, T_1) \end{pmatrix} \times \begin{pmatrix} tf(t_1, T_2) \\ tf(t_2, T_2) \\ \dots \\ tf(t_N, T_2) \end{pmatrix} \quad (4.2)$$

$$= \sum_{i=1}^{N} tf(t_i, T_1) tf(t_i, T_2)$$

However, with the above representation, two semantically equivalent topics may be mapped to dissimilar feature space, if they differ a lot in their co-occurring vocabularies, since the underlying semantic meaning of concepts is neglected using the given representation. To integrate proper embedding knowledge into the topic representation, we define a kernel matrix *M* that incorporates outside knowledge by enriching the VSM document representation through $\tilde{\phi}(T) = \phi(T)M$ . Once we obtain the matrix *M*, a semantic kernel between two given topics of interest $T_1$ and $T_2$ is then defined as below:

$$\begin{aligned} k(T_1, T_2) &= \phi(T_1)MM^{\mathrm{T}}\phi(T_2)^{\mathrm{T}} \\ &= \phi(T_1)M(\phi(T_2)M)^{\mathrm{T}} \\ &= \tilde{\phi}(T_1)\tilde{\phi}(T_2)^{\mathrm{T}} \end{aligned} \quad (4.3)$$

The semantic matrix *M* can be constructed by creating a sequence of successive embeddings to add additional refinement to the semantics of the representation. One of the alternative solutions is defining *M* as below:

$$M = RP \quad (4.4)$$

Where *R* is a diagonal matrix giving the concept weightings or relevance, and *P* is a proximity matrix defining the semantic relatedness between different concepts in the document collection. Given that $\phi(T)$ is composed of a number of real values indicating the number of occurrences of each concept, *R* can be defined using the inverted document frequency to form a variation of the

TFIDF weighting scheme through multiplying $\phi(T)$ by the matrix *R*: $\phi(T)R$. The proximity matrix *P* is defined to address the semantic omission of the TFIDF weighting scheme by relating semantically related concepts together. Furthermore, the entries in *P* are constructed by semantically calculating the relatedness between different concepts rather than solely taking their number of occurrences into account:

$$P_{i,j} = \begin{cases} non-zero & if\ i \neq j,\ and\ t_i\ and\ t_j\ are\ semantically\ related \\ 1 & if\ i = j \end{cases} \tag{4.5}$$

Therefore, formally a semantic kernel between two topics of interest $T_1$ and $T_2$ is defined as below:

$$\begin{aligned} k(T_1, T_2) &= \phi(T_1)RP(RP)^{\mathrm{T}}\phi(T_2)^{\mathrm{T}} \\ &= (\phi(T_1)R)PP^T(\phi(T_2)R)^{\mathrm{T}} \\ &= \tilde{\phi}(T_1)PP^T\tilde{\phi}(T_2)^{\mathrm{T}} \\ &= \tilde{\tilde{\phi}}(T_1)(\tilde{\tilde{\phi}}(T_2))^T \end{aligned} \tag{4.6}$$

Different variations following this definition can be obtained by defining different types of proximity matrices. We will introduce in detail how various types of proximity matrices are defined for various purposes.

### 4.2. Pattern Analysis for Topic and Concept Representation

Given a dictionary *D* containing *n* number of concepts $D = (c_1, c_2, ..., c_n)$, and a topic collection *X* spanned by *r* number of topics: $X = \{t_1, t_2, ...t_r\}$ where $t_i \in X \subseteq \mathbb{R}^n$, a topic $t_i$ can be represented as a space vector spanned by *n* number of concepts from the dictionary *D*: $t_i = < sim(t_i, c_1), sim(t_i, c_2), ..., sim(t_i, c_n) >$, where $sim(t_i, c_k), k \in \{1, 2, ..., n\}$ represents the semantic relatedness between $t_i$ and $c_k$; in the similar way, a concept $c_i$ can also be represented

as a *r* dimensional space vector spanned by *r* number of topics from *X*:

$c_i =< sim(c_i, t_1), sim(c_i, t_2), ..., sim(c_i, t_r) >$, where $sim(c_i, t_k), k \in \{1, 2, ..., r\}$ represents the semantic relatedness between $c_i$ and $t_k$. The questions are: 1) if there exists a linear pattern among the *n* features for a topic $t_i$ from *X* with a label $y_i$ calculated by a linear predictive function $y_i = f(t_i) = \sum_{k=1}^{n} \alpha_k \cdot sim(t_i, c_k)$, how do we model the relationships between two topics so that the loss of the predictive function can be minimized? 2) if there exists a linear pattern among the *r* features for a concept $c_i$ from *D* with a label $y_i$ calculated by a linear predictive function $y_i = f(c_i) = \sum_{k=1}^{r} \alpha_k \cdot sim(c_i, t_k)$, how do we model the relationships between two concepts so that the loss of the predictive function can be minimized? Note that here the difference between a topic and a concept is that a topic from *X* may be represented by more than one concept from *D*.

### 4.2.1. Concept Pattern Analysis

To answer the first question, we represent *X* as the following where each row is a topic related vector $t_i =< sim(t_i, c_1), sim(t_i, c_2), ..., sim(t_i, c_n) >$:

$$X = \begin{bmatrix} sim(t_1, c_1) & sim(t_1, c_2) & . & . sim(t_1, c_n) \\ sim(t_2, c_1) & sim(t_2, c_2) & . & . sim(t_2, c_n) \\ ... & ... & ... & ... \\ sim(t_r, c_1) & sim(t_r, c_2) & . & . sim(t_r, c_n) \end{bmatrix}$$

Under the assumption of this question (i.e. there exists a linear pattern among the *n* number of concepts for a topic), the output labels *Y* for *X* can be calculated through: $Y = X\alpha$ where $Y = (y_1, y_2, ..., y_r)^T$. The training data is in the following format:

$$sim(t_i, c_1), sim(t_i, c_2), ..., sim(t_i, c_n), label_i$$

Given the above *r* training examples with labels $y_i \in Y \subseteq \mathbb{R}$, our task here is predicting the label for a new coming topic vector using the function $f(t_i) = \sum_{k=1}^{n} \alpha_k \cdot sim(t_i, c_k)$. Given the training errors are calculated through: $\xi = Y - f(X) = Y - X\alpha$, we use the standard least squares method to measure the expected loss $E(L(Y, f(X)))$ (where $L(Y, f(X)) = (Y - f(X))^2$ is the loss function) of the predicted label for the *r* training examples in *X*:

$$L(Y, f(X)) = (Y - f(X))^2 = \sum_{i=1}^{r} (y_i - f(t_i))^2 = \sum_{i=1}^{r} \xi_i^2 \tag{4.7}$$

Since we have $\xi^2 = (Y - X\alpha)^T (Y - X\alpha)$, by taking the derivative of the loss function with respect to $\alpha$ and setting it to 0, we get the following:

$$\begin{aligned} &\frac{\partial L(Y, f(X))}{\partial \alpha} = \frac{\partial ((Y - X\alpha)^T (Y - X\alpha))}{\partial \alpha} = 0 \\ &\Leftrightarrow \frac{\partial (Y^T - \alpha^T X^T)(Y - X\alpha)}{\partial \alpha} = 0 \\ &\Leftrightarrow \frac{\partial (Y^T Y - Y^T X\alpha - \alpha^T X^T Y + \alpha^T X^T X\alpha)}{\partial \alpha} = 0 \end{aligned} \tag{4.8}$$

In linear algebra, the trace of a n-by-n square matrix *X* is defined to be the sum of the diagonal elements:

$$tr(X) = x_{11} + x_{22} + ..., x_{nn} = \sum_{i=1}^{n} x_{ii} \tag{4.9}$$

Thus, the trace of a real number is just itself, we then get the following:

$$
\begin{aligned}
&\frac{\partial(Y^TY - Y^TX\alpha - \alpha^TX^TY + \alpha^TX^TX\alpha)}{\partial\alpha} \\
&= \frac{\partial(tr(Y^TY - Y^TX\alpha - \alpha^TX^TY + \alpha^TX^TX\alpha)}{\partial\alpha} = 0 \\
&\Leftrightarrow \frac{\partial(tr(\alpha\alpha^TX^TX)) - \partial(tr(\alpha Y^TX)) - \partial(tr(Y^TX\alpha))}{\partial\alpha} = 0 \\
&\Leftrightarrow X^TX\alpha + X^TX\alpha - X^TY - X^TY = 0 \\
&\Leftrightarrow X^TX\alpha = X^TY \\
&\Leftrightarrow \alpha = (X^TX)^{-1}X^TY
\end{aligned}
\tag{4.10}
$$

Therefore, $\alpha$ needs to maintain as many parameters as the dimensions of the dictionary *D* to minimize the squared loss. The time complexity of choosing the parameter set $\alpha$ is $O(n^3)$. However, it is often the case that $X^TX$ is not invertible, and problems suffering from this difficulty are known as ill-conditioned [63]. To address the problems in these situations, ridge regression is used to find the parameters that minimize the least squares of the loss as below:

$$
L(Y, f(X)) = \lambda\|\alpha\|^2 + \sum_{i=1}^{r}(y_i - f(t_i))^2 \tag{4.11}
$$

Where $\lambda$ is a parameter that controls the complexity of the model. In the same way, by taking the derivative of the loss function against $\alpha$ and setting it to 0, we get the following:

$$
\begin{aligned}
&\frac{\partial L(Y, f(X))}{\partial\alpha} = 2\lambda\alpha + 2X^TX\alpha - 2X^TY = 0 \\
&\Leftrightarrow \lambda\alpha + X^TX\alpha = X^TY \\
&\Leftrightarrow (\lambda I_n + X^TX)\alpha = X^TY
\end{aligned}
\tag{4.12}
$$

Where $I_n$ is a n-by-n identity matrix. Here the matrix $\lambda I_n + X^TX$ is always invertible if $\lambda > 0$, such that $\alpha$ can be represented as:

$$
\alpha = (\lambda I_n + X^TX)^{-1}X^TY \tag{4.13}
$$

Also, according to equation 4.12, $\alpha$ can also be represented as follows:

$$
\begin{aligned}
&\lambda\alpha + X^T X\alpha = X^T Y \\
&\Leftrightarrow \lambda\alpha = X^T (Y - X\alpha) \\
&\Leftrightarrow \alpha = \lambda^{-1} X^T (Y - X\alpha) \\
&\Leftrightarrow \alpha = X^T \beta
\end{aligned}
\tag{4.14}
$$

Where $\beta = \lambda^{-1}(Y - X\alpha)$. Next, we have the following:

$$
\begin{aligned}
&\beta = \lambda^{-1}(Y - X\alpha) \\
&\Leftrightarrow \lambda\beta = Y - X\alpha \\
&\Leftrightarrow \lambda\beta + X\alpha = Y \\
&\Leftrightarrow \lambda\beta + XX^T\beta = Y \\
&\Leftrightarrow (XX^T + \lambda I_r)\beta = Y \\
&\Leftrightarrow \beta = (XX^T + \lambda I_r)^{-1} Y \\
&\Leftrightarrow \beta = (G + \lambda I_r)^{-1} Y
\end{aligned}
\tag{4.15}
$$

Where $I_r$ is a r-by-r identity matrix and *r* is the number of the training examples, $G = XX^T$ is the Gram matrix of the rows of matrix *X*. $G = XX^T$ equals the following:

$$
G = \begin{bmatrix}
sim(t_1, c_1) & sim(t_1, c_2) & . & . & sim(t_1, c_n) \\
sim(t_2, c_1) & sim(t_2, c_2) & . & . & sim(t_2, c_n) \\
... & ... & ... & ... & \\
sim(t_r, c_1) & sim(t_r, c_2) & . & . & sim(t_r, c_n)
\end{bmatrix} \times \begin{bmatrix}
sim(t_1, c_1) & sim(t_2, c_1) & . & sim(t_r, c_1) \\
sim(t_1, c_2) & sim(t_2, c_2) & ... & sim(t_r, c_2) \\
... & ... & ... & ... \\
sim(t_1, c_n) & sim(t_2, c_n) & . & sim(t_r, c_n)
\end{bmatrix} \tag{4.16}
$$

Where $G_{ij} =< t_i, t_j >$ represents the relationships between $t_i$ and $t_j$:

$$
\begin{aligned}
&G_{ij} =< t_i, t_j > \\
&= sim(t_i, c_1) sim(t_j, c_1) + sim(t_i, c_2) sim(t_j, c_2) + \cdots + sim(t_i, c_n) sim(t_j, c_n) \\
&= \sum_{k=1}^{n} sim(t_i, c_k) sim(t_j, c_k)
\end{aligned}
\tag{4.17}
$$

As said earlier, $t_i$ and $t_j$ are two topics represented by their corresponding n-dimensional vector: $t_i =< sim(t_i, c_1), sim(t_i, c_2), \cdots, sim(t_i, c_n) >$ and $t_j =< sim(t_j, c_1), sim(t_j, c_2), \cdots, sim(t_j, c_n) >$, so we now have $< t_i, t_j >= t_i (t_j)^T$ to be the semantic kernel between $t_i$ and $t_j$. Note that the

Gram matrix and the matrix ($G + \lambda I_r$) are both r-by-r matrices, and thus it would be much more efficient to choose $\beta$ by solving a r-by-r matrix than a n-by-n matrix.

Therefore, the answer for the first question is: representing the relationship between two topics $t_i$ and $t_j$ using the dot product of $t_i =< sim(t_i, c_1), sim(t_i, c_2), \cdots, sim(t_i, c_n) >$ and $t_j =< sim(t_j, c_1), sim(t_j, c_2), \cdots, sim(t_j, c_n) >$ can minimize the loss of the predictive function.

### 4.2.2. Topic Pattern Analysis

For the second question, *X* can be represented as the following where each row is a r-dimensional space vector $c_i =< sim(c_i, t_1), sim(c_i, t_2), ..., sim(c_i, t_r) >$:

$$X = \begin{bmatrix} sim(c_1, t_1) & sim(c_1, t_2) & . & .sim(c_1, t_r) \\ sim(c_2, t_1) & sim(c_2, t_2) & ... & sim(c_n, t_r) \\ ... & ... & ... & ... \\ sim(c_n, t_1) & sim(c_n, t_2) & . & .sim(c_n, t_r) \end{bmatrix}$$

Under the assumption of this question (i.e. there exists a linear pattern among the *r* topics for a given concept), the output labels *Y* for *X* can be calculated through: $Y = X\alpha$ where $Y = (y_1, y_2, ..., y_r)^T$. The training data is in the following format:

$$sim(c_i, t_1), sim(c_i, t_2), ..., sim(c_i, t_r), label_i$$

Given the above *n* training examples with labels $y_i \in Y \subseteq \mathbb{R}$, our task here is predicting the label for a new coming concept vector using the function *f*. In the same way, we still do linear and ridge regression to optimize $\alpha$ and we can get the Gram matrix of the rows of matrix *X*, $G = XX^T$ as the following:

$$G = \begin{bmatrix} sim(c_1,t_1) & sim(c_1,t_2) & . & .sim(c_1,t_r) \\ sim(c_2,t_1) & sim(c_2,t_2) & ... & sim(c_n,t_r) \\ ... & ... & ... & ... \\ sim(c_n,t_1) & sim(c_n,t_2) & . & .sim(c_n,t_r) \end{bmatrix} \times \begin{bmatrix} sim(c_1,t_1) & sim(c_2,t_1) & . & .sim(c_n,t_1) \\ sim(c_1,t_2) & sim(c_2,t_2) & ... & sim(c_n,t_2) \\ ... & ... & ... & ... \\ sim(c_1,t_r) & sim(c_2,t_r) & . & .sim(c_n,t_r) \end{bmatrix} \quad (4.18)$$

Where $G_{ij} =< c_i, c_j >$ represents the relationship between $c_i$ and $c_j$:

$$\begin{aligned} & G_{ij} =< c_i, c_j > \\ & = sim(c_i,t_1)sim(c_j,t_1) + sim(c_i,t_2)sim(c_j,t_2) + \cdots + sim(c_i,t_r)sim(c_j,t_r) \\ & = \sum_{k=1}^{r} sim(c_i,t_k)sim(c_j,t_k) \end{aligned} \quad (4.19)$$

Therefore, the answer for the second question is: representing the relationships between two concepts $c_i$ and $c_j$ using the dot product of $c_i$ and $c_j$ can minimize the loss of the predictive function.

In summary, the representation method for capturing topic relationships plays an important role in discovering potential patterns among all the concepts; and the representation method for capturing concept relationships plays also an important role in discovering potential patterns among all the topics. However, the kernel methods show great advantages during this pattern discovery process, since if a linear pattern exists among all of the concepts, the semantic kernel $\phi(T_1)\phi(T_2)^T$ is able to model the semantic relationships between $T_1$ and $T_2$ in the most accurate way.

### 4.3. Proximity Matrix for Concept Vector Enrichment

There are different approaches to construct the proximity matrix *P*. In this work, we take advantage of two types of knowledge derived from Wikipedia (Wikipedia article content and Wikipedia categories) to define *P*.

### 4.3.1. The Proximity Matrix in VSM

The content of Wikipedia articles, as the most important information resource, provides immense knowledge for us to address the limitations of the VSM document representation. In addition, it is demonstrated that two semantically related concepts such as *Distributed Computing* and *Cloud Computing* intend to share more categories in Wikipedia. Therefore, the semantic relatedness between two concepts can be measured by comparing the number of common Wikipedia categories shared by them. In this section we introduce in detail how we utilize the article content and categories to construct the proximity matrix. The dimension of the proximity matrix is determined based on the dimension of the given topic representation. Suppose the topic *T* is represented by a weighted vector of words $\phi(T) =< w_1, w_2, ..., w_n >$ using the VSM where $w_i$ is a word related to *T* appearing in the documents, we then define the proximity matrix *P* for the topic *T* as shown in Figure 4.1. Given that there are *n* number of words related to *T*, the corresponding proximity matrix *P* is defined as a n-by-n matrix.

| | $w_1$ | $w_2$ | ⋯ | $w_n$ |
|---|---|---|---|---|
| $w_1$ | 1 | x | ⋯ | y |
| $w_2$ | x | 1 | ⋯ | z |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $w_n$ | y | z | ⋯ | 1 |

Figure 4.1. The Proximity Matrix in VSM

As shown in Figure 4.1, we can see that *P* is a symmetrical matrix and the diagonal entries are all equal to 1, since the semantic relatedness between a word and itself is always 1. For those off-diagonal entries, using different semantic relatedness measures leads to different

types of proximity matrices. Here we employ two types of semantic relatedness measures: the first measure is the adapted Explicit Semantic Analysis (ESA) by [77] which computes semantic relatedness between words using the content of Wikipedia articles; the second measure is using the proposed Category Space Vectors [79] based on categorical information derived from Wikipedia for semantic relatedness calculation. After all of the off-diagonal entries are obtained, they are then normalized by the maximum value within the proximity matrix.

Formally, the first measure for constructing the proximity matrix *P* using Wiki article content is defined as below:

$$P_{i,j} = \begin{cases} 1 & if \quad i = j \\ Sim_{\text{content}}(c_i, c_j) / Sim\_Max_{\text{content}} & if \quad i \neq j \end{cases} \tag{4.20}$$

Where $Sim_{\text{content}}(c_i, c_j)$ is the adapted ESA similarity [77] between $c_i$ and $c_j$, and $Sim\_Max_{\text{content}}$ is the maximum value in *P* besides the entries on the diagonal line. Under the ESA method [19], each article in Wikipedia is treated as a concept, and each topic of interest (e.g. a person name) is represented by an interpretation vector containing related Wikipedia concepts (articles). Thus, the semantic relatedness between two topics is measured by calculating the angle of the two interpretation vectors. Since the original ESA method is subject to the noise concepts introduced in the interpretation vector, we further introduce a pruning and validation step through an application of a sequence of devised heuristics for noise removal [77]. After the pruning step, we are able to use the resulting interpretation vectors to compute similarities between any two concepts.

Wikipedia categories are utilized in the second measure. Based on the assumption that those concepts (articles) sharing similar categories may be closer to each other in terms of semantic relatedness, a Wikipedia category interpretation vector [79] has been built for each

desired Wiki concept, and the semantic relatedness between two concepts of interest is determined by the percentage of common categories shared by the two corresponding category interpretation vectors. The proximity matrix *P* built using the second measure is defined as below:

$$P_{i,j} = \begin{cases} 1 & if \quad i = j \\ Sim_{category}(c_i, c_j) / Sim\_Max_{category} & if \quad i \neq j \end{cases} \tag{4.21}$$

Where $Sim_{category}(c_i, c_j)$ is the category-based similarity between $c_i$ and $c_j$, and $Sim\_Max_{category}$ is the maximum value in *P* besides the on-diagonal entries.

### 4.3.2. Variations of the Proximity Matrix

In the above section, we only employ the VSM to represent a given topic and thus the features derived from Wikipedia are not able to be embedded into the corresponding proximity matrix. Given a topic *T* of interest, we represent it as a weighted vector with enriched Wikipedia concepts: $\phi(T) = << w_1, w_2, ..., w_n >, < c_1, c_2, ..., c_m >>$ where $w_i$ is a topic-related word contained in the document collection, and $c_i$ is a relevant Wikipedia concept retrieved for the given topic. Then we define the proximity matrix for the topic *T* as illustrated in Figure 4.2.

With relevant Wikipedia concepts embedded into the topic representation, the proximity matrix is composed of four sub-matrices as shown in Figure 4.2.

1) The word-to-word sub-matrix: the upper left sub-matrix in Figure 4.2 is a symmetrical matrix with all of the diagonal entries being 1 and off-diagonal entries representing the similarities between words appearing in the documents.
2) The word-to-concept (or concept-to-word) sub-matrix: the upper right and lower left matrices represent the similarities between a word in the documents and a concept

retrieved from Wikipedia. Note that they are actually the same matrix since we have $similarity(w_i, c_j) = similarity(c_j, w_i)$.

3) The concept-to-concept sub-matrix: the lower right matrix capturing the similarity between two Wikipedia concepts is also a symmetrical matrix with diagonal entries being 1 and off-diagonal entries being the similarities between two Wikipedia concepts.

| | $w_1$ | $w_2$ | ⋯ | $w_n$ | $c_1$ | $c_2$ | ⋯ | $c_m$ |
|---|---|---|---|---|---|---|---|---|
| $w_1$ | 1 | $s(w_1,w_2)$ | ⋯ | $s(w_1,w_n)$ | $s(w_1,c_1)$ | $s(w_1,c_2)$ | ⋯ | $s(w_1,c_m)$ |
| $w_2$ | $s(w_2,w_n)$ | 1 | ⋯ | $s(w_2,w_n)$ | $s(w_2,c_1)$ | $s(w_2,c_2)$ | ⋯ | $s(w_2,c_m)$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $w_n$ | $s(w_n,w_1)$ | $s(w_n,w_2)$ | ⋯ | 1 | $s(w_n,c_1)$ | $s(w_n,c_2)$ | ⋯ | $s(w_n,c_m)$ |
| $c_1$ | $s(c_1,w_1)$ | $s(c_1,w_2)$ | ⋯ | $s(c_1,w_n)$ | 1 | $s(c_1,c_2)$ | ⋯ | $s(c_1,c_m)$ |
| $c_2$ | $s(c_2,w_1)$ | $s(c_2,w_2)$ | ⋯ | $s(c_2,w_n)$ | $s(c_2,c_1)$ | 1 | ⋯ | $s(c_2,c_m)$ |
| ⋮ | ⋮ | ⋮ | ⋱ | | ⋮ | ⋮ | ⋱ | ⋮ |
| $c_m$ | $s(c_m,w_1)$ | $s(c_m,w_2)$ | ⋯ | $s(c_m,w_n)$ | $s(c_m,c_1)$ | $s(c_m,c_2)$ | ⋯ | 1 |

Figure 4.2. The Proximity Matrix with Enriched Features

The value of each entry in the 4 sub-matrices constituting the enriched proximity matrix in Figure 4.2 is calculated using the adapted ESA method [77] or the categorical information provided by Wikipedia [79].

### 4.3.3. The Hybrid Proximity Matrix

Another important issue that we intend to address is the semantic gaps between different information resources used to measure the semantic relatedness between concepts. Linear

combination of various semantic relatedness computing measures is one of the commonly used solutions [73; 77, 79]. However, tuning parameters associated with each weighting scheme are required to distinguish between the contributions of constituent weighting schemes. It also intends to introduce too much residual during the experiments of choosing appropriate values for tuning parameters.

In order to minimize the residual, a fair amount of tests need to be conducted for adjusting the value of each tuning parameter. Therefore, the performance of the combined final weighting scheme largely depends on the accuracy of manually adjusted parameters. Suppose the proximity matrix built using the content of Wikipedia articles is $P_{content}$, and the proximity matrix built using the categorical information derived from Wikipedia is $P_{category}$, we define a hybrid proximity matrix as shown in Figure 4.3 that smoothly resolves the semantic gaps between Wikipedia articles and categories through multiplying the two proximity matrices: $P_H = P_{content} P_{category}$. For simplicity, we do not distinguish between the words in documents and the concepts from Wikipedia in the hybrid proximity matrix. The hybrid proximity matrix $P_H$ is capable of taking both Wikipedia article content and categories into consideration. It is defined as below:

$$P_{i,j} = \begin{cases} 1 & if \quad i = j \\ Sim_H(c_i, c_j) / Sim\_Max_H & if \quad i \neq j \end{cases} \tag{4.22}$$

Where $Sim_H(c_i, c_j) = \sum_{i,j=1}^{n} Sim_{content}(c_i, c_j) \cdot Sim_{category}(c_j, c_i)$ is the combined similarity between $c_i$ and $c_j$, and $Sim\_Max_H$ is the maximum value in *P* besides the on-diagonal entries.

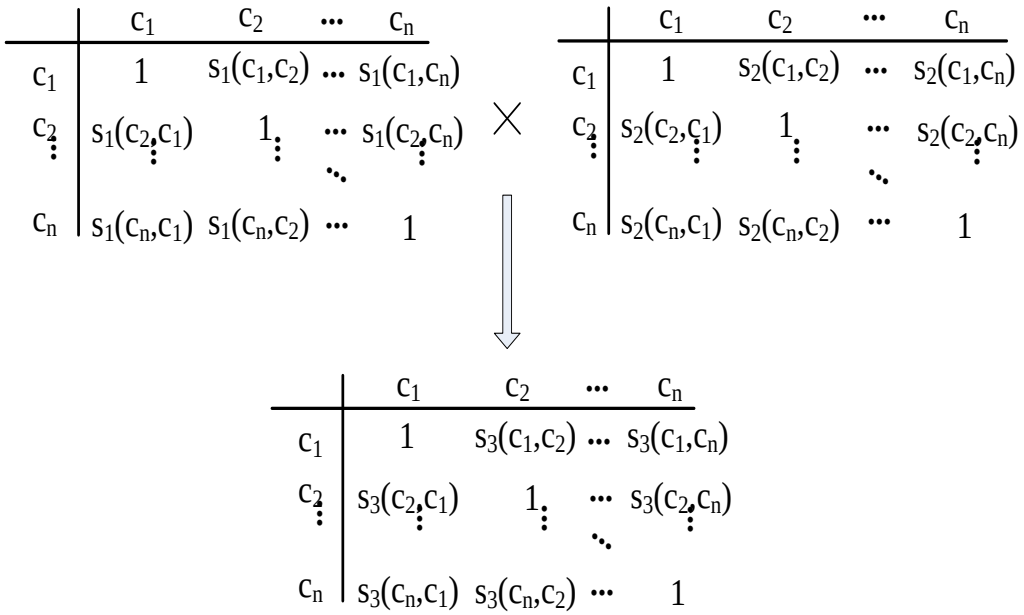


Figure 4.3. The Hybrid Proximity Matrix

## 4.4. Kernel Method for Topic Representation

### 4.4.1. Document-Level Concept Vector Update

As shown in Figure 4.1 the proximity matrix built in the VSM space only contains the concepts appearing in the documents. Using the VSM, we can represent a topic of interest using a number of related document concepts. The importance of each concept is measured based on the TFIDF weighting scheme which ignores word semantics. As introduced in the previous section, two types of semantic kernels (the article content-based kernel and the category-based kernel) can be used to enrich the original VSM representation based on the document collection and help compute the semantic relatedness between words. We give one simple example to illustrate the process of using the copus level kernels to improve semantic relatedness computing. Given the concept "*Clinton*" as a topic of interest, the document-level concept vector for "*Clinton*" and the corresponding proximity matrix are shown in Figure 4.4 and Figure 4.5. Figure

4.6 illustrates the improvement through multiplying the document-level concept vector by the proximity matrix. This is consistent with our understanding that *Hillary* as *Clinton*'s wife should be considered most related to him. *Shelton*, who served as the chairman of the Joint Chiefs of Staff during *Clinton*'s term of office, stays in the second position. At last, *Clancy*, who hardly has a relationship with *Clinton* is degraded to the end of the vector.

| | Clancy | Shelton | Hillary |
|---|---|---|---|
| Clinton | 0.54 | 0.43 | 0.38 |

Figure 4.4. The Concept Vector for the Topic "*Clinton*"

| | Clancy | Shelton | Hillary |
|---|---|---|---|
| Clancy | 1 | 0.113 | 0.147 |
| Shelton | 0.113 | 1 | 1 |
| Hillary | 0.147 | 1 | 1 |

Figure 4.5. The Wikipedia Article Content-based ProximityMatrix for the Topic "*Clinton*"

| | Hillary | Shelton | Clancy |
|---|---|---|---|
| Clinton | 0.889 | 0.871 | 0.644 |

Figure 4.6. The Improved Concept Vector for the Topic "*Clinton*"

We apply the document-level semantic kernel to the early steps of the *SPC* in the following steps:

- Step 1: Conduct independent searches for A and C. Build the A and C profiles. Call these profiles AP and CP respectively.
- Step 2: Compute a B profile (BP) composed of terms in common between AP and CP. The corpus-level weight of a concept in BP is the sum of its weights in AP and CP. This is the first level of intermediate potential concepts generated from the text corpus.

- Step 3: Build the document-level semantic kernels for topics A and C, and update the weight of each concept in BP using the proximity matrix.
- Step 4: Expand the concept chains using the created BP profile together with the topics to build additional levels of intermediate concept lists DP and EP which (i) connect the topics to each concept in BP profile in the sentence level within each semantic type, and (ii) normalize and rank them.
- Step 5: Build the document-level semantic kernels for DP and EP respectively by following the same way in Step 3, and then update the weight of each concept in DP and EP.

### 4.4.2. Document-Level Concept Vector Enrichment with the Content-based Proximity Matrix

However, we observe the improvement achieved by using document-level semantic kernels is still limited to the space of the input document collection. We now demonstrate how the proximity matrix can be used to enrich the topic representation using Wikipedia knowledge. Suppose a topic $T$ is represented by a weighted vector containing words appearing in the documents and concepts generated from Wikipedia as below:

$$\phi(T) = << tfidf(w_1), tfidf(w_2), ..., tfidf(w_n) >, < tfidf(c_1), tfidf(c_2), ..., tfidf(c_m) >> \quad (4.23)$$

Where $tfidf(w_i)$ is the TFIDF value of the word $w_i$ in the documents and $tfidf(c_i)$ is the TFIDF value of Wikipedia concept $c_i$ over Wikipedia data. Before we do the enrichment, the TFIDF values of all Wikipedia concepts equal zero as shown in Figure 4.7 if they do not appear in the

documents. For example, suppose the input topic is "*Abdel Rahman*" and it does not occur in the document collection, then its representation is as follows in Figure 4.7.

| | **Abdullah _Azzam** | **Bin_Ladin** | **New_York_City_ Landmark_Bomb_ Plot** | **Maktab_al-Khidamat** |
|---|---|---|---|---|
| **Abdel Rahman** | **0.22** | **0.12** | **0.0** | **0.0** |

Figure 4.7. The Concept Vector for the Topic "*Abdel Rahman*"

In the concept vector built for the topic "*Abdel Rahman*", the first entry "*Abdullah_Azzam*" is a highly influential Palestinian Sunni Islamic scholar and theologian. He is also known as a teacher and mentor of *Osama bin Laden* who was the founder of al-Qaeda and responsible for the September 11 attacks. The second entry is an alternative spelling of the name of *Osama bin Laden*. The third entry "*New_York_City_Landmark_Bomb_Plot*" is a planned follow-up to the February 1993 World Trade Center bombing designed to inflict mass casualties on American soil by attacking well known landmark targets throughout New York City in the United States."*Abdel Rahman*" is one of the conspirators of it. The last entry "*Maktab_alKhidamat"* was the forerunner to al-Qaeda which was founded in 1984 by *Abdullah Azzam* and *Osama bin Laden* to raise funds and recruit foreign mujahidin for the war against the Soviets in Afghanistan. Since the last two entries i.e. "*New_York_City_Landmark_Bomb_Plot*" and "*Maktab_al-Khidamat*" do not appear in the documents, their corresponding TFIDF values are set to zero.

With the concept vector built for the topic "*Abdel Rahman*", we define the proximity matrix using Wikipedia knowledge as shown in Figure 4.8. In Figure 4.8, each entry in the proximity matrix is calculated using the method we discussed in Section 4.3.2. After obtaining the proximity matrix for the given topic "*Abdel Rahman*", we multiply its concept vector by the

proximity matrix and get the new concept vector with enriched features derived from Wikipedia as shown in Figure 4.9.

| | Abdullah_ Azzam | Bin-Ladin | New_York_City_ Landmark_Bomb_ Plot | Maktab_al-Khidamat |
|---|---|---|---|---|
| **Abdullah_Azzam** | **1** | **1** | **0.0006** | **0.0024** |
| **Bin-Ladin** | **1** | **1** | **0.0009** | **0.0021** |
| **New_York_City_ Landmark_Bomb_ Plot** | **0.0006** | **0.0009** | **1** | **0.0** |
| **Maktab_al-Khidamat** | **0.0024** | **0.0021** | **0.0** | **1** |

Figure 4.8. The Article Content-based Proximity Matrix for the Topic "*Abdel Rahman*"

| | Abdullah_ Azzam | Bin_Ladin | New_York_City_ Landmark_Bomb_ Plot | Maktab_al-Khidamat |
|---|---|---|---|---|
| **Abdel Rahman** | **0.34** | **0.34** | **0.0002** | **0.0008** |

Figure 4.9. The Enriched Concept Vector for the Topic "*Abdel Rahman*"

Therefore, by integrating the Wikipedia article content-based proximity matrix into the representation of a given topic, the original concept vector built using the document-level VSM can be enriched with new concepts from Wikipedia even if the newly introduced concepts do not appear in the document texts literally.

### 4.4.3. Document-Level Concept Vector Enrichment with the Category-based Proximity Matrix

As mentioned earlier, Wikipedia categories in addition to articles can also used to enrich the topic representation. The following example shows how the category-based proximity matrix

can be utilized to achieve this goal. For example, given the topic: "*Blind Sheikh*", the corresponding document-level concept vector is shown in Figure 4.10.

| | Khallad | Salameh | Islamic_Terrorism | Jihadist_Organizations |
|---|---|---|---|---|
| **Blind Sheikh** | **0.20** | **0.16** | **0.0** | **0.0** |

Figure 4.10. The Concept Vector for the Topic "*Blind Sheikh*"

The first two entries "*Khallad*" and "*Salameh*" have non-zero TFIDF values as they occur in the input documents. The last two entries "*Islamic_Terrorism*" and "*Jihadist_Organization*" have zero TFIDF values as they do not appear in the input documents. We define the proximity matrix using the Wikipedia categories as shown in Figure 4.11.

| | Khallad | Salameh | Islamic_Terrorism | Jihadist_Organizations |
|---|---|---|---|---|
| **Khallad** | **1** | **0.73** | **0.82** | **0.39** |
| **Salameh** | **0.73** | **1** | **1** | **0.53** |
| **Islamic_Terrorism** | **0.82** | **1** | **1** | **0.81** |
| **Jihadist_Organizations** | **0.39** | **0.53** | **0.81** | **1** |

Figure 4.11. The Category-based Proximity Matrix for the Topic "*Blind Sheikh*"

We multiply the concept vector for the given topic "*Blind Sheikh"* by the corresponding proximity matrix, and then obtain a new concept vector with two enriched features "*Islamic_Terrorism*" and "*Jihadist_Organizations*" as shown in Figure 4.12.

| | Khallad | Salameh | Islamic_Terrorism | Jihadist_Organizations |
|---|---|---|---|---|
| **Blind Sheikh** | **0.32** | **0.31** | **0.32** | **0.16** |

Figure 4.12. The Enriched Concept Vector for the Topic "*Blind Sheikh*"

Therefore, with the proximity matrix built using Wikipedia categories, the original document-level VSM concept vector can be enriched with new features that are categorically related to our topic of interest.

### 4.4.4. Document-Level Concept Vector Enrichment with the Hybrid Proximity Matrix

As discussed earlier, to address the semantic gap between Wikipedia articles and categories, the hybrid proximity matrix that has both the content of Wikipedia articles and categorical information embedded can be utilized. On the other hand, using the hybrid proximity matrix, we are capable of introducing Wikipedia concepts to the topic representation, which are either article content related or categorical composition related. For example, suppose our topic of interest is "*Ayman Zawahiri*", and it is represented as a weighted vector as shown in Figure 4.13.

| | **Bin_Ladin** | **Essam_al-Qamari** | **Ali_Sayyid_Muhamed_Mustafa_al-Bakri** | **Abdullah_Yusuf_Azzam** | **Afghan_Civil_War** |
|---|---|---|---|---|---|
| **Ayman Zawahiri** | **0.71** | **0.0** | **0.0** | **0.0** | **0.0** |

Figure 4.13. The Concept Vector for the Topic "*Ayman Zawahiri*"

In Figure 4.13, the first entry "*Bin_Ladin*" is a concept appearing in the documents literally, and thus has a non-zero TFIDF value of 0.71. The second and third entries ("*Essam_al-Qamari*" and "*Ali_Sayyid_Muhamed_Mustafa_al-Bakri*") are relevant Wikipedia articles, and the last two entries ("*Abdullah_Yusuf_Azzam*" and "*Afghan_Civil_War*") are relevant Wikipedia categories.

Following the method discussed previously, we defined the hybrid proximity matrix for the given topic as shown in Figure 4.14.

| | Bin_Ladin | Essam_al-Qamari | Ali_Sayyid_Muhamed_Mustafa_al-Bakri | Abdullah_Yusuf_Azzam | Afghan_Civil_War |
|---|---|---|---|---|---|
| **Bin_Ladin** | **1** | **0.82** | **0.97** | **1** | **0.53** |
| **Essam_al-Qamari** | **0.82** | **1** | **0.35** | **0.36** | **0.15** |
| **Ali_Sayyid_Muhamed_Mustafa_al-Bakri** | **0.97** | **0.35** | **1** | **0.66** | **0.38** |
| **Abdullah_Yusuf_Azzam** | **1** | **0.36** | **0.66** | **1** | **0.50** |
| **Afghan_Civil_War** | **0.53** | **0.15** | **0.38** | **0.50** | **1** |

Figure 4.14. The Hybrid Proximity Matrix for the Topic "*Ayman Zawahiri*"

Similarly, through multiplying the concept vector for the given topic "*Ayman Zawahiri*" by the hybrid proximity matrix, we obtain a new concept vector with four enriched features as illustrated in Figure 4.15. Note that using the hybrid proximity matrix, "*Abdullah_Yusuf_Azzam*", who has deep influence on "*Bin Ladin*" and is considered as the Father of Global Jihad, is weighted as the top related concept to "*Ayman Zawahiri*" who is in the list of FBI most wanted terrorists.

| | Bin_Ladin | Essam_al-Qamari | Ali_Sayyid_Muhamed_Mustafa_al-Bakri | Abdullah_Yusuf_Azzam | Afghan_Civil_War |
|---|---|---|---|---|---|
| **Ayman Zawahiri** | **1.66** | **1.37** | **1.62** | **1.66** | **0.87** |

Figure 4.15. The Enriched Concept Vector for the Topic "*Ayman Zawahiri*"

## 4.5. A MapReduce Solution for Proximity Matrix Utilization

### 4.5.1. MapReduce Overview

MapReduce [12], introduced by Google, is a programming model for processing large data sets using a cluster or a grid. It is specifically designed for processing parallelizable problems where the computational cost might be prohibitive or unacceptable using traditional solutions. The main process of using MapReduce to solve a problem is typically composed of two steps: the "*Map*" step which divides a problem into smaller sub-problems and distributes them to worker nodes; the "*Reduce*" step which collects the answers to all the sub-problems and combines them in some way to form the final answer. Figure 4.16 gives an overview of the MapReduce programming model.

The MapReduce framework operates exclusively on <key, value> pairs. Specifically, The "Map" function takes a set of <key, value> pairs as input to a MapReduce job, and produces a set of <key, value> pairs as the output of the job.

A widely used MapReduce implementation is the Apache Hadoop [2] framework that supports data-intensive distributed applications. A MapReduce job defined in Hadoop usually divides the input data into a number of chunks which are stored in Hadoop Distributed File System (HDFS) and processed by the map tasks in a completely parallel manner to improve computing performance. Then Hadoop sorts the outputs of the maps based on user-defined keys, and distributes the sorted outputs to the reduce tasks. The framework automatically takes care of scheduling, monitoring and executing tasks, so that users can focus on the problem itself, e.g. problem dividing, sub-problem merging, etc. Applications using MapReduce include MapReduce enabled classification, clustering [54] and so on.

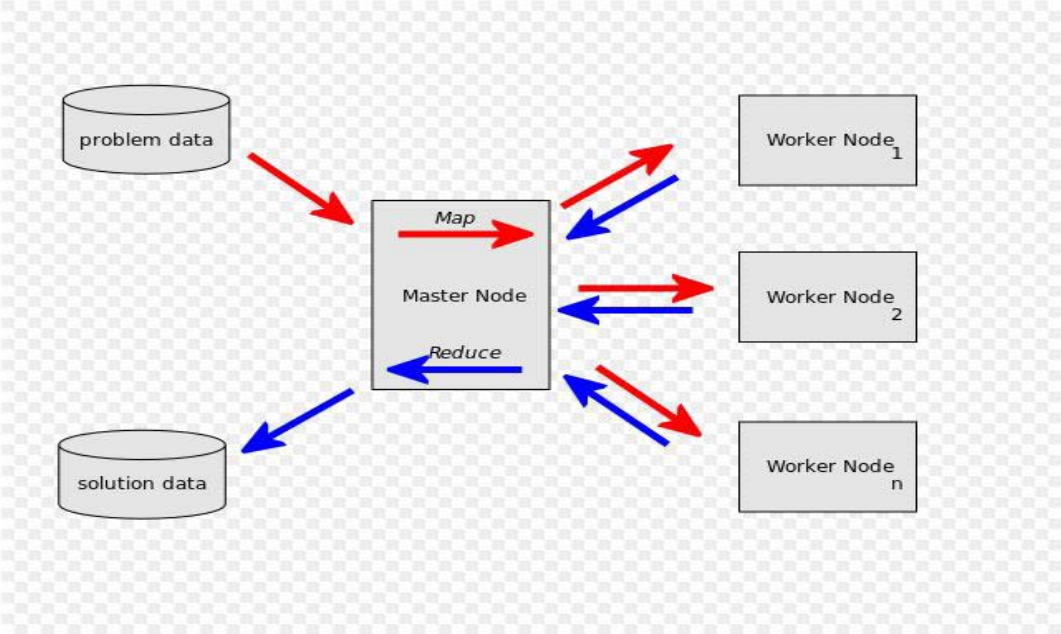


Figure 4.16. MapReduce Overview [Wikipedia, 2012]

### 4.5.2. Proposed MapReduce Algorithm for Document-Level Concept Vector Enrichment

This section introduces a MapReduce solution for enriching a concept vector using its corresponding proximity matrix derived from Wikipedia. We formulate this process of enrichment as an optimization problem. The problem we intend to address here is characterized by being both data-intensive and compute-intensive as we take advantage of over 5,000,000 Wiki articles and 700,000 Wikipedia categories. Out of the consideration of finding a reasonable trade-off point between the data and the computation, we employ semantic types (ontological information) as a data/task splitter for this optimization problem because: (i) it keeps all concepts belonging to the same semantic type ordered based on both document and background knowledge; (ii) the problem of large matrix multiplication is transformed into small sub-matrix multiplication that can be parallelized.

In general, a semantic profile is first uploaded to the HDFS and then passed to the MapReduce job for concept enrichment. Specifically, the *Mappers* receive concepts with the

semantic type as the *Map* key and the weight as the *Map* value, and emit them to the *Reducers*. The *Reducers* collect all concepts belonging to the same semantic type, build the corresponding proximity matrix for them, update the weight of each concept, and at last output a list of reorder concepts. The detailed procedure is illustrated in Figure 4.17.

```
/* Emit each semantic type plus an associated concept belonging to it. */
function Map (Key: SemanticType, Value: WeightedConcept)
        Emit(Key, Value)
end function

/* Collect a list of weighted concepts belonging to a semantic type, and then build the corresponding
proximity matrix for the collected concepts. */
function Reduce (Key: SemanticType, ValueList: WeightedConcept)
        // Initialization:
        initialize the ESA environment
        Dim =  # of weighted concepts in ValueList
        String[] conceptNameVector
        double[] conceptWeightVector
        double[][] proximityMatrix
        for each concept ci in ValueList
                conceptNameVector[i] = ci.getName()
                conceptWeightVector[i] = ci.getWeight()
        end for
        // Start building the ESA-based kernel matrix
        for each i in Dim
           for each j in Dim && j <= i
              if (i == j)
                 proximityMatrix[i][i] = 1.0
                 continue
              end if
              similarityESA = computeESASimilarity(conceptNameVector[i], conceptNameVector[j])
              proximityMatrix[i][j] = similarityESA
              proximityMatrix[j][i] = similarityESA
           end for
        end for
        // Kernel matrix normalization
        normalizeConceptWeight(kernelMatrix)
        // Weight update
        conceptWeightVector = conceptWeightVector * proximityMatrix
        // Reorder concepts according to the updated weights
        rankConcepts(conceptWeightVector);
end function
```

Figure 4.17. MapReduce Algorithm for Enriching a Concept Vector with the Corresponding Wikipedia-based Proximity Matrix

## 4.6. Summary

To summarize, our focus in this chapter has been on representing the semantic relationships between two concepts in a more appropriate and efficient way in our query context. Compared with approaches [73, 77, 79] that linearly combine different weighting schemes associated with tuning parameters for each constituent, we embed various information resources from Wikipedia into semantic kernels that on one hand smoothly resolve the semantic gaps between different approaches, and on the other hand avoid human intervention in constructing a final weighting scheme. Theoretical analysis has been given to demonstrate the effectiveness of the proposed semantic kernels, and different kernel variations have been designed to meet different knowledge discovery needs.

# CHAPTER 5. RELATIONSHIP MINING WITH CONCEPT ASSOCIATION GRAPH

## 5.1. Motivations

This chapter presents a graph-based mining model. The *SPC* model starts with two given topics of interest, builds multiple levels of semantic profiles and links from one of the two topics to the other by going through each semantic profile. It is in essence a profile-based mining model and the resulting paths between the two topics must go along the semantic profiles. We call this search process a profile-guided search. There are two major drawbacks of the profile-guided search: i) due to the combinatorial explosion of enumerating all possible concepts contained in all intermediate semantic profiles between two given concepts, the length of the resulting relationship chains considered in this research is limited to 4; ii) the search is not flexible enough since the search path must strictly go through concepts in one semantic profile to another. For example, suppose A → B → C is a generated relationship chain linking topic A to topic C, B as a subsequent concept after A must be within the semantic profile built between A and C. To address such problems, we propose a graph-based mining model in this chapter. Jin et al has proposed a graph-based approach in [31] which applied association rule mining techniques [1], but their approach needs data preparation in advance while our approach does not require any domain specific lexicon to be built beforehand since Wikipedia knowledge can be automatically employed in the discovery process and serves as our background knowledge repository. Using the graph-guided search, the length of the resulting relationship chains has no limitation unless explicitly specified by the user, and the search will go along the next most relevant concept until reaching the destination topic or the length of the chain has exceeded the limit defined by the

user. Note that the graph-based mining model here is able to answer queries against two topics based on different user needs. For example, one typical search interest might be concentrating on finding the strongest association connecting two concepts, while attention for another search might be centered on finding the *T*-best paths from the source concept to the destination concept. In comparison to traditional document-based mining models where documents are usually domain-specific, the model proposed here is capable of handling a significant amount of queries across domains without being limited to document collections. We implemented an interactive visualization paradigm which assists users for a better understanding and interpretation of the discovered associations.

The architecture of the proposed system is illustrated in Figure 5.1. There are mainly two components, i.e. the graph construction module and search module. We introduce each component in detail in the following sections.

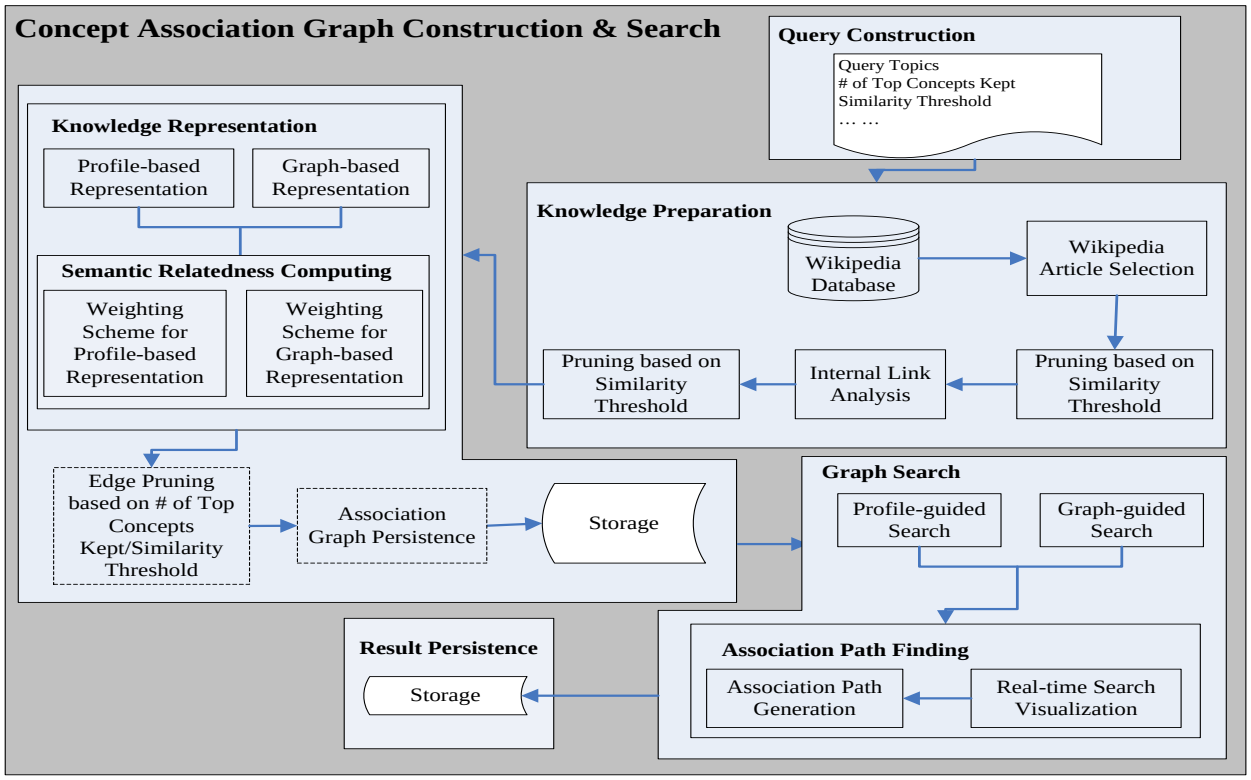


Figure 5.1. System Overview

The approach proposed in this chapter differs from previous approaches in three ways.

1) In comparison to [30, 67, 71], we represent concept associations as a graph with the nodes representing concepts and edges between them representing associations. This is meaningful since the solution being pursued here takes the view that the concepts and associations in a particular domain can be represented as a network [33].
2) The approaches in [31, 32, 33] are coupled with Semantex [66], an Information Extraction (IE) engine for extracting the features (i.e., words or phrases) from document collections as candidate nodes forming the association graph to be built, which decreases the reusability and portability of their approach when applied in mining tasks where the information extraction engine Semantex is helpless or even unavailable. In addition, their approaches pre-determined the constituent nodes and the structure of the graph, which restricts the search capacity to only those pre-determined concepts. While in our system:
    - Data preparation in advance is not required at all compared with [31, 32, 33]. The process of constructing candidate nodes is triggered after topics of interest are determined, which automatically ignores topic-irrelevant nodes and could potentially improve the search performance in terms of speed and accuracy.
    - The search space is spanned by all Wikipedia articles which is a huge extension to the VSM based on corpus level knowledge discovery.
    - Instead of only considering statistical information from text corpus e.g. mutual information[17], concept association strengths here are calculated using an improved weighting model incorporating Wikipedia knowledge.

3) Compared with RelFinder [41], a popular application using RDF knowledge bases to explore connections between concepts, our approach defines various search strategies based on different user interests and achieves better performance.

### 5.2. Query Construction and Knowledge Preparation

The Query Construction component receives search topics from end users and composes a query which will be interpreted by the Knowledge Preparation component for knowledge selection from the Wikipedia database. It is often the case that different users might have different search interests. For example, some might be only interested in the hierarchical associations (i.e. the profile-based associations may be preferable) while others might pay more attention to the relevance of associations discovered. To fit these various specific mining requirements, the query construction module provides a number of preferences for users to specify, such as the threshold of the association strength between concepts for selecting candidate concepts and the graph depth they would like to build. Considering the high cost of enumerating all relevant categories to concepts, Wiki categories are not involved in this model. In particular, we focus on i) exploring article-to-article relationship findings to achieve more fine-grained discovery results in comparison to the *SPC* model, and ii) enabling the visualization of the whole discovery process in real time so that the user can get a better understanding and interpretation of the discovered associations.

Anchor texts, another type of valuable information resource provided by Wikipedia in addition to the textual content of articles, imply rich hidden associations between different Wikipedia concepts. For example, the Wikipedia article talking about "*Osama bin Laden*" contains a great number of potential terrorists who are related to him and terrorism events that he

was involved in. Therefore, through inspecting the anchor texts in each Wikipedia article, we are able to find a fair amount of interesting concepts, and those interesting concepts constitute an important part of the *CAG* we will be building in later steps. Figure 5.2 gives part of the anchors in the article "*Osama bin Laden*".

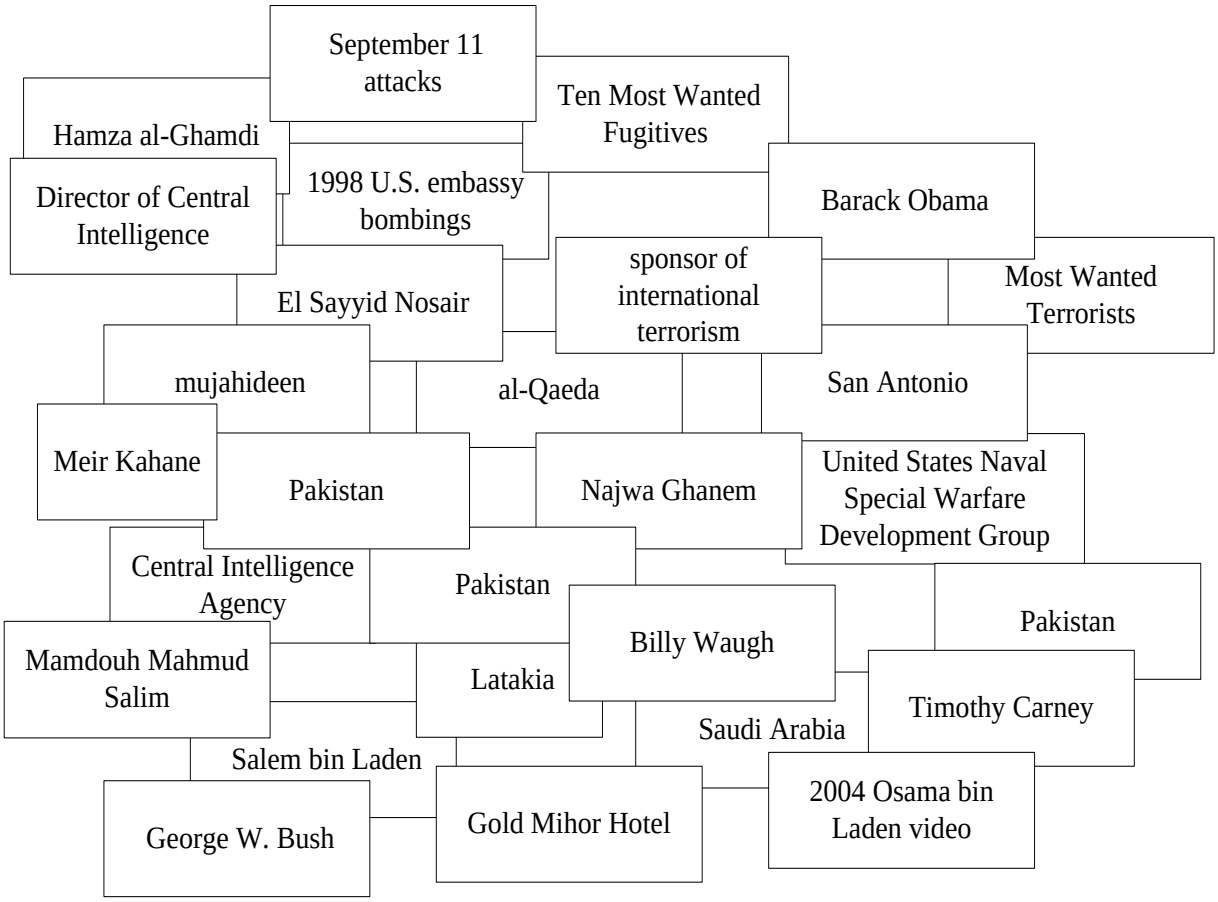


Figure 5.2. Wikipedia Anchors Related to "*Osama bin Laden*"

After a query is constructed, the system starts collecting relevant knowledge from the Wikipedia database. Here we go through a much more rigorous process compared with [77, 79] by inspecting the content and the internal links (i.e. anchor texts) of each Wikipedia article and then applying heuristic strategies with user defined preferences for noise removal.

We now summarize the process of selecting the topic-related Wikipedia concepts in Figure 5.3.

**Algorithm: select topic-related Wikipedia concepts**

```
    /*Inspecting Wikipedia concepts that mention the topics of interest*/
1:  for each c_i in DB_wiki do
2:      if c_i mentions both t_1 and t_2
3:          int n = maximum (# of words in t_1, # of words in t_2)
4:          find n-grams for the page text representing c_i
5:          for each n-gram do
6:              int dis1 = calculateDistance (t_1, n-gram)
7:              int dis2 = calculateDistance (t_2, n-gram)
8:              if dis1 <= THRESHOLD_DIS and dis2 <= THRESHOLD_DIS
9:                  c_i is considered as a candidate concept for t_1 and t_2
10:             else
11:                 c_i is considered as irrelevant to t_1 and t_2
12:             end if
13:         end for
14:     end if
15: end for
16: /*Inspecting internal page links.
17  Note that an internal page link is also treated as a concept in Wikipedia*/
18: for each candidate concept cand_i do
19:     get the internal links from the page text representing cand_i
20:         for each internal link l_i do
21:             double sim1 = calSimilarity(l_i, t_1)
22:             double sim2 = calSimilarity(l_i, t_2)
23:             if sim1 >= THRESHOLD_SIM and sim2 >= THRESHOLD_SIM
24:                 l_i is considered as a candidate concept for t_1 and t_2
25:             else
26:                 l_i is considered as irrelevant to t_1 and t_2
27:             end if
28:         end for
29:     end if
30: end for
```

Figure 5.3. Process of Selecting Topic-related Wikipedia Concepts

Under the Wikipedia database $DB_{wiki}$, each article is treated as a concept $c_i$. For a given topic $t$ of interest, $c_i$ is considered potentially relevant to $t$ if $t$ is mentioned in the article text of $c_i$.

Then an adapted Levenshtein Distance algorithm (i.e. the "*calculateDistance*()" function in Figure 5.3) is devised to measure the relevance of $c_i$ to *t*. Specifically, we take a single word as a unit to calculate the allowable edit operations between $c_i$ and *t*. If the Levenshtein distance between $c_i$ and *t* is under the defined threshold, $c_i$ is viewed as relevant. Since we observe the anchor texts appearing in a Wikipedia article often imply important relations between them and the Wikipedia concept as shown in Figure 5.2, we inspect all anchors $l_i$ within each relevant $c_i$ for candidate concept enrichment. In particular, if *calSimilarity*($l_i$, $t_1$) and *calSimilarity*($l_i$, $t_2$) are both greater than or equal to the *THRESHOLD_SIM* specified by the user, $l_i$ will be selected as a candidate concept for connecting two given topics $t_1$ and $t_2$ (note the "*calSimilarity()*" in Figure 5.3 is a function used to calculate the similarity between two concepts using our adapted ESA method). At last, all the resulting $c_i$ and $l_i$ are viewed as relevant Wikipedia concepts to the two given search topics.

## 5.3. Knowledge Representation

The Knowledge Representation Module captures the semantic relationships between concepts through our defined two representations. One is profile-based and the other is graph-based.

### 5.3.1. Profile-based Representation

The profile-based approach constructs associations between two topics $t_1$ and $t_2$ by first building a profile containing Wiki concepts (i.e. linking concepts between $t_1$ and $t_2$) whose article texts are related to $t_1$ and $t_2$ (the relatedness is measured using the following formula for computing $s_k$), and this process is also applied to build additional levels of profiles. Formally, a profile containing potential concepts connecting topics $t_1$ and $t_2$ is defined as follows:

$$profile(t_1, t_2) = (s_1c_1, s_2c_2..., s_nc_n) \quad (5.1)$$

Where $s_k$ represents the weight of the linking concept $s_k$ appearing in *profile($t_1$, $t_2$)*. Suppose $t_1$ and $t_2$ are spanned by all words appearing in them, i.e., $t_1 =< w_{11}, w_{12}, ..., w_{1i} >$ and $t_2 =< w_{21}, w_{22}, ..., w_{2j} >$, respectively, the weight $s_k$ is computed as follows:

$$s_k = \frac{1}{2}\left(\sum_{w_{1i} \in t_1} tf_{t_1}(w_{1i}) tf \cdot idf_{c_k}(w_{1i}) + \sum_{w_{2j} \in t_2} tf_{t_2}(w_{2j}) tf \cdot idf_{c_k}(w_{2j})\right) \quad (5.2)$$

Where $tf_{t_1}(w_{1i})$ is the frequency of word $w_{1i}$ in $t_1$ and $tf \cdot idf_{c_k}(w_{1i})$ is the $tf \cdot idf$ value of word $w_{1i}$ in Wikipedia article $c_k$, while $tf_{t_2}(w_{2j})$ is the frequency of word $w_{2j}$ in $t_2$ and $tf \cdot idf_{c_k}(w_{2j})$ is the $tf \cdot idf$ value of word $w_{2j}$ in Wikipedia article $c_k$. All linking concepts are ordered according to their weights and users can specify the top *N* concepts they would like to explore prior to the graph construction process.

### 5.3.2. Graph-based Representation

Instead of building multiple levels of profiles between concepts for capturing concept associations, the graph-based approach builds connections between any two concepts when the user specifies the threshold of association strength as the search criterion. Compared with the profile-based approach which represents concept associations as hierarchical profiles, the graph-based approach is much more flexible in structure.

Suppose our search topics are $t_1$ and $t_2$, formally, a *CAG* against $t_1$ and $t_2$ is defined as follows:

$$CAG(t_1, t_2) =< V, E, VAL > \quad (5.3)$$

Where $V = \{v_i \mid v_i \in R(t_1, t_2)\}$ *and* $E = \{e_{i,j} \mid sim(v_i, v_j) >= VAL\}$ given $R(t_1, t_2)$ is a set of relevant Wikipedia concepts to $t_1$ and $t_2$ and *VAL* is a real value in charge of controlling the strength of associations between concepts. The detailed steps of building a *CAG* are illustrated in Figure 5.4. A typical *CAG* built using the graph-based approach is illustrated in Figure 5.5.

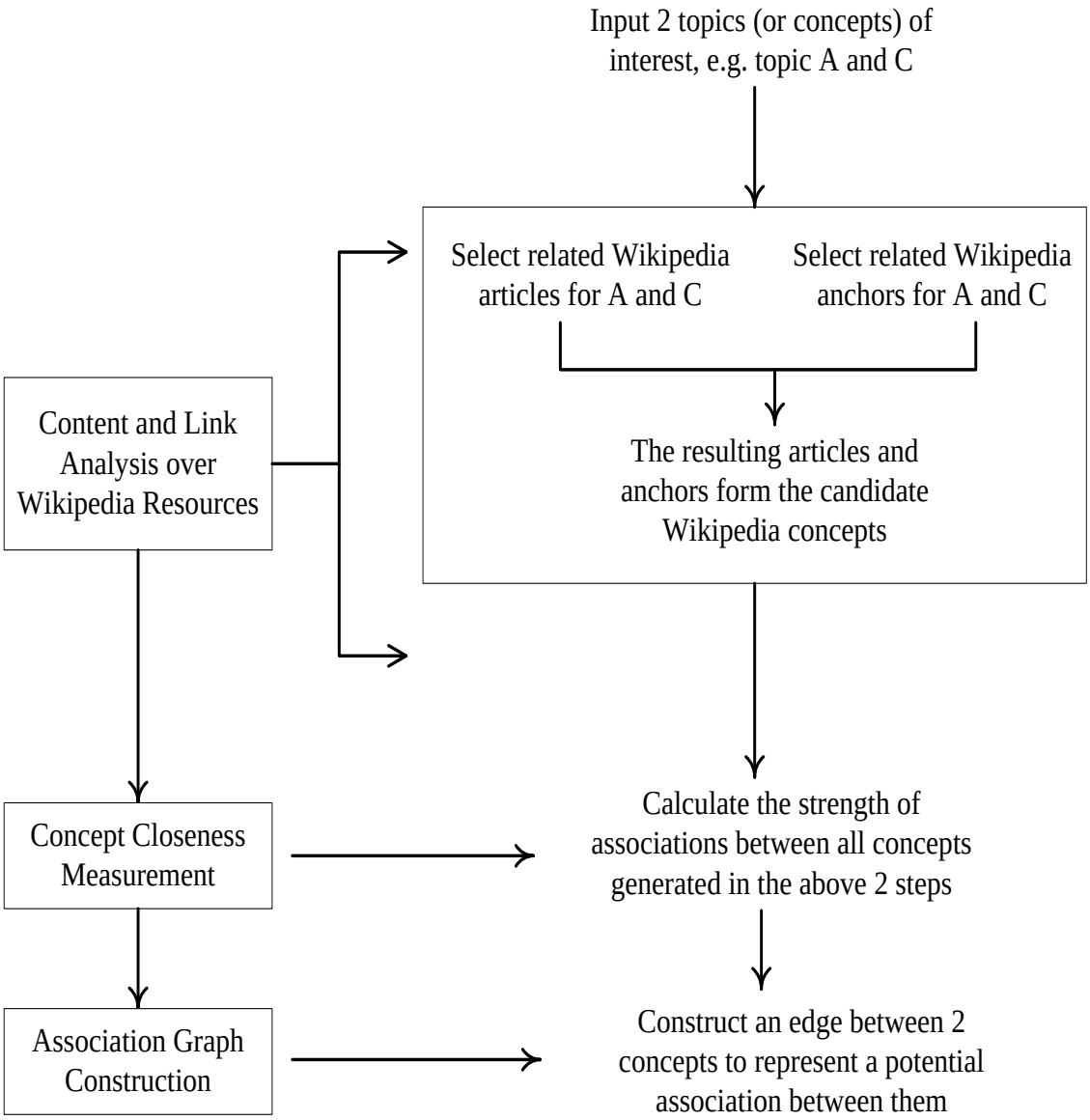


Figure 5.4. Procedure of Building a Concept Association Graph

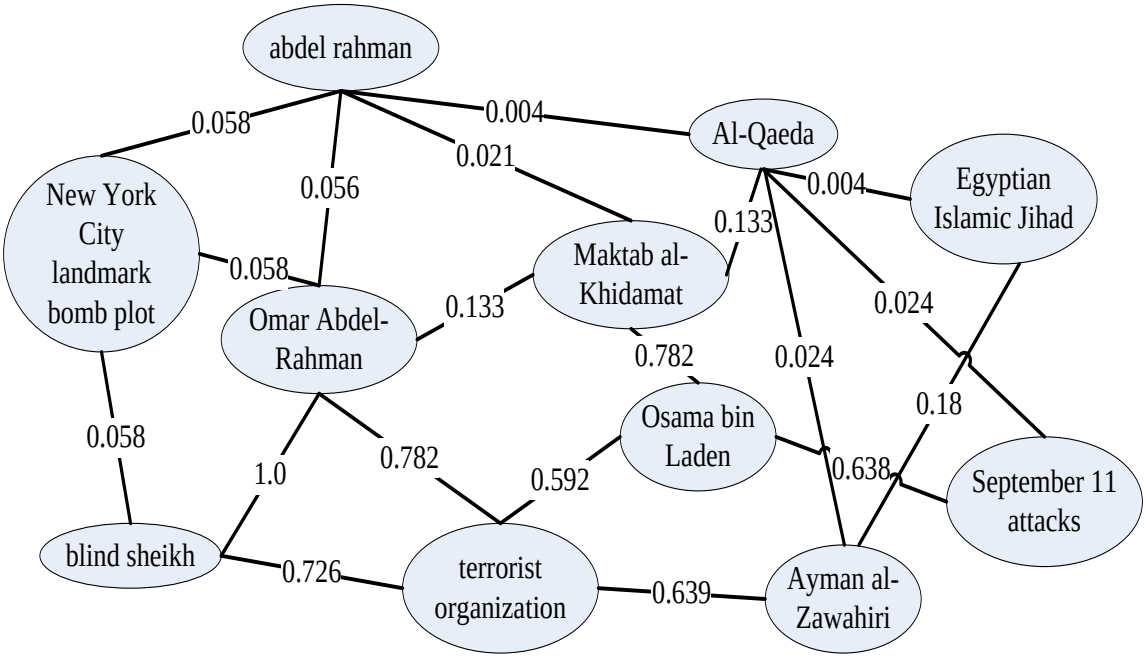


Figure 5.5. Relationships between "*Abdel Rahman*" and "*Blind Sheikh*"

To compute the semantic relatedness between two concepts, e.g., $c_1$ and $c_2$, suppose $v_1$ and $v_2$ are the two interpretation vectors built for $c_1$ and $c_2$, we first go through a sequence of heuristic steps discussed in Section 3.2.2 to remove noise from $v_1$ and $v_2$, and then compute the cosine similarity between them to capture the semantic relatedness between $c_1$ and $c_2$.

### 5.3.3. Summary

We summarize the problem of representing semantic relationships between concepts using the profile-based and graph-based approaches as illustrated in Table 5.1. No matter which approach is adopted, some of the edges representing concept relationships might be pruned according to the user specified parameters such as the number of top linking concepts kept in each profile or the threshold controlling the strength of concept associations.

Table 5.1. Comparison between the Profile-based Approach and the Graph-based Approach

| | **Profile-based Approach** | **Graph-based Approach** |
|---|---|---|
| Associations/ Edges | 1) Linking two concepts/vertices in each level of profile<br>2) capturing concept co-occurrence in Wiki articles | 1) Linking any two concepts/vertices<br>2) capturing concept semantic relatedness |
| User Action | Specifying top *N* concepts desired within each profile | Specifying a threshold based on which an association/edge is added to the graph |
| Search Route | Profile guided | Similarity guided |
| Termination Conditions | Destination topic is reached | Destination topic is reached or all remaining concepts/vertices are all unqualified (i.e. below the specified threshold) |
| Length of Resulting Associations | Up to the number of profiles built | Uncertain |
| Output | Associations from source topic to destination topic | Associations from source topic to destination topic or NULL |

## 5.4. Semantic Relationship Search

Based on the two knowledge representations discussed in the above, we present the algorithm of generating and ranking concept associations in this section. Given a source topic and a destination topic, as well as the user's requirements (such as the maximum length of resulting associations, the threshold of association strength between concepts), the proposed algorithm attempts to find (i) if there is a direct association from the source to the destination topic, or (ii) if they can be connected by several intermediate concepts (paths). In the second case, the intermediate associations are referred to as transitive associations from the source to the destination topic. We consider this problem as an optimization problem that tries to find the top strongest associations at various lengths between topics. As discussed earlier, different from

previous approaches such as [31] which 1) limits the search within domain-specific documents, and 2) gives a higher priority to association degree measured in terms of the length of the links over association strength, our search 1) is spanned by the space of Wikipedia concepts and requires no data preparation or preliminary knowledge, and 2) is more flexible by enabling the user to decide the priority (search by association degree or association strength).

### 5.4.1. The Goodness Function for Concept Association Chains

To capture salient aspects of relationships between topics, we define the function *g* that represents the "goodness" of the resulting association chains. There are various ways to define the goodness function. Through experiments, we found directly using the sum of the weights of its constituent edges as the goodness function *g* achieved the best evaluation result. Suppose a concept association chain is in the form: $c_1 \to c_2 \to ... \to c_n$, *g* is defined as follows:

$$g(c_1, c_n) = \frac{\sum_{i=1}^{n-1} as(c_i, c_{i+1})}{n} \tag{5.4}$$

Where $as(c_i, c_{i+1})$ is the strength of association calculated using our adapted ESA method. Note that this goodness function is able to give a global ranking for chains with different lengths. We call a path an optimal path from $c_1$ to $c_n$ if $g(c_1, c_n)$ returns the maximum among all the possible paths. The premise is that the goodness of a resulting chain should be penalized as its length increases.

### 5.4.2. Profile-guided Search

Starting with the source topic, the profile-guided search goes through each built concept profile and retrieves the top ranked concept within each profile until reaching the destination

topic. Given the concepts within each profile are already ordered when the knowledge representation module is completed, finding an optimal chain that maximizes the goodness function can be solved in $O(k)$ time with $O(k)$ space where *k* is the length of the chain specified by the user. However, it is often the case that the user might be interested in exploring more than one optimal chain. This problem then becomes finding *T*-best chains with path length *k*. It can be proved that the *T*-best chains must be found if we keep only the top *T* concepts within each profile. Also, enumerating all possible chains takes $O(T^{k-1})$ time with $O(T^{k-1})$ space. Then the problem of finding *T*-best chains with path length *k* can be efficiently solved using the algorithms (Figure 5.6) proposed in the next section.

### 5.4.3. Graph-guided Search

Instead of walking through concept profiles, the graph-guided search considers only the semantic relatedness between concepts and expends the most relevant concept as the search proceeds. Suppose there are *N* number of concepts in the *CAG*, the problem of finding the best chain can be solved in $O(\frac{(N-1)!}{(N-k-2)!})$ time with $O(N)$ space, where *k* is the length of the chain. If the user wants to find *T*-best chains, it can also be proved that the *T*-best chains must be found if we keep only the top *T* concepts that have the strongest association strength related to the currently being examined concept during each search step. Figure 5.6 gives two algorithms for finding the *T*-best related concepts for the current concept. Algorithm 1 takes $O(N)$ time, and Algorithm 2 takes $O((N-T)T)$ time.

**Algorithms: find *T*-best concepts**

**Algorithm 1:** T-Best-OrderStatistics

```
1:      T-Best-OrderStatistics (array candidateArr, int t)
2:          int size = # of concepts in candidateArr
3:          for i = 1 to t do
4:              SelectKthOrder(candidateArr, 0, size-1, t)
5:          end for
6:      end function
7:      SelectKthOrder (array candidateArr, int start, int end, int k)
8:          int curPos = randomly select one position from candidateArr
9:          int tmpPos = # of elements before curPos
10:         if (tmpPos == k)
11:             concept at tmpPos is the kth order statistic
12:         else if (tmpPos > k)
13:             SelectKthOrder(candidateArr, start, tmpPos-1, k)
14:         else
15:             SelectKthOrder(candidateArr, tmpPos+1, end, k-tmpPos)
16:         end if
17:     end function
```

**Algorithm 2:** T-Best-TmpArr

```
1:      T-Best-TmpArr (array candidateArr, int t)
2:          array tmpArr
3:          for i = 0 to t-1 do
4:              tmpArr [i] = candidateArr[i]
5:          end for
6:          concept_min = the least-relevant concept in tmpArr
7:          for i = t to end of candidateArr do
8:              if candidateArr [i] is more relevant than concept_min
9:                  remove concept_min from tmpArr
10:                 add candidateArr [i] to tmpArr
11:             end if
12:         end for
13:         return tmpArr
14:     end function
```

Figure 5.6. Algorithms for Finding the *T*-best Related Concepts for the Current Concept

## 5.5. Summary

In summary, this chapter presents a relationship mining model that is graph-based and directly interfaced to Wikipedia. The proposed algorithm can automatically build a Concept Association Graph (*CAG*) from Wikipedia for two given topics of interest, and generate a ranked

list of concept chains as potential associations between them. Compared with the *SPC* model presented in the previous chapter, this graph-based model has the following advantages: 1) no pre-determined documents and domain specific knowledge are required to prepare for the search process; 2) anchor texts are incorporated into the knowledge discovery process which act as an effective aid; 3) the length of chains is not longer restricted to 4; 4) more flexible search strategies can be provided to the user. However, because the algorithm designed in this chapter performs unidirectional search from the source topic to the destination topic, the association discovered may be different from the one starting from the destination topic. To tackle this situation, some parameters such as the similarity threshold between concepts are provided to balance the search and further improve the search accuracy. In general, the *SPC* model and the graph-based model emphasize different aspects of a mining task, and which one is preferable depends on specific search requirements.

# CHAPTER 6. EXPERIMENTS AND EVALUATIONS

This chapter presents the experimental results of the proposed knowledge discovery approaches in this dissertation. A challenging task for the evaluation was constructing an evaluation data set, since there are no standard data sets available for quantitatively evaluating the concept association chains. We evaluate the performance of our system by looking at:

a) How good are the proposed models in helping the user identify the potential relationships between two topics of interest?
b) How good are the proposed models in ranking important concepts?
c) Are the proposed models able to discover unobvious relationships between concepts?
d) Are the proposed models able to discover relationships not contained in the documents?

## 6.1. Processing Wikipedia Dumps

Wikipedia offers free copies of the entire content in the form of XML files [14]. It is an ever-updating knowledge base, and releases the latest dumps to interested users regularly. The version used in this work was released on April 05, 2011, which was separated into 15 compressed XML files and altogether occupied 29.5 GB after decompression. An open source tool MWDumper [50] was used to import the XML dumps into our MediaWiki database, and after the parsing process, we identified over 5 million articles and 0.7 million categories as shown in Table 6.1.

Table 6.1. Wikipedia Content of the April 05 Dump

| **Wikipedia Resource** | **Resource #** |
|---|---|
| Articles/Concepts | 5,553,542 |
| Redirected Articles/Concepts | 5,156,719 |
| Categories | 794,778 |
| Page-out Links | 215,832,350 |
| Redirect Links | 5156719 |

## 6.2. Evaluation Data

An open source document collection pertaining to the 9/11 attack, including the publicly available 9/11 commission report was used in our evaluation. The report consists of Executive Summary, Preface, 13 chapters, Appendix and Notes. Each of them was considered as a separate document resulting in 337 documents. The whole collection was processed using Semantex [66] and concepts were extracted and mapped to the counterterrorism domain ontology [28]. A significant amount of query pairs selected by the assessors covering various scenarios (e.g., ranging from popular entities to rare entities) were conducted and used as our evaluation data. We selected chains of lengths ranging from 1 to 4 in terms of the number of associations. The chains were selected by going through the same procedure with [33] as follows:

1) We chose various pairs of topics: in the counterterrorism corpus, the topics were mostly named entities.
2) For each topic pair, the relevant paragraphs for either topic respectively were then manually inspected: we selected those where there was a logical connection between the two topics.

3) After achieving agreement among all annotators, we then generated the concept chains for these topic pairs.

The above process generated 37 chains in 9/11 corpus as shown in Table 6.2 which will be used as truth chains for later experiments.

Table 6.2. Truth Chains

| **L1 (Length 1)** | |
|---|---|
| abdel_rahman::blind_sheikh | abdel_rahman → blind_sheikh |
| abdullahi_farah::jumale | abdullahi_farah→ jumale |
| adel::ffi | adel→ ffi |
| alexis::lloyd_salvetti | alexis→ lloyd_salvetti |
| american_muslim::khifa | american_muslim→ khifa |
| atta::dekkers | atta→ dekkers |
| crawford::khalilzad | crawford→ khalilzad |
| donovan::wall_street | donovan→ wall_street |
| easton_police_department::lee_ hanson | easton_police_department→ lee_hanson |
| glenn::pressler | glenn→ pressler |
| global_positioning_system::jarrah | global_positioning_system→ jarrah |
| kenya::mohamed | kenya→ mohamed |
| martha_stewart::saudi_arabia | martha_stewart→ saudi_arabia |
| saudi_arabian_ministry::thumairy | saudi_arabian_ministry→ thumairy |
| **L2 (Length 2)** | |
| abdullah::world_trade_ organization | abdullah →kingdom →world_trade_organization |
| abdullahi_farah::uae | abdullahi_farah→jumale →uae |
| ajaj::ali | ajaj→unite_state→ali |
| amal:: sudanese | amal → cia → sudanese |

Table 6.2. Truth Chains (continued)

| | |
|---|---|
| **L2 (Length 2)** | |
| betty_ong:: madeline | betty_ong→flight_attendant→ madeline |
| christopher_steele::perez | christopher_steele→pilot →perez |
| dekkers::jones_aviation | dekkers→ atta →jones_aviation |
| marty_miller:: oakley | marty_miller→ unocal → oakley |
| **L3 (Length 3)** | |
| abdullahi_farah::jumale | abdullahi_farah→al-barakaat → bin_ladin →jumale |
| amal :: sudanese | amal → cia → bin_ladin → sudanese |
| ayman_zawahiri::national_ islamic_front | ayman_zawahiri→ bin_ladin →turabi →national_islamic_front |
| ayman_zawahiri::qaeda_presence | ayman_zawahiri→ bin_ladin→taliban→qaeda_presence |
| binalshibh::pistole | binalshibh →fund → fbi →pistole |
| brian_david_sweeney::peter | brian_david_sweeney→flight_attendant→passenger →peter |
| clandestine_service:: counterterrorist | clandestine_service→covert_action→white_house→counterterrorist |
| elhassan::fadil_abdelgani | elhassan→explosive→cousin→fadil_abdelgani |
| general_shelton::roger_cressey | general_shelton→clark→counterterrorism_security_group →roger_cressey |
| gore::stephen_hadley | gore→white_house→national_security_adviser→stephen _hadley |
| karachi::usama_asmurai | karachi→yousef→manila→usama_asmurai |
| khalil_deek::turkey | khalil_deek→abu_hoshar→recruit→turkey |
| **L4 (Length 4)** | |
| abdullahi_farah::uae | abdullahi_farah→al-barakaat→ bin_ladin →jumale→uae |
| ahmad_taha::ayman_zawahiri | ahmad_taha→usama_bin_ladin→ egyptian_islamic_jihad→leader→ayman_zawahiri |

Table 6.2. Truth Chains (continued)

| **L4 (Length 4)** | |
|---|---|
| john_ross :: lorie_gottesman | john_ross →ins→fbi_document→suqami→ lorie_gottesman |

## 6.3. Evaluation of Semantic Path Chaining

The experiment processed all query pairs as illustrated in Table 6.2 and generated concept chains with different lengths ranging from 1 to 4 using the *SPC* approach. The objectives of the evaluation were to demonstrate how the incorporation of Wikipedia knowledge was able to help i) improve the ranking of topic-related concepts, and ii) identify topic-related concepts not appearing in the 9/11 document collection literally.

### 6.3.1. Parameter Tuning

As mentioned in Section3.4. , a combination of corpus-level TFIDF-based similarity, Wiki-article content-based similarity and category-based similarity was used to as the final weighting scheme to rank the concepts detected by the system. $\lambda_1$ and $\lambda_2$ are two tuning parameters that need to be tuned so that the similarity between concepts best match the judgements from our assessors. To accomplish this, we first built a set of training data composed of 10 query pairs randomly selected from the evaluation set, and then generated the intermediate profiles for each of them using our proposed *SPC* model. Among each of the intermediate profiles, we selected the top 5 concepts (links) within each semantic type, and compared their rankings with the assessors' judgements. The values of $\lambda_1$ and $\lambda_2$ were tuned in the range of [0.1, 1]. We set $\lambda_1 + \lambda_2 = 1$ or $\lambda_2 = 0$ to evaluate the contributions of each individual part, i.e., the

Wiki-article content-based similarity and the category-based similarity, respectively, in the final weighting scheme. In other words, the performance of only using Wikipedia articles as the outside knowledge could be observed by holding $\lambda_1 + \lambda_2 = 1$, and the performance of only using Wikipedia categories as the outside knowledge could be observed by setting $\lambda_2 = 0$. When only Wikipedia articles were considered, the best performance was achieved with $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$. When only Wikipedia categories were taken into account, the best performance was achieved with $\lambda_1 = 0.3$ and $\lambda_2 = 0$. When both articles and categories were utilized, the best performance was achieved with $\lambda_1 = 0.1$ and $\lambda_2 = 0.3$. These settings were used in our later experiments.

### 6.3.2. Experimental Results

Before proceeding to the evaluation of the *SPC* model, we first conducted an experiment to demonstrate the improved performance of our adapted ESA method against the original ESA.

We selected 10 concepts that we have knowledge about as shown in Table 6.3 and then built the interpretation vectors for each of them using the original ESA and our adapted ESA respectively. We calculated the averaged precision ratio defined as below to measure the performance of the two approaches.

$$aveP = \left(\sum_{i=1}^{N} \frac{concepts\ found\ and\ relevant}{total\ concepts\ found}\right) / N \qquad (6.1)$$

Where *N* is the number of concepts used for building the interpretation vectors. The results are illustrated in Figure 6.1, where the X-axis indicates the number of concepts kept in each of the interpretation vectors, while the Y-axis indicates the averaged precision ratio. It is obvious that

our adapted ESA achieves significant improvement for identifying topic-related Wikipedia concepts.

Table 6.3. 10 Concepts Used for the Interpretation Vector Construction

| Semantic Type | Belonging Concept |
|---|---|
| Person | George Bush |
| | Bill Clinton |
| Organization | Central Intelligence Agency |
| | United States Federal Government |
| Event | World War |
| | September 11 attacks |
| | Lewinsky Scandal |
| Science | Data Mining |
| | Natural Language Processing |
| | Artificial Intelligence |

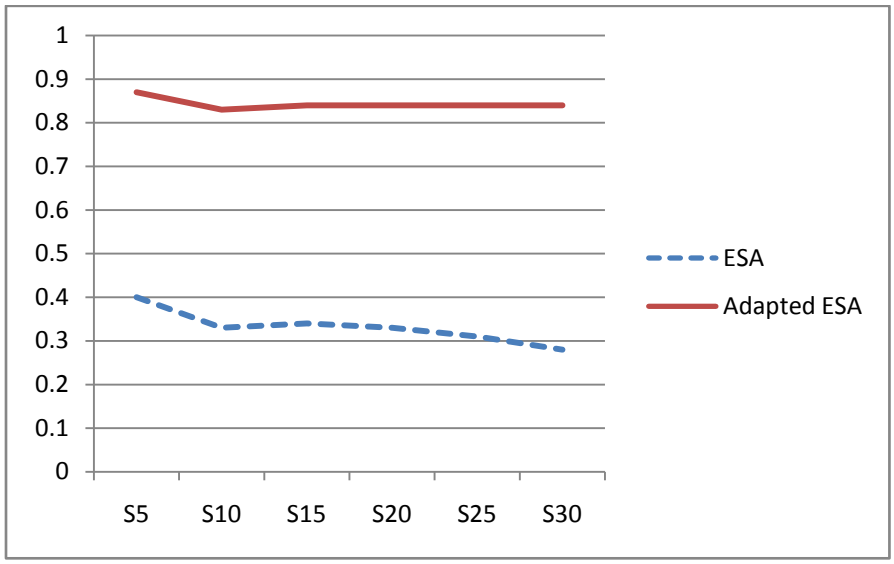


Figure 6.1. The Averaged Precision Ratio for the Generated Interpretation Vectors of the 10 Concepts in Table 6.3

Table 6.4 shows the top 15 concepts built in the interpretation vectors for 4 sample concepts. For example, for “Lewinsky Scandal”, the top 15 concepts in the interpretation vector built using our adapted ESA include most of the people involved in this event in addition to Clinton and Lewinsky themselves, such as Linda Tripp who secretly recorded Lewinsky's confidential phone calls about her relationship with Clinton, and Betty Currie who was the personal secretary of Clinton and well known in the scandal for handling gifts given to Lewinsky by Clinton. However, most of the top concepts identified using the original ESA were not that irrelevant.

Table 6.4. Top 15 Concepts in the Sample Interpretation Vectors Using the Adapted ESA and the Original ESA

| **Input** | **#** | **Original ESA** | **Adapted ESA** |
|---|---|---|---|
| Data Mining | 1 | Open-cast_mining | Relational_classification |
| | 2 | Opencast_Mining | Relational_data_mining |
| | 3 | Mining_engineer | Data_Mining_Extensions |
| | 4 | Open_cast_mining | Biological_data |
| | 5 | data | Java_Data_Mining |
| | 6 | Mine_(industry) | Weather_Data_Mining |
| | 7 | Open-cast_mine | National_Center_for_Data_Mining |
| | 8 | Golden_Source_of_data | Privacy_preserving_data_mining |
| | 9 | Data_withholding | Structure_mining |
| | 10 | Data_Havens | Oracle_Data_Mining |
| | 11 | Data_Warehousing | Cross_Industry_Standard_Process_for_ Data_Mining |
| | 12 | Data_Transfer | Knowledge_discovery |
| | 13 | Data_rate_(disambiguation) | Data_Pre-processing |

Table 6.4. Top 15 Concepts in the Sample Interpretation Vectors Using the Adapted ESA and the Original ESA (continued)

| Input | # | Original ESA | Adapted ESA |
|---|---|---|---|
| Data Mining | 14 | Data_General_One | Data_mining_agent |
| | 15 | Data_matrix_(disambiguation) | Sequence_mining |
| Central Intelligence Agency | 1 | Agency_(disambiguation) | United_States._Central_Intelligence_ Agency |
| | 2 | United_States._Central_Intelligence_Agency | Central_Intelligence_Agency_Museum |
| | 3 | Starfleet_Intelligence | Central_Intelligence_Agency_library |
| | 4 | Nigerian_intelligence | The_Agency |
| | 5 | Virginia_farmboys | National_Intelligence_Agency_(United_States) |
| | 6 | Directorate_for_Inter-Service_Intelligence | Agency |
| | 7 | Process_of_intelligence | Office_of_Scientific_Intelligence |
| | 8 | 14th_Intelligence_Company | Intelligence_officer |
| | 9 | Intelligence_augmentation | Security_agency |
| | 10 | Human_intelligence_ (disambiguation) | John_N._McMahon |
| | 11 | Israeli_Intelligence_Agency | National_Intelligence_Board |
| | 12 | Agência_Brasileira_de_ Inteligência | Director_of_the_Central_Intelligence_ Agency |
| | 13 | Central_(disambiguation) | Military_Intelligence_Division |
| | 14 | Administrative_agency | Private_intelligence_agency |
| | 15 | Job_agency | Intelligence_agency |
| Lewinsky Scandal | 1 | Scandal-mongering | Clinton:_His_Struggle_with_Dirt |
| | 2 | HIV-tainted-blood_scandal | Monica_Lewinsky |

Table 6.4. Top 15 Concepts in the Sample Interpretation Vectors Using the Adapted ESA and the Original ESA (continued)

| Input | # | Original ESA | Adapted ESA |
|---|---|---|---|
| Lewinsky Scandal | 3 | Scandal_of_Scientology | Lewinsky_scandal |
| | 4 | The_Scandal_of_Scientology_(book) | Linda_Tripp |
| | 5 | Iraq_War_Scandal_(disambiguation) | Susan_Schmidt |
| | 6 | CDU_contribution_scandal | Kramerbooks_&_Afterwords |
| | 7 | Parmalat_scandal | Betty_Currie |
| | 8 | Coingate | Monica |
| | 9 | Black_Mist_Scandal | Affair |
| | 10 | Scandal_(disambiguation) | Breuer |
| | 11 | 2006_Reuters_fake_photos_scandal | Charles_Ruff |
| | 12 | Boesky_scandal | Robert_S._Bennett |
| | 13 | Panama_scandal | Mark_Whitaker |
| | 14 | Sex_scandals | David_Horsey |
| | 15 | Shell_Scandal_of_1915 | 1983_congressional_page_sex_scandal |

Also, to evaluate the performance of the original ESA and the adapted ESA in semantic profile generation, we selected 10 query pairs as shown in Table 6.5 and then generated the semantic profiles for them. Each concept in the semantic profile was weighted using the original ESA and our adapted ESA respectively. We again calculated the averaged precision ratio to measure the percentage of the relevant concepts in the generated profile. The results are shown in Figure 6.2, where the X-axis indicates the number of concepts kept in each generated semantic

profile, while the Y-axis indicates the averaged precision ratio. It is demonstrated that for semantic profile generation, our adapted ESA also performs much better than the original ESA.

Table 6.5. 10 Query Pairs Used for the Semantic Profile Generation

| **Topic A** | **Topic C** |
|---|---|
| George Bush | Al Gore |
| Jennifer Aniston | Angelina Jolie |
| Sadam Hussein | Gulf War |
| Northern Alliance | European Union |
| Wall Street | New York Times |
| Steve Jobs | Mark Zuckerberg |
| Knowledge Discovery | Document Classification |
| Abdel Rahman | Blind Sheikh |
| Saudi Arabia | Kuwait |
| Terrorist Attack | Bill Clinton |

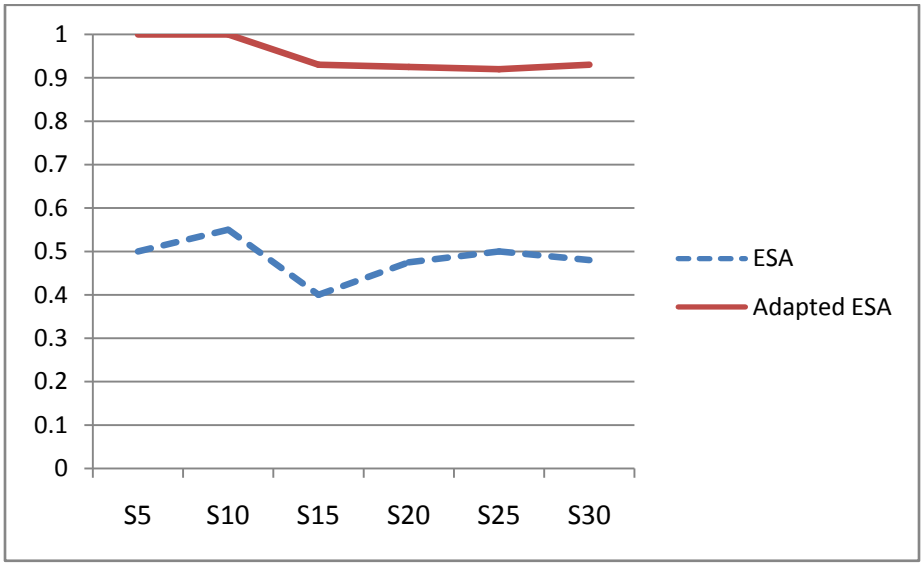


Figure 6.2. The Averaged Precision Ratio for the Intermediate Semantic Profiles Built for the 10 Query Pairs in Table 6.5

Table 6.6 shows the top 15 concepts generated in the semantic profiles for 2 sample query pairs: "*George Bush :: Al Gore*" and "*Jennifer Aniston :: Angelina Jolie*". For example, for the query pair "*Jennifer Aniston :: Angelina Jolie*", the top 10 concepts in the semantic profile generated using our adapted ESA include the most relevant persons, firms, events, etc. regarding "*Jennifer Aniston*" and "*Angelina Jolie*". Noise concepts were effectively removed from the semantic profile and key concepts were boosted to higher positions such as "*Screen International Security Services*", a security firm that ever provided security services to"*Jennifer Aniston*" and "*Angelina Jolie*". Also, another observation was that the original ESA failed to assign some of the very important concepts related to the two given topics higher ranks such as "*Brad Pitt*".

Table 6.6. Top 15 Concepts in the Sample Semantic Profiles Built Using the Adapted ESA and the Original ESA

| **Input** | **#** | **Original ESA** | **Adapted ESA** |
|---|---|---|---|
| George Bush :: Al Gore | 1 | George_Rose_(disambiguation) | Electoral_history_of_George_W._Bush |
| | 2 | Electoral_history_of_George_W._Bush | Al_Gore_presidential_campaign,_2000 |
| | 3 | Sir_Ralph_Gore,_4th_Baronet | Snippy |
| | 4 | St_George_Gore-St_George | Non-rigid_designator |
| | 5 | Sir_Arthur_Gore,_1st_Baronet | United_States_presidential_election_in_Massachusetts,_2000 |
| | 6 | Electoral_history_of_George_H._W._Bush | High_Performance_Computing_and_Communication_Act_of_1991 |
| | 7 | Al_Gore_presidential_campaign,_2000 | Millie_(dog) |
| | 8 | Tennis_at_the_1908_Summer_Olympics_–_Men's_indoor_doubles | United_States_presidential_election_in_the_District_of_Columbia,_2000 |
| | 9 | The_Betrayal_of_America | John_Prescott_Ellis |

Table 6.6. Top 15 Concepts in the Sample Semantic Profiles Built Using the Adapted ESA and the Original ESA (continued)

| Input | # | Original ESA | Adapted ESA |
|---|---|---|---|
| George Bush :: Al Gore | 10 | James_Howard_Gore | George_H._W._Bush |
| | 11 | The_Accidental_President | United_States_presidential_election_ in_Alaska,_2004 |
| | 12 | Snippy | Solid_South |
| | 13 | Donegal_County_(Parliament_ of_Ireland_constituency) | Florida_gubernatorial_election,_ 2002 |
| | 14 | 1901_Wimbledon_Championships _–_Gentlemen's_Singles | That's_My_Bush! |
| | 15 | Non-rigid_designator | United_States_presidential_election_ in_Nevada,_2000 |
| Jennifer Aniston :: Angelina Jolie | 1 | Brangelina | Brangelina |
| | 2 | Jennifer_Aniston | Jennifer_Aniston |
| | 3 | Look_Models_Management | Forbes_Celebrity_100 |
| | 4 | Forbes_Celebrity_100 | Angelina_Jolie |
| | 5 | Angelina_Jolie | Screen_International_Security_ Services |
| | 6 | Screen_International_Security_ Services | Fabian_Waintal |
| | 7 | Fabian_Waintal | Avi_Korein |
| | 8 | A_Midsummer_Night's_Rave | Big_Questions |
| | 9 | Avi_Korein | Brad_Pitt |
| | 10 | Mann_Village_Theatre,_ Westwood | Lana_Marks |
| | 11 | Jennifer_Santiago | 2002_Kids'_Choice_Awards |
| | 12 | The_Boy_Who_Grew_Flowers | The_Next_Best_Thing_(TV_series) |
| | 13 | Jolie_Jenkins | Where_My_Dogs_At? |
| | 14 | Big_Questions | Independent_Spirit_Award_for_Best _Female_Lead |
| | 15 | 2008_Ford_World_Women's_ Curling_Championship | Online_Film_Critics_Society_ Award_for_Best_Actress |

A quantitative evaluation has also been conducted where the following precision measure has been used.

$$precision = \frac{concept\ chains\ found\ and\ correct}{total\ concept\ chains\ found} \tag{6.2}$$

Table 6.7 through Table 6.9 make a comparison between the search results of our baseline where the corpus-level TFIDF-based statistical information is used to generate chains without the involvement of Wikipedia data. The table entries can be read as follows: $S_N$ / $W_N$ means the top $N$ concepts are kept in the search results where $S_N$ stands for the concepts appearing in the documents and $W_N$ stands for the concepts derived from Wikipedia. $L_N$ indicates the resulting chains of length $N$. The entries in the three tables stand for the precision values.

Specifically, Table 6.7 shows the improvement achieved by integrating the Wiki-article content-based measure over the baseline. For chains of length 1, it is demonstrated that adding the top 15 relevant Wikipedia articles achieves the best performance. For chains of lengths 2, 3 and 4, the best performance is obtained when adding the top 20 relevant articles. Table 6.8 presents the result when the relevant Wiki categories are used to improve the discovery model. For chains of length 1 through 4, the best performance is obtained when adding the top 20 relevant categories. Table 6.9 demonstrates the overall benefit when both the Wiki article contents and Wiki categories are incorporated, and the results show that adding the top 20 relevant articles and categories achieves the best performance. It is easy to observe that the search performance has been significantly improved with the integration of Wikipedia knowledge with the best performance achieved when both the Wiki article contents and categories are involved.

Table 6.7. The Effect of Integrating the Adapted ESA Technique (Original ESA+ Vector Cleaning)

| | | **Baseline/Wiki-ESA** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.8932 | 0.8778 | 0.8762 | 0.8711 | 0.8721 | 0.8719 |
| | W10 | 0.8993 | 0.8790 | 0.8787 | 0.8758 | 0.8745 | 0.8745 |
| | W15 | 0.9018 | 0.8819 | 0.8812 | 0.8787 | 0.8773 | 0.8778 |
| | W20 | 0.8977 | 0.8793 | 0.8770 | 0.8729 | 0.8751 | 0.8755 |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9152 | 0.9098 | 0.9005 | 0.8969 | 0.8936 | 0.8920 |
| | W10 | 0.9177 | 0.9121 | 0.9051 | 0.8983 | 0.8973 | 0.8997 |
| | W15 | 0.9198 | 0.9148 | 0.9083 | 0.9025 | 0.8995 | 0.9021 |
| | W20 | 0.9235 | 0.9177 | 0.9117 | 0.9052 | 0.9041 | 0.9048 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9150 | 0.9111 | 0.9008 | 0.9038 | 0.8939 | 0.8907 |
| | W10 | 0.9199 | 0.9140 | 0.9056 | 0.9059 | 0.8989 | 0.8932 |
| | W15 | 0.9235 | 0.9166 | 0.9107 | 0.9065 | 0.9001 | 0.8977 |
| | W20 | 0.9286 | 0.9189 | 0.9129 | 0.9124 | 0.9055 | 0.9029 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8449 | 0.8279 | 0.8125 | 0.8054 | 0.7933 | 0.7878 |
| | W10 | 0.8455 | 0.8288 | 0.8129 | 0.8039 | 0.7955 | 0.7892 |
| | W15 | 0.8470 | 0.8279 | 0.8121 | 0.8038 | 0.7960 | 0.7915 |
| | W20 | 0.8560 | 0.8285 | 0.8130 | 0.8049 | 0.7977 | 0.7944 |

Table 6.8. The Effect of Integrating Wikipedia Categories

| | | **Baseline/Wiki-CSV** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.8981 | 0.8813 | 0.8856 | 0.8815 | 0.8781 | 0.8785 |
| | W10 | 0.9155 | 0.8959 | 0.8992 | 0.9022 | 0.8989 | 0.8993 |
| | W15 | 0.9224 | 0.9052 | 0.9065 | 0.9134 | 0.9077 | 0.9075 |
| | W20 | 0.9288 | 0.9137 | 0.9177 | 0.9206 | 0.9162 | 0.9167 |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9177 | 0.9095 | 0.9017 | 0.8977 | 0.8944 | 0.8927 |
| | W10 | 0.9189 | 0.9144 | 0.9074 | 0.9005 | 0.8972 | 0.8986 |
| | W15 | 0.9207 | 0.9182 | 0.9098 | 0.9074 | 0.9013 | 0.9015 |
| | W20 | 0.9266 | 0.9237 | 0.9126 | 0.9106 | 0.9055 | 0.9069 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9182 | 0.9102 | 0.9018 | 0.8981 | 0.8933 | 0.8912 |
| | W10 | 0.9237 | 0.9155 | 0.9077 | 0.9022 | 0.8970 | 0.8955 |
| | W15 | 0.9265 | 0.9197 | 0.9096 | 0.9073 | 0.9005 | 0.8993 |
| | W20 | 0.9305 | 0.9249 | 0.9124 | 0.9098 | 0.9042 | 0.9028 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8470 | 0.8295 | 0.8144 | 0.8079 | 0.7984 | 0.7902 |
| | W10 | 0.8477 | 0.8297 | 0.8145 | 0.8076 | 0.7994 | 0.7899 |
| | W15 | 0.8520 | 0.8295 | 0.8137 | 0.8054 | 0.7973 | 0.7921 |
| | W20 | 0.8575 | 0.8309 | 0.8142 | 0.8059 | 0.8001 | 0.7969 |

Table 6.9. The Effect of Integrating both ESA and Wikipedia Categories

| | | **Baseline/Wiki-ESA-CSV** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.9105 | 0.8900 | 0.8871 | 0.8905 | 0.8825 | 0.8926 |
| | W10 | 0.9285 | 0.9077 | 0.9065 | 0.9077 | 0.9044 | 0.9120 |
| | W15 | 0.9371 | 0.9165 | 0.9157 | 0.9189 | 0.9128 | 0.9197 |
| | W20 | 0.9428 | 0.9225 | 0.9177 | 0.9290 | 0.9175 | 0.9247 |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9186 | 0.9105 | 0.9025 | 0.8988 | 0.8965 | 0.8968 |
| | W10 | 0.9206 | 0.9138 | 0.9098 | 0.9023 | 0.9017 | 0.9013 |
| | W15 | 0.9292 | 0.9237 | 0.9135 | 0.9107 | 0.9087 | 0.9047 |
| | W20 | 0.9298 | 0.9248 | 0.9128 | 0.9145 | 0.9102 | 0.9088 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9188 | 0.9135 | 0.9028 | 0.8989 | 0.8946 | 0.8938 |
| | W10 | 0.9249 | 0.9161 | 0.9075 | 0.9063 | 0.9017 | 0.8998 |
| | W15 | 0.9259 | 0.9174 | 0.9117 | 0.9115 | 0.9045 | 0.9030 |
| | W20 | 0.9321 | 0.9255 | 0.9150 | 0.9147 | 0.9092 | 0.9075 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8480 | 0.8312 | 0.8169 | 0.8108 | 0.7997 | 0.7935 |
| | W10 | 0.8490 | 0.8315 | 0.8182 | 0.8087 | 0.8005 | 0.7948 |
| | W15 | 0.8548 | 0.8321 | 0.8156 | 0.8071 | 0.8013 | 0.7950 |
| | W20 | 0.8578 | 0.8349 | 0.8145 | 0.8071 | 0.8043 | 0.8002 |

We then performed queries using the 37 query pairs in Table 6.2 to further evaluate the performance of the *SPC* model in detecting the 37 truth chains. The search results are illustrated in Table 6.10. We observed that the *SPC* model successfully discovered a majority of the truth chains ranging from length 1 to length 4.

Table 6.10. Search Results of the Truth Chains

| **Truth Chains** | **Found** |
|---|---|
| abdel_rahman → blind_sheikh | √ |
| abdullahi_farah→ jumale | × |
| adel→ ffi | √ |
| alexis→ lloyd_salvetti | √ |
| american_muslim→ khifa | √ |
| atta→ dekkers | √ |
| crawford→ khalilzad | √ |
| donovan→ wall_street | √ |
| easton_police_department→ lee_hanson | √ |
| glenn→ pressler | √ |
| global_positioning_system→ jarrah | √ |
| kenya→ mohamed | √ |
| martha_stewart→ saudi_arabia | √ |
| saudi_arabian_ministry→ thumairy | √ |
| abdullah →kingdom →world_trade_organization | √ |
| abdullahi_farah→jumale →uae | × |
| ajaj→unite_state→ali | × |
| amal → cia → sudanese | √ |

Table 6.10. Search Results of the Truth Chains (continued)

| Truth Chains | Found |
|---|---|
| ayman_zawahiri→ bin_ladin →turabi →national_islamic_front | × |
| ayman_zawahiri→ bin_ladin→taliban→qaeda_presence | × |
| binalshibh →fund → fbi →pistole | √ |
| brian_david_sweeney→flight_attendant→passenger →peter | √ |
| clandestine_service→covert_action→white_house→counterterrorist | √ |
| elhassan→explosive→cousin→fadil_abdelgani | × |
| general_shelton→clark→counterterrorism_security_group→roger_cressey | √ |
| gore→white_house→national_security_adviser→stephen_hadley | √ |
| karachi→yousef→manila→usama_asmurai | × |
| khalil_deek→abu_hoshar→recruit→turkey | √ |
| abdullahi_farah→al-barakaat→ bin_ladin →jumale→uae | × |
| ahmad_taha→usama_bin_ladin→ egyptian_islamic_jihad→leader→ayman_zawahiri | √ |
| john_ross →ins→fbi_document→suqami→ lorie_gottesman | √ |

Through incorporating the world knowledge into the cross-document knowledge discovery process, we are able to obtain a significant amount of knowledge not present in the given documents. For concept relationship discovery, the proposed approach is capable of discovering topic-related concepts not appearing in the documents literally. Table 6.11 shows the newly discovered semantic relationships where the linking concepts can only be acquired through integrating information from multiple documents or from Wikipedia knowledge. For instance, the discovered relationship chain "*Atta →Marwan_al-Shehhi → Huffman_Aviation → dekkers,*" can be interpreted as the following: "*Marwan_al-Shehhi*" was the hijacker-pilot of

United Airlines Flight 175, crashing the plane into the South Tower of the World Trade Center as part of the September 11 attacks. It was revealed after the September 11th attacks that "*Atta*"and"*Marwan_al-Shehhi*" had both attended the school named "*Huffman_Aviation*" to learn how to fly small aircraft. Also, the "*Huffman_Aviation*" flight-training school was purchased by Dutchman Rudi Dekkers in 1999.

Table 6.11. Instances of Enriched Semantic Relationships

| **Query Pair** | **Resulting Chain** |
|---|---|
| **L2 (Length 2)** | |
| abdel_rahman :: blind_sheikh | abdel_rahman →omar_abdel-rahman → blind_sheikh |
| atta :: dekkers | atta→planning_of_the_september_11_attacks →dekkers |
| ahmad_taha :: ayman_zawahiri | ahmad_taha →osama_bin_laden→ ayman_zawahiri |
| marty_miller :: oakley | marty_miller →unocal_corporation → oakley |
| gore :: stephen_hadley | gore →the_vulcans→ stephen_hadley |
| alexis :: lloyd_salvetti | alexis →michael_hurley → lloyd_salvetti |
| ajaj:: ali | ajaj →ramzi_yousef → ali |
| martha_stewart :: saudi_arabia | martha_stewart →steve_cohen_(magician) → saudi_arabia |
| **L3 (Length 3)** | |
| atta :: dekkers | atta →marwan_al-shehhi →huffman_aviation→ dekkers |
| amal :: sudanese | amal →islamic_jihad_organization→ cia → sudanese |
| karachi :: usama_asmurai | karachi →hamid_mir → bin_ladin → usama_asmurai |
| binalshibh :: pistole | binalshibh →ziad_jarrah →fbi→ pistole |
| ayman_zawahiri :: national_islamic_front | ayman_zawahiri→al-qaeda→ bin_ladin →national_islamic_front |

Table 6.11. Instances of Enriched Semantic Relationships (continued)

| Query Pair | Resulting Chain |
|---|---|
| **L4 (Length 4)** | |
| kenya :: mohamed | kenya →kai_hirschmann → afghanistan →mohamed_omer → mohamed |
| gore :: stephen_hadley | gore →hal_bidlack→national_security →national_security_advisor_(united states) → stephen_hadley |
| crawford :: khalilzad | crawford→george_w._bush's_second_term_as_president_of_the_united_states→ afghan →zalmay_khalilzad → khalilzad |
| atta :: dekkers | atta →marwan_al-shehhi →huffman_aviation →planning_of_the_september_11_attacks→ dekkers |
| ahmad_taha :: ayman_zawahiri | ahmad_taha→osama_bin_laden→bin_ladin →september_11_attacks→ayman_zawahiri |

### 6.3.3. Summary

To summarize, the *SPC* mining model successfully answers a majority of the tested queries. By taking advantage of knowledge derived from Wikipedia, this model is able to provide a much more comprehensive knowledge repository to support various queries and effectively complements existing knowledge contained in text corpus. By applying a sequence of heuristic strategies to clean the Wikipedia concept vector which we observe contains a fair amount of noise and is not precise enough to represent the contextual clues related to topics of interest, this model achieves better search results in terms of the precision ratio.

## 6.4. Experimental Results for Kernel Methods

The objectives of this section are to evaluate how the various semantic kernels proposed in this dissertation perform in capturing the semantic relationships between concepts. We

interpreted the search results using the precision ratio and the mean average precision measure to demonstrate the effectiveness of the kernel methods.

### 6.4.1. Experimental Results

Table 6.12 through Table 6.14 summarize the results we obtain on executing queries from the evaluation set using the precision defined in Section 0. In particular, Table 6.12 shows the improvement achieved by integrating the Wiki-article content-based kernel over the baseline; Table 6.13 presents the results when the relevant Wiki categories are used to build the semantic kernel; Table 6.14 demonstrates the overall benefit when utilizing the hybrid semantic kernel where both the article content and categories are incorporated. The observations are consistent with the findings in Table 6.7 through Table 6.9, and the best performance is observed by applying the hybrid semantic kernel.

Table 6.12. The Effect of Using the Article-Content-based Kernel

| | | **Baseline/Article-Content-based Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.9048 | 0.8861 | 0.8842 | 0.8808 | 0.8798 | 0.8798 |
| | W10 | 0.9074 | 0.8889 | 0.8870 | 0.8836 | 0.8826 | 0.8826 |
| | W15 | 0.9086 | 0.8902 | 0.8884 | 0.8850 | 0.8840 | 0.8840 |
| | W20 | 0.9067 | 0.8886 | 0.8868 | 0.8834 | 0.8825 | 0.8825 |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9155 | 0.9106 | 0.9007 | 0.8974 | 0.8945 | 0.8928 |
| | W10 | 0.9226 | 0.9139 | 0.9075 | 0.9041 | 0.9011 | 0.8995 |

Table 6.12. The Effect of Using the Article-Content-based Kernel (continued)

| | | **Baseline/Article-Content-based Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_2$** | W15 | 0.9272 | 0.9184 | 0.9120 | 0.9087 | 0.9057 | 0.9040 |
| | W20 | 0.9306 | 0.9217 | 0.9154 | 0.9120 | 0.9090 | 0.9074 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9157 | 0.9109 | 0.9009 | 0.8976 | 0.8946 | 0.8930 |
| | W10 | 0.9228 | 0.9142 | 0.9077 | 0.9044 | 0.9014 | 0.8997 |
| | W15 | 0.9275 | 0.9187 | 0.9123 | 0.9090 | 0.9059 | 0.9043 |
| | W20 | 0.9309 | 0.9220 | 0.9157 | 0.9124 | 0.9093 | 0.9077 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8456 | 0.8271 | 0.8119 | 0.8056 | 0.7932 | 0.7901 |
| | W10 | 0.8473 | 0.8279 | 0.8119 | 0.8039 | 0.7941 | 0.7898 |
| | W15 | 0.8479 | 0.8290 | 0.8127 | 0.8041 | 0.7967 | 0.7933 |
| | W20 | 0.8562 | 0.8295 | 0.8135 | 0.8055 | 0.7985 | 0.7965 |

Table 6.13. The Effect of Using the Category-based Kernel

| | | **Baseline/Category-based Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.9202 | 0.9030 | 0.9011 | 0.8979 | 0.8969 | 0.8969 |
| | W10 | 0.9347 | 0.9195 | 0.9175 | 0.9148 | 0.9138 | 0.9138 |
| | W15 | 0.9437 | 0.9299 | 0.9280 | 0.9255 | 0.9246 | 0.9246 |
| | W20 | 0.9497 | 0.9370 | 0.9352 | 0.9329 | 0.9320 | 0.9320 |

Table 6.13. The Effect of Using the Category-based Kernel (continued)

| | | **Baseline/Category-based Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9185 | 0.9103 | 0.9042 | 0.9012 | 0.8987 | 0.8974 |
| | W10 | 0.9252 | 0.9168 | 0.9106 | 0.9076 | 0.9050 | 0.9037 |
| | W15 | 0.9297 | 0.9211 | 0.9150 | 0.9120 | 0.9093 | 0.9080 |
| | W20 | 0.9329 | 0.9282 | 0.9183 | 0.9152 | 0.9126 | 0.9113 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9185 | 0.9102 | 0.9037 | 0.9005 | 0.8976 | 0.8960 |
| | W10 | 0.9253 | 0.9167 | 0.9103 | 0.9071 | 0.9041 | 0.9025 |
| | W15 | 0.9298 | 0.9211 | 0.9148 | 0.9116 | 0.9086 | 0.9070 |
| | W20 | 0.9331 | 0.9283 | 0.9181 | 0.9149 | 0.9120 | 0.9104 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8469 | 0.8288 | 0.8139 | 0.8076 | 0.7961 | 0.7929 |
| | W10 | 0.8498 | 0.8297 | 0.8135 | 0.8057 | 0.7959 | 0.7905 |
| | W15 | 0.8532 | 0.8301 | 0.8141 | 0.8056 | 0.7988 | 0.7941 |
| | W20 | 0.8583 | 0.8323 | 0.8153 | 0.8084 | 0.8015 | 0.7995 |

Table 6.14. The Effect of Using the Hybrid Kernel

| | | **Baseline/Hybrid Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | Baseline | 0.8844 | 0.8689 | 0.8700 | 0.8668 | 0.8597 | 0.8546 |
| | W5 | 0.9267 | 0.9104 | 0.9084 | 0.9054 | 0.9044 | 0.9044 |

Table 6.14. The Effect of Using the Hybrid Kernel (continued)

| | | **Baseline/Hybrid Kernel** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **$S_5$** | **$S_{10}$** | **$S_{15}$** | **$S_{20}$** | **$S_{30}$** | **$S_{40}$** |
| **$L_1$** | W10 | 0.9402 | 0.9259 | 0.9239 | 0.9213 | 0.9204 | 0.9204 |
| | W15 | 0.9482 | 0.9353 | 0.9334 | 0.9311 | 0.9302 | 0.9302 |
| | W20 | 0.9522 | 0.9403 | 0.9385 | 0.9364 | 0.9355 | 0.9355 |
| **$L_2$** | Baseline | 0.9174 | 0.9081 | 0.8998 | 0.8959 | 0.8917 | 0.8888 |
| | W5 | 0.9168 | 0.9121 | 0.9057 | 0.9033 | 0.8994 | 0.8980 |
| | W10 | 0.9263 | 0.9173 | 0.9144 | 0.9090 | 0.9058 | 0.9041 |
| | W15 | 0.9332 | 0.9285 | 0.9194 | 0.9162 | 0.9101 | 0.9092 |
| | W20 | 0.9334 | 0.9295 | 0.9190 | 0.9173 | 0.9147 | 0.9139 |
| **$L_3$** | Baseline | 0.9180 | 0.9109 | 0.9003 | 0.8964 | 0.8922 | 0.8893 |
| | W5 | 0.9199 | 0.9168 | 0.9055 | 0.9013 | 0.8988 | 0.8972 |
| | W10 | 0.9273 | 0.9188 | 0.9130 | 0.9097 | 0.9061 | 0.9037 |
| | W15 | 0.9298 | 0.9233 | 0.9167 | 0.9144 | 0.9093 | 0.9085 |
| | W20 | 0.9354 | 0.9297 | 0.9187 | 0.9162 | 0.9134 | 0.9122 |
| **$L_4$** | Baseline | 0.8444 | 0.8265 | 0.8109 | 0.8027 | 0.7919 | 0.7865 |
| | W5 | 0.8498 | 0.8294 | 0.8176 | 0.8105 | 0.7979 | 0.7964 |
| | W10 | 0.8518 | 0.8320 | 0.8166 | 0.8084 | 0.7973 | 0.7927 |
| | W15 | 0.8544 | 0.8328 | 0.8156 | 0.8069 | 0.8022 | 0.7988 |
| | W20 | 0.8599 | 0.8374 | 0.8167 | 0.8098 | 0.8050 | 0.8017 |

We also used the adapted *MAP* measure as shown below for our evaluation:

$$Adapted - MAP(Q) = (\sum P(k_{s,w}))/|Q| \quad (6.3)$$

Where $Q$ is a set containing all query pairs, $s$ = {5, 10, 15, 20, 30, 40} and $w$ = {5, 10, 15, 20} indicate the top $N$ concepts kept in the search results ($s$ stands for the concepts appearing in the document collection and $w$ stands for the concepts derived from Wikipedia). $P(k_{s,w})$ is the precision where the top $s$ concepts from documents and the top $w$ concepts from Wikipedia were used.

Figure 6.3 through Figure 6.6 interpret the search results using the *MAP* measure. *SN* where *N={5,10,15,20,30,40}* and *WN* where *N={5,10,15,20}* represent the same as in Table 6.12 through Table 6.14. The baseline is the Vector Space Model applied only in the document collection . For *WN-X* where *X={A,C,H}*, *A* indicates the article content-based kernel, *C* indicates the category-based kernel and *H* indicates the hybrid kernel. We observe that the kernel-based approach consistently achieves better performance for different lengths of chains than the baseline solution, and the hybrid kernel achieves the highest MAP values for chains of different lengths.

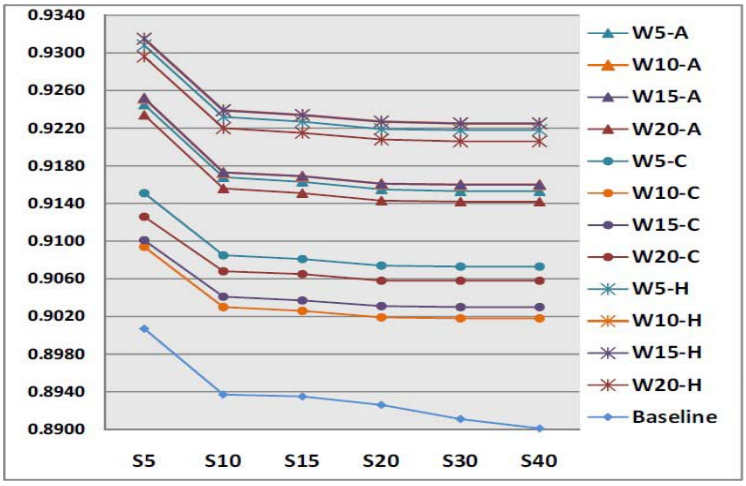


Figure 6.3. Adapted MAP for Chains of Length 1

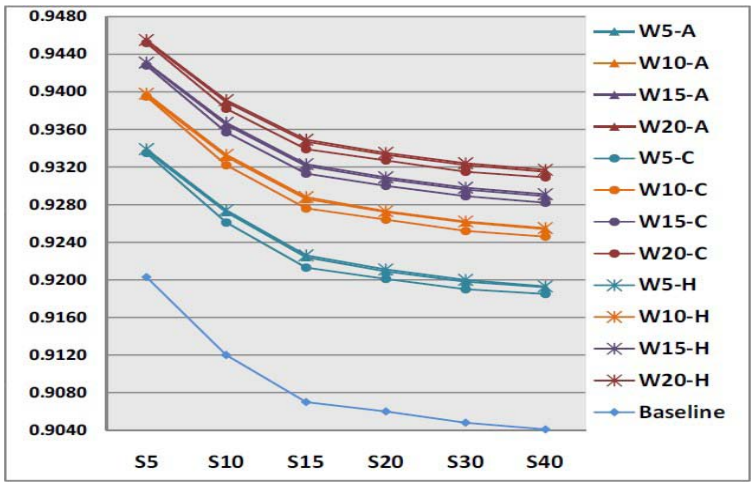


Figure 6.4. Adapted MAP for Chains of Length 2

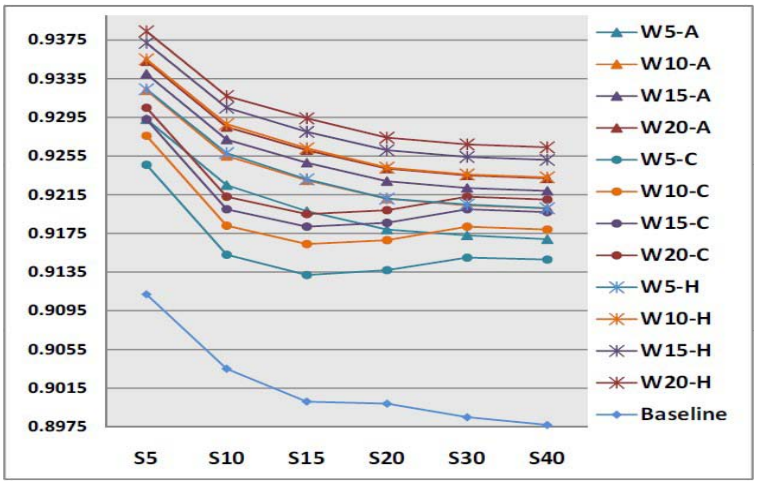


Figure 6.5. Adapted MAP for Chains of Length 3

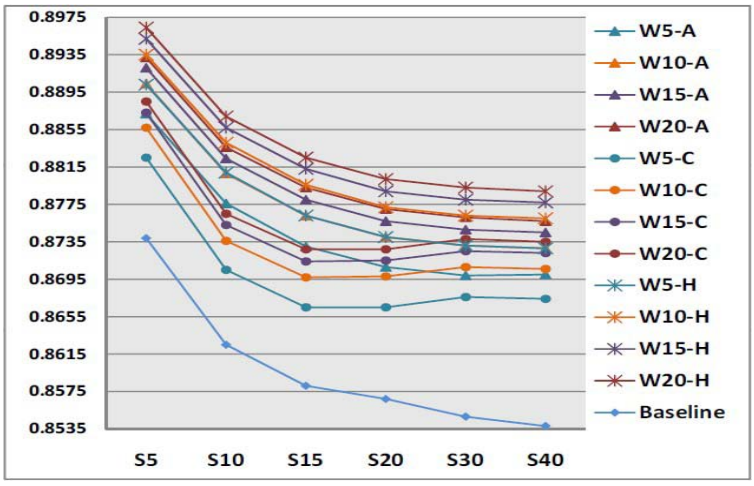


Figure 6.6. Adapted MAP for Chains of Length 4

### 6.4.2. Summary

Compared with the *SPC* mining model described above and other approaches that incorporate background knowledge, the semantic kernel-based mining model is capable of i) avoiding human intervention in weighing the contributions of different information resources in semantic relatedness calculation, which is often difficult to achieve in reality, and ii) minimizing the residual of computing semantic relatedness between concepts.

## 6.5. Experimental Results for Concept Association Graph

We also adopted various strategies to evaluate the performance of the profile-based approach and the *CAG*-based approach in a graphical context. The effectiveness of these two approaches was also demonstrated in comparison to a competitive baseline model: the RelFinder system which is a DBpedia-based application for exploring relationships between given objects in RDF data.

### 6.5.1. Experimental Results

Figure 6.7 and Figure 6.8 illustrate the generated concept associations at various lengths. We observed that among the discovered associations, a signification amount of them are unapparent hidden associations that were not mentioned in the 9/11 commission report at all. This is extremely important for counterterrorism hypothesis generation, since finding common knowledge (e.g. *Bill Clinton* is the predecessor of *George Bush*) does not have much meaning in helping identify potential terrorists and prevent possible attacks like the 9/11 attack.

<**Paths** source="george_bush" destination="bin_ladin">
<**Paths length=2**>
<Path id="2" value="george_bush → planning_of_the_september_11_attacks → bin_ladin">
<Path id="3" value="george_bush → porter_goss → bin_ladin">
<**Paths length=3**>
<Path id="4" value="george_bush → john_e._mcLaughlin → porter_goss → bin_ladin">
<Path id="5" value="george_bush → george_tenet → national_security → bin_ladin">
<Path id="6" value="george_bush → porter_goss → planning_of_the_september_11_attacks → bin_ladin">
<**Paths length=4**>
<Path id="7" value="george_bush → michael_hayden_(general) → porter_goss → joint_inquiry_into_intelligence_community_activities_before_and_after_the_terrorist_attacks_of_september_11,_2001 → bin_ladin">
<Path id="8" value="george_bush → central_security_service → u.s._intelligence_agencies → joint_inquiry_into_intelligence_community_activities_before_and_after_the_terrorist_attacks_of_september_11,_2001 → bin_ladin">
<Path id="9" value="george_bush → director_of_the_central_intelligence_agency → 9/11_commission → war_on_terrorism → bin_ladin">

Figure 6.7. Association Chains Connecting "*George Bush*" to "*Bin Ladin*" Generated Using the Profile-based Approach

For example, the discovered association chain in Figure 6.8: "*george_bush → william_h.t._bush → porter_goss → united_states_intelligence_community → national_security → terrorism →bin_ladin_determined_to_strike_in_us → bin_ladin*" implies the following meaning: "*Porter Goss*", who was the first Director of the Central Intelligence Agency (DCIA) and the last Director of Central Intelligence, was a member of the Psi Upsilon fraternity alongside "*William H.T. Bush*", the uncle of President *George W. Bush*. He has a close tie to the "*United States Intelligence Community*", which contributes to preserve the "*National Security*". "*Bin Ladin Determined To Strike in US*" is a "*Terrorism*" attack that was planned by "*Bin Ladin*"

and finally came true on Tuesday, September 11, 2001. Figure 6.7 shows the resulting concept associations linking from "*George Bush*" to "*Bin Ladin*" using the profile-based approach. Figure 6.8 gives the associations from "*George Bush*" to "*Bin Ladin*" using the *CAG*-based approach.

<**Paths** source="george_bush" destination="bin_ladin">
<**Paths length=5**>
<Path id="1" value="george_bush → george_tenet → national_security_council → central_intelligence_agency → september_11,_2001_attacks → bin_ladin">
<Path id="2" value="george_bush → porter_goss → mahmud_ahmed → abdul_salam_zaeef → joint_inquiry_into_intelligence_community_activities_before_and_after_the_terrorist_attacks_of_september_11,_2001 → bin_ladin">
<**Paths length=6**>
<Path id="3" value="george_bush → united_states_national_security_council → richard_a._clarke → stephen_hadley → al-qaeda → taliban → bin_ladin">
<**Paths length=7**>
<Path id="4" value="george_bush → william_h.t._bush → porter_goss → united_states_intelligence_community → national_security → terrorism → bin_ladin_determined_to_strike_in_us → bin_ladin">
<Path id="5" value="george_bush → federal_government → richard_a._clarke → 9/11_commission → saddam_hussein → terrorism → bin_ladin_determined_to_strike_in_us → bin_ladin">
<**Paths length=8**>
<Path id="6" value="george_bush → bill_clinton → george_tenet → central_intelligence_agency → inter_services_intelligence → terrorism → taliban → osama_bin_laden → bin_ladin">

Figure 6.8. Association Chains Connecting "*George Bush*" to "*Bin Ladin*" Generated Using the *CAG*-based Approach

The next experiment was designed to evaluate the performance of the *CAG*-based approach in answering domain-specific queries. We used the 37 counterterrorism query pairs shown in Table 6.2 as our search topics. Table 6.15 shows whether the queries can be answered or not using RelFinder and the *CAG*-based approach, respectively. Since RelFinder always first

transforms an input topic into a suggested topic that matches its backend RDF database, in Table 6.15, the "*Suggested Source/Destination Topic*" columns indicate the transformed topics by RelFinder based on the input topics. The results in Table 6.15 demonstrate that RelFinder falls far behind the *CAG*-based approach in response to these domain specific queries.

Table 6.15. Query Results in Counterterrorism Domain

| **Query Pair** | **RelFinder** | | | ***CAG*** |
|---|---|---|---|---|
| | **Suggested Source Topic** | **Suggested Destination Topic** | **Associations Found?** | **Associations Found?** |
| abdel rahman-blind sheikh | Omar Abdel-Rahman | Blind Sheikh | √ | √ |
| abdullahi farah-jumale | × | × | × | × |
| abdullahi farah-uae | × | × | × | × |
| abdullah-world trade organization | Abdullah II of Jordan | World Trade Organization | √ | × |
| adel-ffi | Saif al-Adel | FFI | × | × |
| ahmad taha-ayman zawahiri | Abu-Yasir Rifa'i Ahmad Taha | Ayman Zawahiri | × | √ |
| ajaj-ali | Ahmed Ajaj | ali | × | √ |
| alexis-lloyd salvetti | × | × | × | × |
| amal-sudanese | Amal | × | × | √ |
| american muslim-khifa | American Muslim | × | × | × |
| atta-dekkers | Mohamed Atta | × | × | √ |
| atta-huffman | Mohamed Atta | × | × | √ |
| ayman zawahiri-national islamic front | Ayman Zawahiri | National Islamic Front | × | × |
| ayman zawahiri-qaeda presence | Ayman Zawahiri | × | × | √ |

Table 6.15. Query Results in Counterterrorism Domain (continued)

| Query Pair | RelFinder | | | *CAG* |
|---|---|---|---|---|
| | **Suggested Source Topic** | **Suggested Destination Topic** | **Associations Found?** | **Associations Found?** |
| betty ong-madeline | Betty Ong | Madeline | √ | √ |
| binalshibh-pistole | × | pistole | × | × |
| brian david sweeney-peter | Brian David Sweeney | × | × | × |
| christopher steele-perez | × | × | × | × |
| clandestine service-counterterrorist | Clandestine service | Counterterro-rist | × | √ |
| counterterrorist center-khalid shaykh ballushi | Counterterrorist Intelligence Center | × | × | × |
| crawford-khalilzad | × | Zalmay Khalilzad | × | √ |
| dekkers-jones aviation | × | × | × | × |
| donovan-wall street | Donovan | Wall Street | × | √ |
| easton police department-lee hanson | × | × | × | × |
| elhassan-fadil abdelgani | × | × | × | √ |
| general shelton-roger cressey | × | Roger Cressey | × | × |
| george bush-bin ladin | George W. Bush | Bin Ladin | × | √ |
| glenn-pressler | Glenn A. Fine | Larry Pressler | √ | √ |
| global positioning system-jarrah | × | Ziad Jarrah | × | × |
| gore-stephen hadley | Al Gore | | √ | × |
| john ross-lorie gottesman | John Ross | × | × | √ |

Table 6.15. Query Results in Counterterrorism Domain (continued)

| Query Pair | RelFinder | | | *CAG* |
|---|---|---|---|---|
| | **Suggested Source Topic** | **Suggested Destination Topic** | **Associations Found?** | **Associations Found?** |
| karachi-usama asmurai | Karachi | × | × | × |
| kenya-mohamed | Kenya | Mohamed Atta | × | √ |
| khalil deek-turkey | Khalil Deek | | × | × |
| martha stewart-saudi arabia | Martha Stewart | Saudi Arabia | × | √ |
| marty miller-oakley | × | Robert B. Oakley | × | √ |
| oakley-unocal | Robert B. Oakley | Unocal | × | √ |
| saudi arabian ministry-thumairy | Saudi Arabian Ministry of Oil | × | × | × |

In particular, RelFinder was designed to find ontological relationships between concepts. However, the ability of finding non-ontological linking terms between concepts of interest also plays an important role in this process. For example, given a query pair "*Albert Einstein::Kurt Godel*", one of the discovered non-ontological concepts in the path connecting them, "*Princeton, New Jersey*", is the "*deathPlace*" of "*Albert Einstein*" where "*deathPlace*" represents the ontological relationship between "*Albert Einstein*" and "*Princeton, New Jersey*". Similarly, other non-ontological linking terms such as "*Hans Hahn*", "*David Hume*" and "*Gottfried Wilhelm Leibniz*", also need to be found when the potential connection between them is a multi-stage path. Therefore, although applied in DBpedia and served for ontological relationship discovery, both RelFinder and our system have the same rationale that involves the identification of potential non-ontological concepts connecting two given topics. To further evaluate this, we have used the

37 truth chains shown in Table 6.2 to measure how many of them can be generated by RelFinder and the *CAG*-based approach, respectively. As shown in Figure 6.9, for example, suppose $A \rightarrow B \rightarrow C$ is a truth chain representing the relationship between topics *A* and *C*, we examine whether concept *B* can be found as a relevant concept to link *A* and *C* by RelFinder and our *CAG*-based approach, respectively. The results demonstrate that the *CAG*-based approach performs much better in finding these non-ontological concepts than RelFinder.

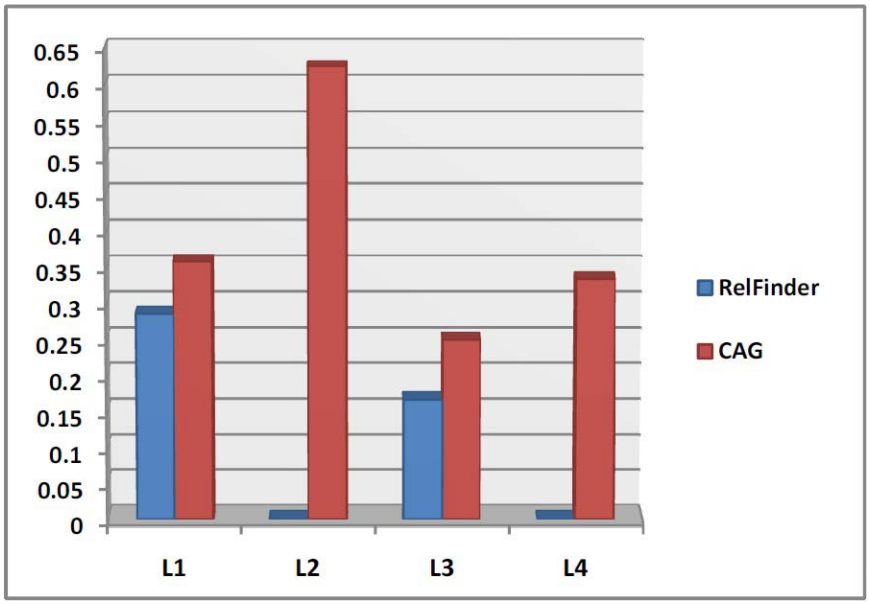


Figure 6.9. The Discovered Truth Chain Percentage in the Counterterrorism Domain Using RelFinder and *CAG*

Moreover, we have used the four representative query pairs designed for RDF database queries used by Relfinder to evaluate the performance of our system in answering these same queries. We found that i) the discovered relationships using RelFinder were far too limited; ii) not only did our approach successfully answer the given queries, but also introduced a fair amount of unapparent hidden relationships that RelFinder failed to uncover. As shown in Table 6.16, for example, the relationship chain "*albert_einstein →phillip_forman →u.s._citizenship →kurt_ gödel*" has two very unapparent relationships between "*Albert Einstein*" and "*Kurt*

*Gödel*": "*Phillip Forman*" knew *Einstein* and had administered the oath at *Einstein*'s own citizenship hearing. He was also the judge while "*Kurt Gödel*" was in his "*U.S. Citizenship*" exam.

Table 6.16. Search Results with the Representative Query Pairs Using the Graph-based Approach

| **Query Pair** | **Unapparent Relationship Chains** | **Goodness** |
|---|---|---|
| albert_einstein :: kurt_ gödel | albert_einstein →albert_einstein_award → kurt_gödel | 0.4279 |
| | albert_einstein →albert_einstein_award → julian_schwinger → kurt_gödel | 0.2855 |
| | albert_einstein →einstein_field_equations → gödel_metric →kurt_gödel | 0.0619 |
| | albert_einstein →phillip_forman →u.s._citizenship →kurt_gödel | 0.0032 |
| | albert_einstein →god_was_impersonal → kurt_gödel | 0.0009 |
| albert_einstein :: stuttgart | albert_einstein →einstein_on_the_beach →stuttgart | 0.0990 |
| | albert_einstein →akhnaten_(opera) →orchestra_pit →stuttgart | 0.0141 |
| | albert_einstein →arnold_sommerfeld →stuttgart | 0.0063 |
| | albert_einstein → theory_of_relativity →hans_reichenbach →engineering →stuttgart | 0.0040 |
| | albert_einstein →hans_reichenbach →stuttgart | 0.0035 |
| | albert_einstein →heinrich_sontheim →stuttgart | 0.0031 |
| | albert_einstein →columbia_university →herbert_eulenberg →stuttgart | 0.0019 |
| | albert_einstein → hendrik_lorentz → common_sense → max_abraham →stuttgart | 0.0007 |

Table 6.16. Search Results with the Representative Query Pairs Using the Graph-based Approach (continued)

| Query Pair | Unapparent Relationship Chains | Goodness |
|---|---|---|
| leipzig :: berlin | leipzig →friedrich_heinrich_von_der_hagen → berlin | 0.0297 |
| | leipzig →espenhain → berlin | 0.0292 |
| | leipzig →joachim_wilhelm_franz_philipp_von_holtzendorff → berlin | 0.0175 |
| | leipzig →friedrich_engel_(mathematician) → berlin | 0.0143 |
| | leipzig →gustav_hirschfeld → berlin | 0.0059 |
| duisburg :: essen | duisburg →universität_duisburg-essen → essen | 0.2040 |
| | duisburg →second_world_war → united_states_army_air_forces → bombing_of_essen_in_world_war_II→ essen | 0.0645 |
| | duisburg → university_of_duisburg → kees_schouhamer_immink → essen | 0.0614 |
| | duisburg →germany → essen | 0.0401 |
| | duisburg →european_route_of_industrial_heritage → essen | 0.0008 |

Figure 6.10 shows the system interface and the mining results for "*Bill Clinton*" and "*Bin Ladin*" as the search topics, and Figure 6.11 gives the mining results given "*Gore*" and "*Stephen Hadley*" as the search topics.

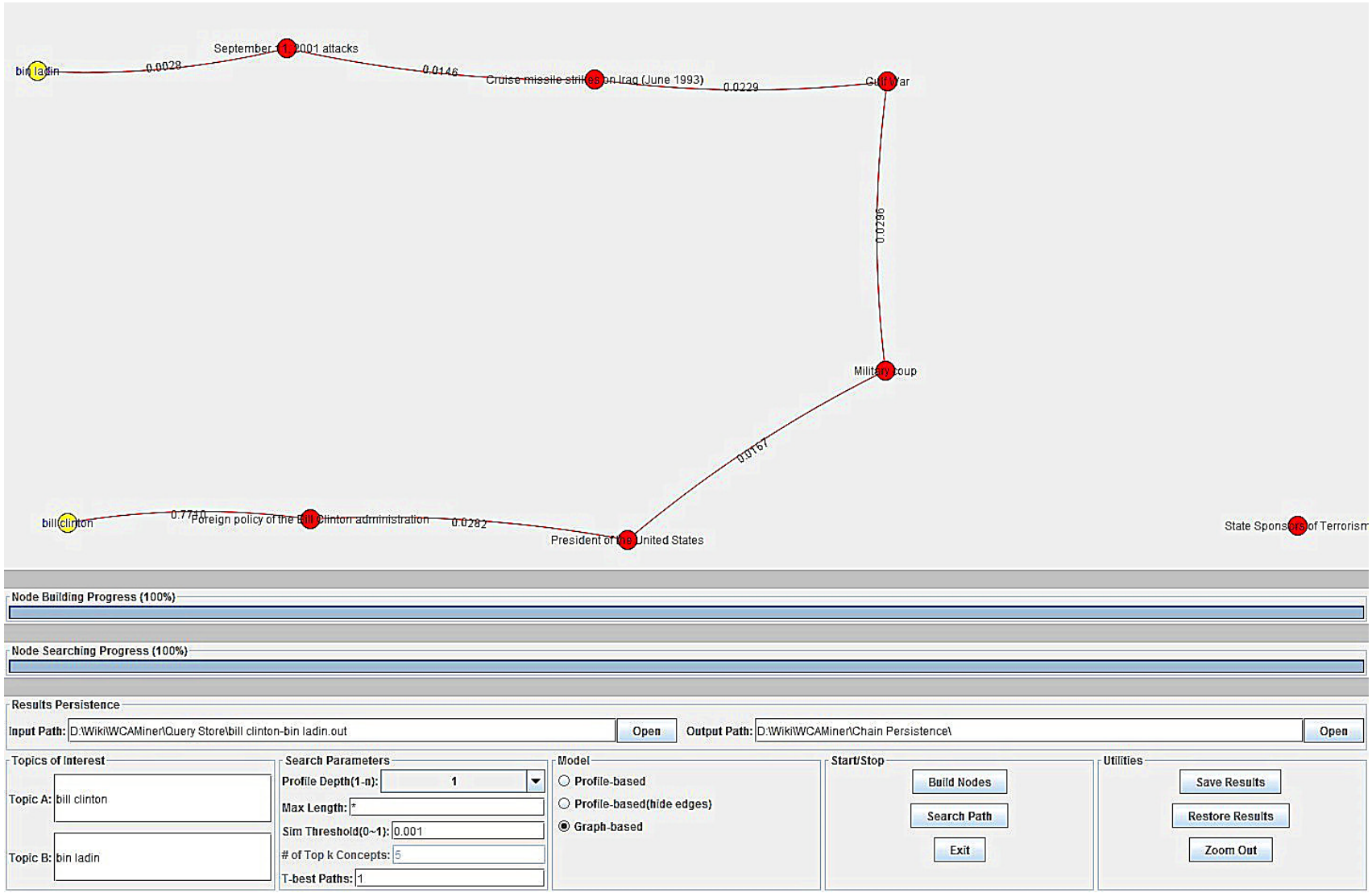


Figure 6.10. The Best Relationship Chain Discovered for the Query Pair "*Bill Clinton :: Bin Ladin*" Using the *CAG*-based Approach

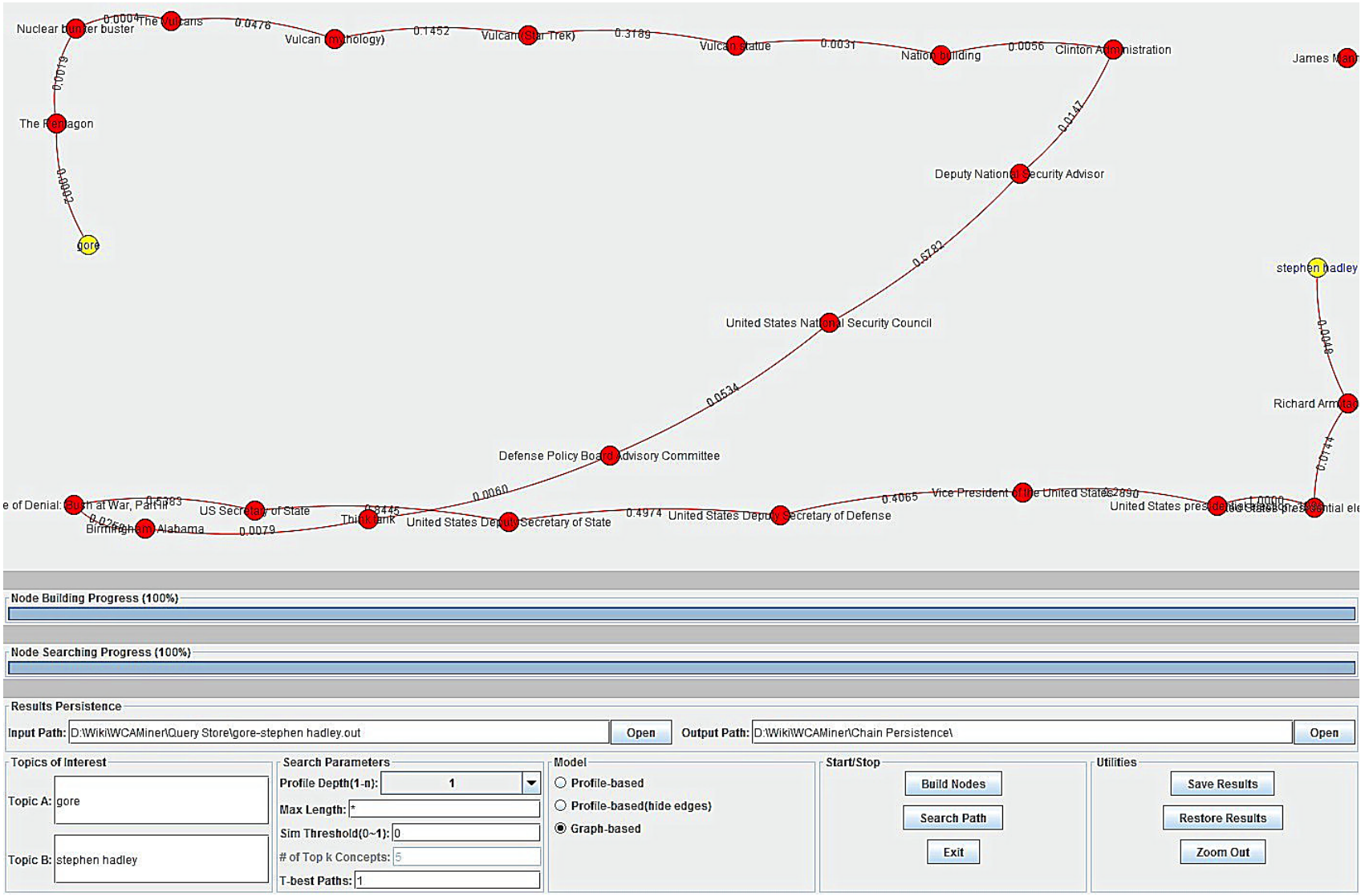


Figure 6.11. The Best Relationship Chain Discovered for the Query Pair "*Gore :: Stephen Hadley*" Using the *CAG*-based Approach

### 6.5.2. Summary

As a post analysis, the *CAG*-based approach handles a majority of the queries (including domain-specific and general queries) with a considerably high precision at various lengths. We give credit to the Wiki-integrated semantic kernel which improves the semantic relatedness measurement and the well-designed pruning strategies which filter out either noise or unfavorable concepts and associations. The ability of discovering unapparent hidden associations of the *CAG*-based approach outshines other competitors and makes our system more promising for hypothesis generation. For the missed truth chains, the reason might be our adapted Levenshtein Distance algorithm cannot identify all alias names or some alias names are not available from Wikipedia. Another reason might be the strict pruning strategies applied in our system which improve the accuracy but compromise the recall of the returned associations.

# CHAPTER 7. CONCLUSIONS

## 7.1. Conclusions

In this dissertation, we propose a comprehensive framework for discovering semantic relationships between concepts across text documents. For text mining research, the widely used text representation is based on VSM which represents text as a collection of words, however, this representation does not address the semantic contents of words without background knowledge introduced, the discovered resultsare limited to the concepts appearing in the text literally, which could lead to a great discovery loss because concepts that are closely related to the topics of interest will be viewed as completely irrelevant unless they are mentioned in the text. We overcome such limitations by presenting various mining models as follows:

a) The Semantic Path Chaining (*SPC*) model focuses on mining semantic relationships between concepts across multiple documents by taking the extensive background knowledge from Wikipedia into consideration. Specifically, we focus on detecting cross-document semantic relationships between concepts where most of them cannot be uncovered by the traditional paradigm. We also go one step further by incorporating the knowledge from Wikipedia to help identify more potential relationships that do not occur literally in the existing document corpus. The experiments were conducted using a large set of queries covering various scenarios, and compared with a purely VSM-based representation model, the original ESA method, and the approach only incorporating the Wikipedia category information. The results demonstrate the effectiveness of our proposed new hybrid solution

combing valuable resources, and show much broader and well-rounded coverage of significant relationships between concepts.

b) The kernel methods focus on representing the semantic relationships between concepts in a new space that embeds different information resources from Wikipedia. We present various types of semantic kernels by inspecting over 5,000,000 Wikipedia articles and 700,000 Wikipedia categories, so that the relationships revealed are not limited to those appearing in the document collection literally, and concept closeness can be measured in such a way that word semantics is addressed. Specifically, we employ the Explicit Semantic Analysis (ESA) technique to help build a Wiki-article content-based semantic kernel that captures concept closeness in the space of Wikipedia articles. In addition, we utilize the categorical information provided by Wikipedia to build a Wiki-category-based semantic kernel that measures semantic relatedness between concepts in the space of Wikipedia categories. To address the semantic gaps between different information resources (e.g. Wiki articles and categories), as well as the unavoidable residuals introduced by linear combination of different semantic relatedness weighting schemes, we also develop a hybrid semantic kernel that integrates both Wiki articles and categories.

c) The Concept Association Graph (*CAG*)-based mining model focuses on relationship mining in the space of Wikipedia without being limited by predetermined documents. We leverage the articles and anchors provided by Wikipedia to serve our knowledge discovery task. Different from traditional approaches coupled with either pre-defined concept dictionaries [77, 79] or information extraction engines [30, 33], the proposed approach performs much better in terms of flexibility and portability, and achieves

improved search quality in terms of accuracy. By representing the concept relationships as a graph, this model is able to perform mining tasks where user interests might be different, and can be considered as: (1) a proposal of a new knowledge discovery framework that combines information retrieval, association mining and link analysis techniques; (2) a proposal of a better modeling of knowledge representation and semantic relatedness estimation; (3) a proposal of an effective approach for hypothesis generation; (4) a proposal of graphically visualizing the knowledge discovery process; (5) a proposal of an interactive system design.

## 7.2. Limitations

Our focus in this dissertation has been on utilizing the knowledge derived from Wikipedia to serve traditional relationship discovery tasks. The *SPC* model for detecting relationships between two given topics is based on building different levels of semantic profiles. This approach adopts a very strict criterion for selecting topic-related concepts. This is based on the observation that the computational complexity significantly increases as the length of desired concept chains increasesdue to the “combinatorial explosion”resulting from generating the semantic profiles for multilevel intermediate concepts connecting two topics. Therefore, we only studied the concept chains up to length 4 in this dissertation. Specifically, given two topics of interest A and C, BP is the semantic profile built between A and C, DP is the semantic profile built between A and each concept in BP, and EP is the profile built between C and each concept in BP. The concepts inBPstand for those co-occur with both A and C in the same sentence (we call it first level co-occurrence), while the concepts in DP are those co-occur with A and eachlinking concept in BP (we call it second level co-occurrence), and the concepts in EP are

those co-occur with C and each linking concept in BP, thus guarantee the global relevance to A and C.

Another key feature of the *SPC* model is that it is coupled with a domain-specific dictionary. It could restrict the application of the *SPC* in other domains if such dictionaries are unavailable. The *CAG*-based model overcomes this limitation by interpreting the given queries directly in Wikipedia space. Since Wikipedia is a cross-domain encyclopedia, the *CAG*-based model is automatically able to handle queries across different domains. However, domain-specific ontological information is not represented in the *CAG*-based model, making it fall behind the *SPC* model in terms of interpreting the nature of relationships discovered.

Also, in this dissertation, we focus on answering queries for two given topics. In many cases, a system capable of finding the relationships for three or more topics is needed. However, we do not support such queries. One of the challenges for our system to have such features is the performance issues brought about by processing the mass data of Wikipedia.

Despite all these limitations, we believe the approaches proposed in this dissertation can benefit many other tasks such as question answering, document classification/clustering, and cross–document summarization, and we hope that this dissertation provides a solid foundation for advancing the cross-document knowledge discovery research.

### 7.3. Future Work

Wikipedia also provides some other valuable information resources which were not used in this study. These valuable resources may be combined with our defined semantic types to further contribute to ontology modeling. Furthermore, Wikipedia articles oftencontain “redirect” links, and pages pointed to by these links may indicate the synonymy relation and be helpful to

better estimate the semantic relatedness between concepts.As a cross language knowledge base, we also plan to apply Wiki knowledge into a cross-lingual setting to better serve different query purposes. Moreover, the infobox information provided by Wikipedia presents another valuable evidence source for different aspects related to the concept being described. We will be exploring the usage of these resources and evaluating their performance in our future work.

# REFERENCES


[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast Discovery of Association Rules.*Advances in knowledge discovery and data mining,12*, 307-328.

[2] Apache software foundation, hadoop mapreduce. [Online]. Available at: http://hadoop.apache.org/

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722-735). Springer Berlin Heidelberg.

[4] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM press.

[5] Bagga, A., & Baldwin, B. (1998, August). Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1* (pp. 79-85). Association for Computational Linguistics.

[6] Banerjee, S., & Pedersen, T. (2003, August). Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI* (Vol. 3, pp. 805-810).

[7] Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines.In *Proceedings of the 16th international conference on World Wide Web* (pp. 757-7660. ACM

[8] Budanitsky, A., & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, *32*(1), 13-47.

[9] Das-Neves, F., Fox, E. A., & Yu, X. (2005, October). Connecting topics in document collections with stepping stones and pathways. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 91-98). ACM.

[10] Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web (pp. 519-528). ACM.

[11] Davies, R. (1989). The creation of new knowledge by information retrieval and classification. *Journal of Documentation, 45*(4), 273-301.

[12] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107-113.

[13] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, *41*(6), 391-407.

[14] Denoyer, L., & Gallinari, P. (2007). The wikipedia xml corpus. In *Comparative Evaluation of XML Information Retrieval Systems* (pp. 12-19). Springer Berlin Heidelberg.

[15] Dumais, S. T., Letsche, T. A., Littman, M. L., & Landauer, T. K. (1997, March). Automatic cross-language retrieval using latent semantic indexing. In *AAAI spring symposium on cross-language text and speech retrieval* (Vol. 15, p. 21).

[16] Faloutsos, C., McCurley, K. S., & Tomkins, A. (2004, August). Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 118-127). ACM.

[17] Fano, R. (1961). Transmission of information.

[18] Gabrilovich, E., & Markovitch, S. (2006, July). Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI* (Vol. 6, pp. 1301-1306).

[19] Gabrilovich, E., & Markovitch, S. (2007, January). Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI* (Vol. 7, pp. 1606-1611).

[20] Gibson, D., Kleinberg, J., & Raghavan, P. (1998, May). Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space---structure in hypermedia systems: links, objects, time and space---structure in hypermedia systems* (pp. 225-234). ACM.

[21] Gooi, C. H., & Allan, J. (2004). Cross-Document Coreference on a Large Scale Corpus. In *HLT-NAACL* (pp. 9-16).

[22] Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. *arXiv preprint cmp-lg/9808002*.

[23] Grefenstette, G. (1992, June). SEXTANT: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics* (pp. 324-326). Association for Computational Linguistics.

[24] Gurevych, I., Müller, C., & Zesch, T. (2007, June). What to be?-electronic career guidance based on semantic relatedness. In *Annual Meeting-Association for Computational Linguistics* (Vol. 45, No. 1, p. 1032).

[25] Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., ... & Scheel, U. (2010, January). Faceted wikipedia search. In Business Information Systems (pp. 1-11). Springer Berlin Heidelberg.

[26] Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., & Stegemann, T. (2009). Relfinder: Revealing relationships in rdf knowledge bases. In *Semantic Multimedia* (pp. 182-187). Springer Berlin Heidelberg.

[27] Heim, P., Lohmann, S., & Stegemann, T. (2010). Interactive relationship discovery via the semantic web. In *The Semantic Web: Research and Applications* (pp. 303-317). Springer Berlin Heidelberg.

[28] Hotho, A., Staab, S., & Stumme, G. (2003). Wordnet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*.

[29] Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

[30] Jin, W., & Srihari, R. K. (2006, December). Knowledge Discovery across Documents through Concept Chain Queries. In *Proceeding of the Sixth IEEE International Conference on Data Mining Workshops* (pp. 448-452). IEEE.

[31] Jin, W., & Srihari, R. K. (2007, March). Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 807-811). ACM.

[32] Jin, W., Srihari, R. K., & Ho, H. H. (2007, October). A text mining model for hypothesis generation. In *Proceedings of the19th IEEE International Conference on Tools with Artificial Intelligence* (Vol. 2, pp. 156-162). IEEE.

[33] Jin, W., Srihari, R. K., Ho, H. H., & Wu, X. (2007, October). Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques. In *Proceeding of the Seventh IEEE International Conference on Data Mining* (pp. 193-202). IEEE.

[34] Jin, W., Srihari, R. K., & Wu, X. (2007). Mining concept associations for knowledge discovery through concept chain queries. In *Advances in Knowledge Discovery and Data Mining* (pp. 555-562). Springer Berlin Heidelberg.

[35] Jin, W., Srihari, R. K., & Singh, A. (2008, April). Generating hypotheses from the web. In *Proceedings of the 17th international conference on World Wide Web*(pp. 1211-1212). ACM.

[36] Jing, L., Zhou, L., Ng, M. K., & Huang, J. Z. (2006). Ontology-based distance measure for text clustering. In *Proceedings of the Text Mining Workshop, SIAM International Conference on Data Mining*.

[37] Kibble, R., & van Deemter, K. (2000). Coreference Annotation: Whither?. In *LREC*.

[38] Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review, 104*(2), 211.

[39] Lee, L. (1999, June). Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 25-32). Association for Computational Linguistics.

[40] Lee, M. D., Pincombe, B. M., & Welsh, M. B. (2005). An empirical evaluation of models of text document similarity. *Cognitive Science*.

[41] Lehmann, J., Schüppel, J., & Auer, S. (2007). Discovering Unknown Connections-the DBpedia Relationship Finder. *CSSW, 113*, 99-110.

[42] Lin, D. (1998, July). An information-theoretic definition of similarity. In *ICML*(Vol. 98, pp. 296-304).

[43] Luo, G., Tang, C., & Tian, Y. L. (2007, May). Answering relationship queries on the web. In *Proceedings of the 16th international conference on World Wide Web* (pp. 561-570). ACM.

[44] Mani, I., & Bloedorn, E. (1999). Summarizing similarities and differences among related documents. *Information Retrieval, 1*(1-2), 35-67.

[45] Martin, P. (2003). Correction and extension of WordNet 1.7. In *Conceptual Structures for Knowledge Creation and Communication* (pp. 160-173). Springer Berlin Heidelberg.

[46] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM, 38*(11), 39-41.

[47] Milne, D., Medelyan, O., & Witten, I. H. (2006, December). Mining domain-specific thesauri from wikipedia: A case study. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence* (pp. 442-448). IEEE Computer Society.

[48] Milne, D. (2007, April). Computing semantic relatedness using wikipedia link structure. In *Proceedings of the new zealand computer science research student conference.*

[49] Müller, C., & Gurevych, I. (2009). Using wikipedia and wiktionary in domain-specific information retrieval. In *Evaluating Systems for Multilingual and Multimodal Information Access* (pp. 219-226). Springer Berlin Heidelberg.

[50] MWDumper. Software available at: http://www.mediawiki.org/wiki/Manual:MWDumper.

[51] Neves, F. A. D. (2004). *Stepping stones and pathways: Improving retrieval by chains of relationships between documents* (Doctoral dissertation, Virginia Polytechnic Institute and State University).

[52] Otterbacher, J., Erkan, G., & Radev, D. R. (2005, October). Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 915-922). Association for Computational Linguistics.

[53] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web.

[54] Papadimitriou, S., & Sun, J. (2008, December). Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining. *In Proceeding of the Eighth IEEE International Conference on Data Minning* (pp. 512-521). IEEE

[55] Potthast, M., Stein, B., & Anderka, M. (2008). A Wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval* (pp. 522-530). Springer Berlin Heidelberg.

[56] Radev, D. R., Libner, K., & Fan, W. (2002). Getting answers to natural language questions on the Web. *Journal of the American Society for Information Science and Technology, 53*(5), 359-364.

[57] Resnik, P. (2011). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *arXiv preprint arXiv:1105.5444.*

[58] Rodríguez, M. D. B., Hidalgo, J. M. G., & Agudo, B. D. (1997). Using WordNet to complement training information in text categorization. *arXiv preprint cmp-lg/9709007*.

[59] Salahli, M. A. (2009). An approach for measuring semantic relatedness between words via related terms. *Mathematical and Computational Applications, 14*(1), 55.

[60] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613-620.

[61] Schütze, H., & Pedersen, J. O. (1997). A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing & Management,33*(3), 307-318.

[62] Scott, S., & Matwin, S. (1998, August). Text classification using WordNet hypernyms. In *Proceedings of the Workshop On Usage Of WordNet In Natural Language Processing Systems* (pp. 38-44).

[63] Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

[64] Sorg, P., & Cimiano, P. (2008). Cross-lingual information retrieval with explicit semantic analysis.

[65] Srihari, R. K., Lamkhede, S., & Bhasin, A. (2005, October). Unapparent information revelation: a concept chain graph approach. In *Proceedings of the 14th ACM international conference on Information and knowledge management*(pp. 329-330). ACM.

[66] Srihari, R. K., Li, W., Niu, C., & Cornell, T. (2003, May). Infoxtract: A customizable intermediate level information extraction engine. In *Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems-Volume 8* (pp. 51-58). Association for Computational Linguistics.

[67] Srinivasan, P. (2004). Text mining: generating hypotheses from MEDLINE.*Journal of the American Society for Information Science and Technology, 55*(5), 396-413.

[68] Strube, M., & Ponzetto, S. P. (2006, July). WikiRelate! Computing semantic relatedness using Wikipedia. In *AAA*I (Vol. 6, pp. 1419-1424).

[69] Swanson, D. R. (1986). Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine, 30*(1), 7.

[70] Swanson, D. R. (1991, September). Complementary structures in disjoint science literatures. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 280-289). ACM.

[71] Swanson, D. R., & Smalheiser, N. R. (1999). Implicit text linkages between Medline records: Using Arrowsmith as an aid to scientific discovery. *Library trends, 48*(1), 48-59.

[72] Wang, P., & Domeniconi, C. (2008, August). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 713-721). ACM.

[73] Wang, P., Hu, J., Zeng, H. J., Chen, L., & Chen, Z. (2007, October). Improving text classification by using encyclopedia knowledge. In *Proceedings of the Seventh IEEE International Conference on Data Mining* (pp. 332-341). IEEE.

[74] Weeber, M., Vos, R., Klein, H., Aronson, A. R., & Molema, G. (2003). Generating hypotheses by discovering implicit associations in the literature: a case report of a search for new potential therapeutic uses for thalidomide.*Journal of the American Medical Informatics Association, 10*(3), 252-259.

[75] Wong, S. M., Ziarko, W., & Wong, P. C. (1985, June). Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 18-25). ACM.

[76] Xu, X., Mete, M., & Yuruk, N. (2005, March). Mining concept associations for knowledge discovery in large textual databases. In *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 549-550). ACM.

[77] Yan, P., & Jin, W. (2012). Improving cross-document knowledge discovery using explicit semantic analysis. In *Proceedings of the 14th international conference on Data Warehousing and Knowledge Discovery* (pp. 378-389). Springer Berlin Heidelberg.

[78] Yan, P., & Jin, W. (2013). A New Approach for Improving Cross-Document Knowledge Discovery Using Wikipedia. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems* (pp. 291-296). Springer Berlin Heidelberg.

[79] Yan, P., & Jin, W. (2013). Mining Semantic Relationships between Concepts across Documents Incorporating Wikipedia Knowledge. In *Advances in Data Mining. Applications and Theoretical Aspects* (pp. 70-84). Springer Berlin Heidelberg.