A Study on Web Search based on Coordinate Relationships

Meng Zhao

ABSTRACT

Existing search engines always return content similar web pages or images in accordance with inputted queries. However, such similarity-based search does not always result in good results. This thesis aims at establishing methodologies to complement the deficiencies of similarity-based search by taking into account *coordinate relationships*. Coordinate relationships exist at different levels, such as term, sentence and document levels. However, many previous studies focus on coordinate relationship at the term level, especially between two terms. We extend to study on coordinate relationship between pairs of terms, moreover, coordinate relationships at the sentence and document levels and demonstrate the effectiveness for Web search. Besides, our concentrations are not limited to the text retrieval field, but also the image retrieval field. More specifically, we address the following three research topics in this thesis:

1. **Paraphrasing Sentential Queries based on Coordinate Relationships**
   The effectiveness of retrieval decreases with the increase in query length. We target at sentential queries, a type of long queries, and propose a method called *sentential query paraphrasing* for improving their retrieval performance, especially on recall. Briefly, given a sentential query, our method acquires paraphrases from the noisy Web and uses them to avoid returning no answers. We are motivated by the assumption that a relation can be represented either intensionally (referred to as *paraphrase templates*) or extensionally (referred to as *coordinate tuples*) and propose a mutual reinforcement algorithm based on it. Experimental results show that our method can acquire more paraphrases from the noisy Web. Besides, with the help of paraphrases, more Web pages can be retrieved, especially for those sentential queries that could not find any answers with its original expression.

2. **Structuring Search Results based on Coordinate Relationships**
   We propose a method to structure search results of a user-given query by distinguishing coordinate relationships from similarity relationships in the documents. We take into account documents that are mutually exclusive in semantics (called *coordinate documents*) and assume that such documents should not be grouped into the same cluster. Correspond-

ingly, we also consider documents that are mutually inclusive in semantics and assume that such documents should be grouped into the same cluster. Therefore, on the basis of these two types of constraints, documents are clustered in a manner closer to that of human cognition, e.g., news articles are organized according to events they describe. Experimental results show the effectiveness of our method and illustrate the importance of coordinate relationships in finding coordinate documents and structuring search results.

3. **Panoramic Image Search based on Similarity and Adjacency**
   We introduce a new image search method, called *panoramic image search*, which is a trial in image retrieval, and show its application to similar landscape discovery. Briefly, taken an image or a few images of a place as the input, the output is images of other places that are similar to the query place from a certain perspective, referred to as "landscape". We believe that a single image cannot completely exhibit a landscape. Therefore, we also consider the surroundings. That is the physical surroundings around the spot captured in the single image. Consequently, a set of images is used to describe a landscape. In order to find such images, we consider not only similarity relationship between images but also adjacency relationship between images, and propose an image ranking algorithm called PanoramaRank. Experimental results show the effectiveness of our method to find similar landscapes.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Professor Katsumi Tanaka. Since I joined Tanaka Laboratory, he always supports my research through his great research vision and numerous discussions. Personally, he is very kind and cares much about the future of every student.

I am grateful to my thesis committee member Professor Masatoshi Yoshikawa for thesis supervision and his sharp comments on my research during my PhD course. He has been my research advisor for six years and gave me many valuable advice, which makes me really impressed. I would like to thank my thesis committee member Professor Sadao Kurohashi for his helpful comments and suggestions to my research. I really appreciate numerous comments, constructive suggestions from my research advisor Professor Hayato Yamana at Waseda University. He spent much time discussing my research when I visited his office.

I would like to show my great appreciation to my research advisors Professor Toshikazu Wada at Wakayama University and Professor Yoshiharu Ishikawa at Nagoya University for their numerous comments, discussions and valuable feedback.

I would like to express my special thanks to Associate Professor Hiroaki Ohshima. Whenever I start a new research topic, he helped me develop the idea and make it explicit. He spent a lot of time, even his spare time, discussing my research.

I wish to thank Associate Professor Adam Jatowt, Assistant Professor Takehiro Yamamoto, Assistant Professor Makoto P. Kato, Associate Lecturer Yusuke Yamamoto for good comments and suggestions during the laboratory meeting.

I want to thank secretaries of Tanaka Laboratory: Ms. Ikebe, Ms. Sato, Ms. Shiraishi and Ms. Ashiwa for their kindly help in both of my school life and daily life in Japan. I also want to thank my current and former colleagues, especially Dr. Yoshiyuki Shoji, Dr. Kosetsu Tsukuda and Ms. Yating Zhang for their fruitful discussions.

Finally, I would like to express my deepest gratitude to my parents Longping Zhao and Hong Chen for their strong support to my study for long years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. With the help of IR technologies, we can search by a keyword, known as keyword-based IR. Conventional search engines, such as Google[1], Bing[2], are well performed on keyword queries. In contrast, the state-of-the-art IR technologies also allow us to search by an object itself, known as content-based IR (CBIR[3]). CBIR allows people to retrieve objects based on the understanding of their contents and of their components. It has attracted a lot of research interest in recent years.

However, conventional technologies in neither keyword-based IR nor CBIR are based on similarity relationship [23][16][40][4]. Therefore, with the help of CBIR technologies, people can get content similar objects in accordance with a given one. For example, given a document as an input, conventional CBIR technologies would output similar documents in surface level, or moreover in semantic level[4]. Given an image as an input, conventional CBIR technologies would output similar images. An example is shown in Figure 1.1. Given an image of the main gate of Kyoto University as the input, other images taken in front of the main gate or from similar angles are retrieved.

Search based on similarity relationship works for some information needs. However, it does

[1]http://www.google.com

[2]http://www.bing.com

[3]Note that CBIR can be also referred to content-based image retrieval, which queries by an image and searches based on visual content. However, in this thesis, we denote CBIR a broader concept, content-based information retrieval.

[4]Here, surface level refers to terms that appears in the documents, while semantic level refers to the meaning of terms and abstract topics that occur in the documents.

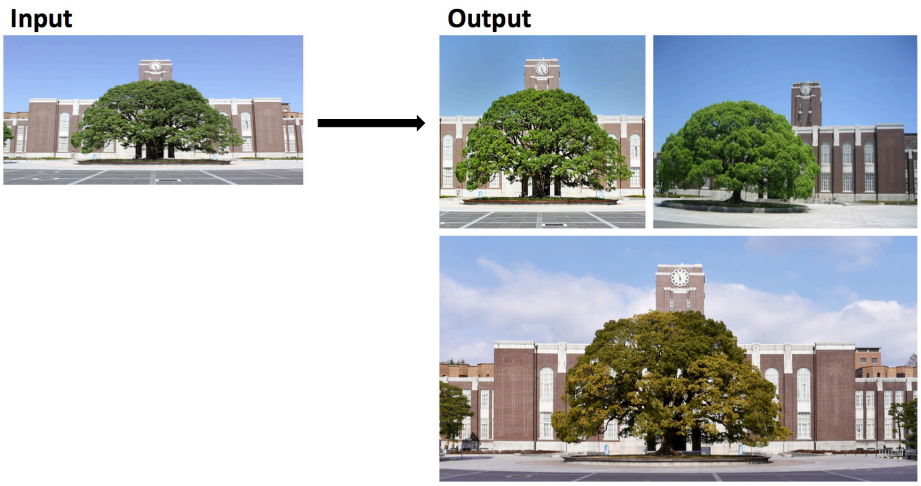


Figure 1.1: An example of image search, given an image of the main gate of Kyoto University as the input.

not guarantee that people would always obtain their desired information. Moreover, we believe that there are some information needs **more than similarity**. The following show some examples (also can be found in Figure 1.2).

Information Need 1 Finding Similar but Different Sentences

A large percentage, 90.3%, of search queries consist of less than four words on average [8], which indicates the majority of queries are still short queries. However, it has been observed that queries of length five words or more are becoming more common, with a year-over-year rate of 10% growth, while shorter queries, averaging those one to four words in length, are becoming less common, with a 2% decrease [25]. The expression rarity of long queries would be a conceivable reason why long queries, especially sentential queries, fail in retrieving any useful information. Take a Web search for example. Suppose people want to find more information about vitamin c in lemons and think of a sentential query such as "lemons are considered a high vitamin c fruit". Neither Google nor Bing return any matches for such a query (at the time of writing this thesis). However, we found that its paraphrases, such as "lemons are rich in vitamin c" or "lemons contain a high amount of vitamin c", can retrieve much more Web pages if used as queries.

In this information need, people want to find sentences that convey the same (or similar) meaning but in different vocabulary. From the above example, we know that here is no corresponding mechanism in existing search engines to restate queries by their paraphrases. Besides, it is difficult to acquire paraphrases by classical information retrieval technologies.

Information Need 2 Finding Similar but Different Documents

Traditional search technologies could retrieve similar documents in accordance with a given one. However, similar documents will bring little additional information so that it is difficult to gain more information. For example, given as the input a news article stating the occurrence of the Oregon school shooting, articles stating the occurrence of the same school shooting by other news agents are considered as better output by traditional search technologies. In the most extreme case, these news articles contain the same information, e.g., occurrence time, place. Therefore, users will not obtain more information if they read them. It seems these similar articles are less likely to make users satisfied.

We assume that a document, especially a news article, is a combination of subjects and actions, which are presented by nouns (to be specific, proper nouns) and verbs, respectively. Based on this assumption, it is intuitive to consider two kinds of "similar but different" documents. Note that documents should be under the same topic in both of the two cases. In the first case, documents have similar subjects but different actions. We know that with the development of a news event, its subjects change a little. In contrast, its actions change a lot. Therefore, in this case, similar but different documents are follow-ups (or followed-ups) of the given news article. In the second case, documents have similar actions but different subjects. We know that similar news events have similar developments. We assume that actions can locate the development phase of an event. Therefore, in this case, similar but different documents denote documents stating different events but in the same development phase. For the before-mentioned news article that states the occurrence of the Oregon school shooting, articles stating the occurrence of other school shooting events are regarded as its similar but different ones.

In this information need, people want to find documents stating the same events but in different development phases, or documents stating the same development phase but in different events. However, it is difficult for conventional CBIR technologies to accomplish the goal.

Information Need 3 Finding Similar but Different Places

Content-based image retrieval has been studied extensively. Currently, it is possible to search for images based on their contents, irrespective of features such as their color, brightness, and texture. As a result, some recent studies have focused on the development of search systems that try to detect other images, similar to the original query image. Figure 1.1 is an example of Google's image search. We find that it is possible to search for images whose contents are similar to the user-selected query image. However, among a variety of search intents, we believe that attempts to find places similar to a user-familiar

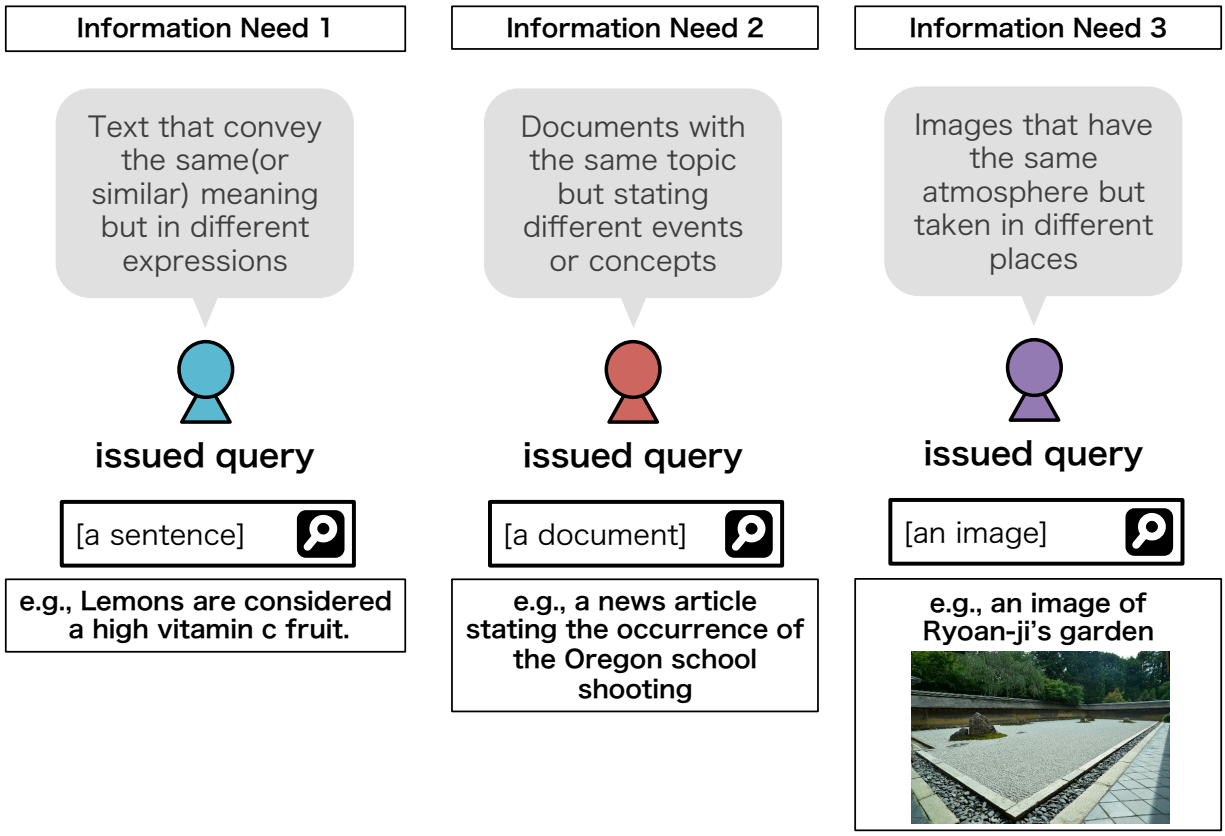


Figure 1.2: Examples of several information needs showing that searches are more than similarity.

place are common. For example, user might imagine a clock tower in front of which is a big camphor tree and the main gate of Kyoto University. The task is to find places with scenes that are similar to those in users' imaginations. In this case, the desired results should be different from the place in users' mind.

In this information need, people want to find images that have similar atmosphere but taken in different places. However, we find that it is difficult to realize this goal when we take an image that represents the scene imagined by the user and conduct an image search.

From the above discussion, we know that conventional CBIR technologies, which are based on similarity relationship, are not sufficiently supporting people doing all kinds of searches, especially in the cases that people want to find **similar but different** information. The problems (not limited to the before-mentioned ones) stem from one fundamental problem: *searches are more than similarity*. This thesis aims at establishing methodologies to complement the deficiencies out of reach to similarity-based search by taking into account **coordinate relationships** (details in Chapter 3).

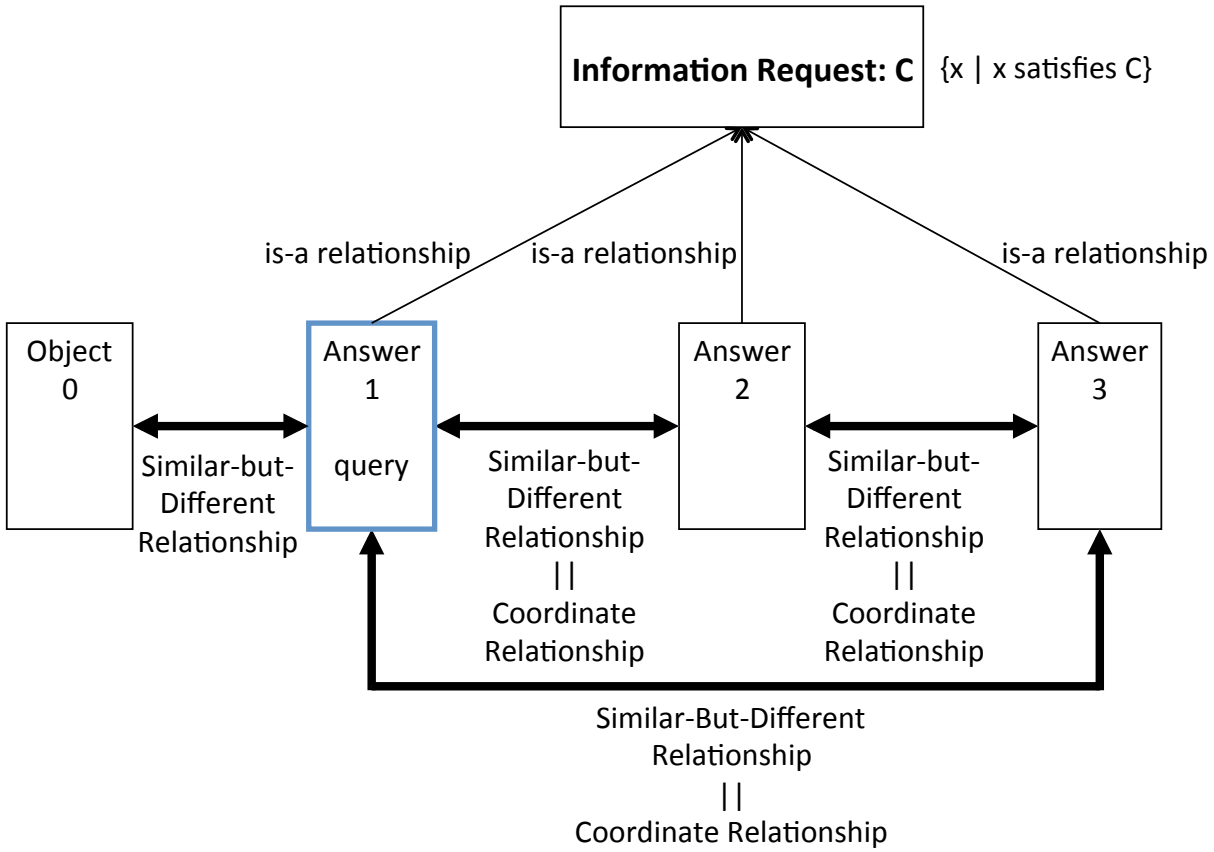


Figure 1.3: The model of similar-but-different information retrieval.

1.2 Approach

To complement the deficiencies of similarity-based search, we propose a model of similar-but-different information retrieval, shown in Figure 1.3, based on *coordinate relationships*, which allows users to search by more than similarity.

Similar to all the other IR models, at the very beginning, there is an information need in people's mind and based on this information need, a query is formulated. In CBIR, the query is an example satisfied the information need, corresponded to the blue frame in Figure 1.3. We use a one-way arrow to indicate the *is-a* relationship, viz., an object meets the information request. As we mentioned in the above, conventional CBIR technologies are based on similarity relationship, which means they only consider similarity between objects, e.g., sentences, documents, images, etc. On contrast, we tackle *similar-but-different* relationship, or more precisely, *coordinate relationship* between objects. Besides, different definitions of coordinate relationship between objects (details in Chapter 3) can be considered according to people's information request. Especially, we target at the three information needs in Figure 1.2.

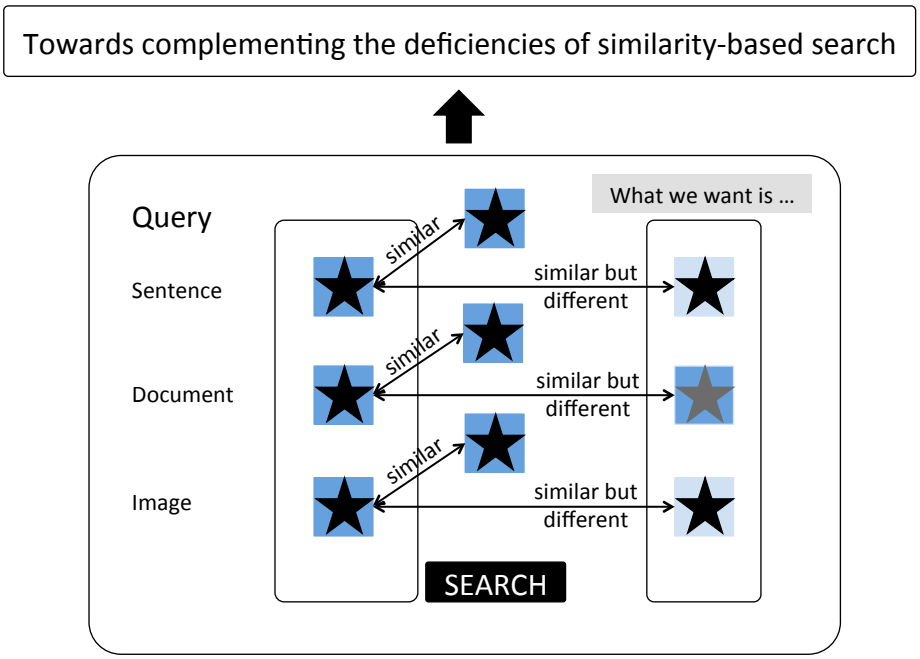


Figure 1.4: Overview of our approach.

With the help of the model, it is able to search for similar but different information. Especially, in this thesis, we concentrate on sentences, documents and images, illustrated in Figure 1.4. A square indicates a query, which can be a sentence, a document, or an image. A pentacle inside each square denotes the main idea of each object. In details, the main idea of an sentence refers to the entities occurred in the sentence; the main idea of a document refers to events or concepts stated in the document; the main idea of an image refers to the main objects taken in the image. Therefore, we can see that conventional search technologies are more likely to retrieve similar objects in accordance with the queries, shown as the squares with the same background color. As we discussed before that such kinds of results will not always make users satisfied. In some cases, users may want to search similar but different objects. For example, similar but different sentences are sentences that convey the same meaning with the query, but in different vocabulary. Similar but different documents are documents with the same topic as that of the query, but describing different events or concepts. Similar but different images are images similar in content but taken from different locations. Figure 1.4 uses squares in the right frame to present.

The key concept of our Web search method is *coordinate relationship*. Coordinate relationships exist at different levels, such as term, sentence, passage and document levels. However, many previous studies focus on coordinate relationship at the term level. On the term level, two terms are coordinate to each other if the terms share a hypernym. For example, because both "Umpqua Community College" and "Virginia Polytechnic Institute and State University" belong

to the *school* category, they are coordinate terms. Here, "school" is their common hypernym. We extend to study on coordinate relationship between tuples of terms and demonstrate the effectiveness for Web search in Chapter 4. We define tuples holding the same relation coordinated to each other. For example, the highConcentration[5] relation exists between entities in both the tuples *(lemons, vitamin c)* and *(apples, pectin)*. Therefore, *(apples, pectin)* is a coordinate tuple of *(lemons, vitamin c)*. We focus on the relation between entities in the tuples to obtain similar but different sentences.

We also extend to study on coordinate relationship between documents and illustrate the importance of coordinate relationships in structuralization of search results in Chapter 5. Simply, coordinate documents state the same topic, but describing different events or concepts. For example, a news article stating the occurrence of the Oregon school shooting is coordinate to a news article stating the occurrence of the Virginia Tech shooting. Such documents are mutually exclusive in semantics and consequently, we separate them into different groups when doing clustering to find whole stories of each event.

Besides, our concentrations are not limited to the text retrieval field, but also the image retrieval field, which means we also study on coordinate relationship between images (in Chapter 6). Similar but different images can be considered in two ways: (1) Images share the same (or similar) objects, but they capture different areas of a place[6], e.g., images of the rock garden of Ryoan-ji that are taken from different angles. Images that hold the coordinate relationship in this case represent the same atmosphere. Such images as a whole can describe a certain landscape. (2) Images have similar objects, but they are taken in different places. e.g., images of the rock garden at Ryoan-ji and Daisen-in. Based on images that hold the coordinate relationship in this case, we can find similar landscapes in different places.

1.3 Thesis Organization

In Chapter 2, we survey and briefly overview previous studies related to the research topics presented in this thesis. Each chapter after Chapter 2 in this thesis corresponds to a research shown in Figure 1.5:

- Chapter 3

  We make a detailed introduction to our key concept *coordinate relationships* in this chapter.

- Chapter 4

[5]We define highConcentration relation as the relation between a food and a certain nutrient such that the food contains a high amount of the nutrient.

[6]We denote "place" a generalized concept of "area", which means when we mention "place", we indicate a broader "area". Therefore, an area is a part of a place.

		Relation
		Element-Element
Element	Term Tuple	Paraphrasing Sentential Queries based on Coordinate Relationships [Chapter 4]
	Term Set	Structuring Search Results based on Coordinate Relationships [Chapter 5]
	Document	Structuring Search Results based on Coordinate Relationships [Chapter 5]
	Image	Panoramic Image Search based on Similarity and Adjacency[Chapter 6]

Figure 1.5: The thesis overview.

The effectiveness of retrieval decreases with the increase in query length. We target at sentential queries, a type of long queries, and propose a method called *sentential query paraphrasing* for improving their retrieval performance, especially on recall. Briefly, given a sentential query, our method acquires paraphrases from the noisy Web and uses them to avoid returning no answers. We are motivated by the assumption that a relation can be represented either intensionally (referred to as *paraphrase templates*) or extensionally (referred to as *coordinate tuples*) and propose a mutual reinforcement algorithm based on it. Experimental results show that our method can acquire more paraphrases from the noisy Web. Besides, with the help of paraphrases, more Web pages can be retrieved, especially for those sentential queries that could not find any answers with its original expression.

- Chapter 5

  We propose a method to structure search results of a user-given query to distinguish coordinate relationships from similarity relationships in the documents. We take into account documents that are mutually exclusive in semantics (called *coordinate documents*) and assume that such documents should not be grouped into the same cluster. Correspondingly, we also consider documents that are mutually inclusive in semantics and assume that such documents should be grouped into the same cluster. Therefore, on the basis of these two types of constraints, documents are clustered in a manner closer to that of human cognition, e.g., news articles are organized according to events they describe. Experimental results show the effectiveness of our method and illustrate the importance of coordinate relationships in finding coordinate documents and structuring search results.

- Chapter 6

  We introduce a new image search method, called *panoramic image search*, which is a trial

in image retrieval, and show its application to similar landscape discovery. Briefly, taken an image or a few images of a place as the input, the output is images of other places that are similar to the query place from a certain perspective, referred to as "landscape". We believe that a single image cannot completely exhibit a landscape. Therefore, we also consider the surroundings. That is the physical surroundings around the spot captured in the single image. Consequently, a set of images is used to describe a landscape. In order to find such images, we consider not only similarity relationship between images but also adjacency relationship between images, and propose an image ranking algorithm called PanoramaRank. Experimental results show the effectiveness of our method to find similar landscapes.

- Chapter 7
  This chapter summarizes the thesis and addresses some directions to be explored in future work.

CHAPTER 2

RELATED WORK

2.1 Paraphrases

We dedicate this section to the discussion of the previous work on relation extraction and paraphrase acquisition.

2.1.1 Semantic Relation Extraction

Snowball [1], KnowItAll [17], and TextRunner [68] are well-known information extraction systems. All of them extract valuable information from plain-text documents by using lexical-syntactic patterns.

Given a handful of example tuples, such as an organization-location tuple <*o,l*>, Snowball finds segments of text in the document collection where o and l occur close to each other, and analyzes the text that "connects" o and l to generate patterns. It extracts different relationships from the Web by using the bootstrap method.

KnowItAll is an autonomous, domain-independent system that extracts information from the Web. The primary focus with the system is extracting entities. The input to KnowItAll is a set of entity classes to be extracted, such as "capital", "movie" or "ceo", while the output is a list of entities extracted from the Web. Note that it only uses generic hand-written patterns, such as "including" and "is a".

Compared to these two systems in which relation types are predefined, TextRunner discovers relations automatically. Extractions take the form of a tuple $t = (e_i, r_{i,j}, e_j)$, where e_i and e_j are strings meant to denote entities, and $r_{i,j}$ is a string meant to denote a relationship between them. A deep linguistic parser is deployed to obtain dependency graph representations by parsing thousand of sentences. For each pair of noun phrases (e_i, e_j), TextRunner traverses the dependency graph, especially the part connecting e_i and e_j, to find a sequence of words that comprises

a potential relation $r_{i,j}$ in tuple t.

As our method is based on the mutual reinforcement relationship of templates and entity tuples, we can simultaneously identify templates that convey the same meaning and entity tuples that have the same relation. Therefore, it is also possible to use our method to automatically extract entity tuples of user-indicated relations.

2.1.2 Paraphrase Acquisition

Paraphrase acquisition is a task of acquiring paraphrases of a given text fragment. Some approaches have been proposed for acquiring paraphrases at word, or phrasal level. However, these techniques are designed to be only suitable for specific types of resources. Shinyama et al. [59] and Wubben et al. [63] acquired paraphrases from news articles. For example, Shinyama et al. [59] argued that news articles by different news agents reporting the same event of the same day can contain paraphrases. Thus, they proposed an automatic paraphrase acquisition approach based on the assumption that named entities are preserved across paraphrases.

Paşca and Dienes proposed a different method [51]. They use inherently noisy, unreliable Web documents rather than clean, formatted documents. They assumed that if two sentence fragments have common word sequences at both extremities, then the variable word sequences in the middle are potential paraphrases of each other. Therefore, their acquired paraphrases are almost word-, or phrase-level ones, while our aim is to obtain sentential paraphrases.

Pantel and Pennacchiotti proposed a method to harvest semantic relations in [52]. They use the Web to filter incorrect instances to get generic patterns and then use the generic patterns to extract reliable instances. However, they did not consider the mutual reinforcement between patterns and instances. In the other words, good patterns are helpful to extract good instances, and vise versa, good instances are helpful to extract good patterns. In contrast, we take the mutual reinforcement into account and can get good patterns and good instances simultaneously.

Yamamoto and Tanaka [65] also concentrated on improving search results responded by sentential queries. Unlike our focus on paraphrases, they generally collected several types of sentence substitutions, such as generalized or detailed sentences. They used these substitutions to retrieve more information.

2.2 Clustering and News Event Mining

We devote this section to a discussion of the previous work on classic clustering, event detection and tracking in news streams.

2.2.1 Clustering

Clustering algorithms aim to create groups (called clusters) that are internally coherent, but clearly different from each other. In other words, objects within the same cluster should be as similar as possible; and objects in one cluster should be as dissimilar as possible from objects in other clusters.

According to the structure of the clusters, they can be broadly grouped into two categories: flat clustering and hierarchical clustering. Flat clustering [14][37] creates a flat set of clusters, where the clusters are independent of each other. K-means [37] is the most important flat clustering algorithm, which takes an iterative refinement technique. Given k, the number of clusters, k objects are randomly selected as the initial cluster centers. The algorithm then moves the cluster centers around in space to minimize the average squared Euclidean distance of the objects from their cluster centers. This is done iteratively by alternating between two steps until a stopping criterion is met: reassigning objects to the cluster with the closest centroid, and recalculating each centroid based on the objects in each new cluster. Conversely, hierarchical clustering [20][21][48] results in a hierarchy of clusters. Therefore, the clusters can be visualized using a tree structure (a dendrogram). Its algorithms are either top-down (divisive) or bottom-up (agglomerative). For example, bottom-up algorithms regard each object as a singleton cluster at the very beginning and then agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all the objects. Hierarchical clustering algorithms do not require a pre-given number of clusters.

2.2.2 News Event Mining

Allan et al. introduced the concept of new event detection (NED) [2]. NED identifies news stories that discuss an event that has not been reported in the past. They proposed a possible definition of an event as something that happens at a particular time and place. They found that news stories about the same event often occur in clumps , and that there must be something about the story that makes its appearance worthwhile. Finally, they used a single pass clustering algorithm to detect new events. Kumaran and Allan [30] tackled the same problem and employed text classification techniques as well as named entities to improve the performance of NED.

Compared to NED, Yang et al. proposed the concept of retrospective news event detection (RED) [67]. RED is defined as the discovery of previously unidentified events in a historical news corpus. Both the contents and time information contained in a news article are very helpful for RED. Because multiple studies, including [67], only focus on the use of the contents, Li et al. [34] considered a better representation of news events, which effectively models both the contents and the time information. Although NED and RED can detect and track news events in

some way, but lack of connections between events to show their relationships.

Nallapati et al. [45] made a similar attempt to ours. They also thought that viewing a news topic as a flat collection of stories is not efficient for users to quickly understand the topic. Therefore, they introduced an event model to capture the rich structure of news events and their dependencies on a news topic, such as causality or temporal-ordering between pairs of news events. Their algorithm first groups news stories into unique events in the topic using agglomerative clustering with time decay and then constructs dependencies among them.

Feng and Allan [18][19] also dealt with the problem that news topics are treated as a flat list, ignoring the intrinsic connections between each story. They clustered text passages and then created links with scenario-specific rules to generate incident threading [18]. Then, they removed the assumption that a news story covers a single topic, and consequently, extended the incident threading to the passage level [19].

2.3 Image Retrieval

The use of a set of local interest points for image matching can be traced back to 30 years ago. Recently, there have been extensive studies into methods used to find words, in the same manner as those used in natural language processing and information retrieval, (referred to as visual words). This has been done in order to generate a similar process for retrieval using visual words to search for content-similar images based on Term Frequency Inverse Document Frequency (TF-IDF). Descriptors extracted from the local region are applied to a clustering algorithm, such as k-means, to create clusters that refer to visual words. In a variety of interest point detectors and feature descriptors, the Speeded Up Robust Features (SURF) proposed by Bay et al. in 2006 [6] is a novel scale- and rotation-invariant detector-descriptor. They apply integral image for image convolutions, and a Hessian matrix-based method for the detector (Fast-Hessian detector), and calculate the Haar-wavelet to identify the orientation of the interest points for invariance to rotation. Finally, 64 dimensional descriptors are determined.

Nister and Stewenius [46] propose a scheme, which is robust to background clutter and occlusion. They use Scale-Invariant Feature Transform (SIFT)[36], a feature descriptor that inspired SURF, for extraction from local regions. They then perform k-means clustering on extracted local region descriptors to hierarchically generate a vocabulary tree. However, k represents the number of children for each node of the tree, as opposed to the final number of clusters. As a result, for the purpose of retrieving each descriptor, it is required to traverse the tree and at each level, the descriptor is compared to each current cluster center and the closest one is chosen to continue propagating downwards.

Regarding the overlap among images, we determined that it resembles the construction of

panoramas, in which detection of regions that can be connected is required. Brown and Lowe[11] describe an approach based on extracting invariant local image features to select matching images. They extract SIFT features from all of the images and determine k nearest-neighbors for each feature using a kd-tree. Then they detect m images(they use $m = 6$) as candidate matching images, which have the maximum number of feature matches to the current image. Next, they use RANSAC, an iterative method to estimate parameters of a mathematical model, to find geometrically consistent feature matches, with the purpose of determining the homography between pairs of images. To verify image matches, they employ a probabilistic model. Up until this step, the connected parts of the panorama are confirmed.

What we called adjacency is essentially a degree of spatially verified inliers, which denotes how much overlap there is between two images. Although it has been used in re-ranking of retrieved images in [13][33][55][56], in which it is called spatial verification, their objective is to retrieve a specific object, while we aim to search the surroundings of a specific object rather than the object itself. Especially, in [33], an approach for modeling landmark sites is stated, which seems a similar job as what we are trying to do. However, note "landmark" is a particular part of a building, while "landscape" is a whole environment, containing "landmark" and also its surroundings. Object-based image retrieval, which devotes to associate a set of images of the same object, has been extensively explored. However, as far as know, visual surrounding search has not been extensively studied yet.

CHAPTER 3

COORDINATE RELATIONSHIPS

In this chapter, we address the key concept "coordinate relationships" in the thesis and discuss the relationships between terms, term tuples, term sets, sentences, documents and images, respectively. Here, we use the symbol "$\|$" to express the coordinate relationship between elements.

3.1 Coordinate Relationship between Terms

Coordinate relationships exist at different levels, such as term, sentence and document levels. Previous studies [43][47][60] concentrate on coordinate relationship at the term[1] level and define it between terms as below:

$$t \parallel_C t' \Leftarrow \exists C(t \text{ belongs to } C \wedge t' \text{ belongs to } C)$$

where C denotes a concept. In other words, two terms are coordinate to each other if they share any common hypernym. Figure 3.1 is a diagram that shows the coordinate relationship between terms. However, since the root of any entity is *object*, any two terms can be considered as coordinate to each other. Consequently, we can figure out that when considering the coordinate relationship between terms, actually it is needed to explicitly or implicitly indicate a hypernym that they share. We add a subscript C after the symbol "$\|$" to denote a certain hypernym.

Take the terms "lemon" and "grapefruit" for example. Because both of them belong to the *citrus fruit* category, they are coordinate terms. Here, "citrus fruit", which corresponds to *C* in the definition, is their common hypernym.

3.2 Coordinate Relationship between Term Tuples

When considering the coordinate relationship between term tuples, we borrow the idea of Bollegala et al. in their work [10] that a relation can be expressed extensionally by enumerating all the

[1] Actually, it also includes the phrase.

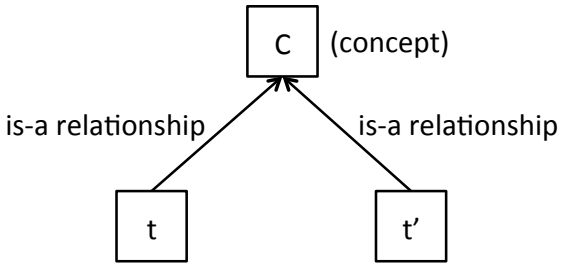


Figure 3.1: Coordinate relationship between terms.

instances (represented by term tuples) of that relation. Therefore, we can get a similar diagram, shown in Figure 3.2, to Figure 3.1. Accordingly, we define the coordinate relationship between term tuples as below:

$$
\begin{aligned}
(t_1, t_2) \parallel_{C_1,C_2,R} (t_1', t_2') \Leftarrow\ & \exists C_1(t_1 \text{ belongs to } C_1 \wedge t_1' \text{ belongs to } C_1) \wedge \\
& \exists C_2(t_2 \text{ belongs to } C_2 \wedge t_2' \text{ belongs to } C_2) \wedge \\
& \exists R((t_1, t_2) \text{ belongs to } R \wedge (t_1', t_2') \text{ belongs to } R)
\end{aligned}
$$

where C_i denotes a concept, R denotes a relation. In other words, two term tuples are coordinate to each other if their corresponding terms are coordinate to each other, meanwhile both of them represent the same relation.

Take the *highConcentration* relation for example. An extensional definition of *highConcentration* is a set of all pairs of a food and a certain nutrient in which the food is a rich source of the nutrient, including but not limited to

- *(lemons, vitamin c)*
- *(apples, pectin)*
- *(strawberries, potassium)*
- *(raspberries, fiber)*
- *(bananas, iron)*

Therefore, according to our definition, *(apples, pectin)* is a coordinate tuple of *(lemons, vitamin c)*, because *lemons* is coordinate to *apples* under the hypernym such as *fruits*, *vitamin c* is coordinate to *pectin* under the hypernym such as *nutrient*, and there exists the *highConcentration* relation (corresponding to R in the definition) both between *lemons* and *vitamin c* and between *apples* and *pectin*.

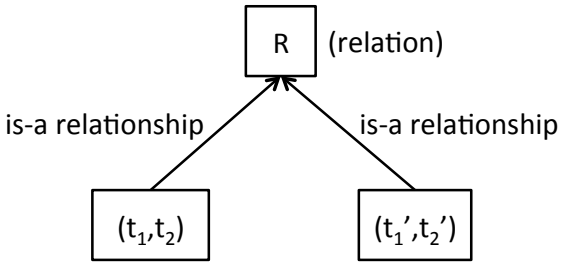


Figure 3.2: Coordinate relationship between term tuples.

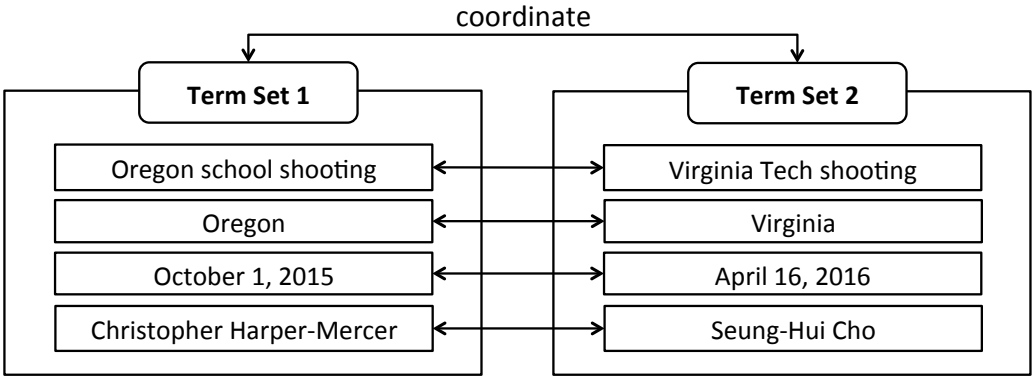


Figure 3.3: An example of two term sets coordinate to each other.

3.3 Coordinate Relationship between Term Sets

Because a term set is composed of multiple single terms, we define the coordinate relationship between term sets based on that between terms:

$$T \parallel T' \Leftarrow \forall t \in T\ \exists t' \in T'\ \exists C(t \text{ belongs to } C \wedge t' \text{ belongs to } C) \wedge$$
$$\forall t' \in T'\ \exists t \in T\ \exists C(t \text{ belongs to } C \wedge t' \text{ belongs to } C)$$

where C denotes a concept. In other words, two term sets are coordinate to each other if for each term in a set, we can find a coordinate term in another set.

Figure 3.3 shows an example of two term sets coordinate to each other. A two-way arrow between two terms indicates that there holds the coordinate relationship between these two terms. Therefore, from this figure, we can find that for each term in the term set 1, there exists a coordinate term in the term set 2. According to our definition, the two term sets are coordinate to each other.

3.4 Coordinate Relationship between Sentences

In this section, let us consider the coordinate relationship between sentences.

We assume that a sentence can be mapped by a template and an entity tuple, denoted by $s = (tem, tup)$, where tem indicates the template, tup indicates the entity tuple that can be also

written as $(t_1, t_2, ...)$. Take as an example the sentence "lemons are rich in vitamin c". It can be generated by the following two parts:

- the template: ***X** are rich in **Y***

- the entity tuple: *(lemons, vitamin c)*

Still, we borrow the idea of Bollegala et al. in their work [10] that a relation can be defined intensionally by listing all the paraphrase templates of that relation. Here, paraphrase templates denote the templates that convey the same meaning but in different expression. Take the *high-Concentration* relation for example. An intensional definition of *highConcentration* is described by templates, including but not limited to

- ***X** are rich in **Y***

- ***X** are an excellent source of **Y***

- ***X** are full of **Y***

- ***X** contain a high amount of **Y***

- ***X** are abundant in **Y***

Because a characteristic of the coordinate relationship between two elements is that they are "similar but different" to each other, there are two ways when considering similar but different points between two sentences.

Case 1

Two sentences convey the same meaning but in different vocabulary, defined as

$$s \parallel s' \Leftarrow \exists M(tem \text{ belongs to } M \wedge tem' \text{ belongs to } M) \wedge tem \neq tem' \wedge tup = tup'$$

Here, M denotes a meaning.

Because we already know in the above example that the five templates expressing the *high-Concentration* relation, filling the same entity tuple, e.g., *(lemons, vitamin c)*, in these five templates can get paraphrases as below:

- *Lemons are rich in vitamin c.*

- *Lemons are an excellent source of vitamin c.*

- *Lemons are full of vitamin c.*

- *Lemons contain a high amount of vitamin c.*

- *Lemons are abundant in vitamin c.*

Since all these sentences express the *highConcentration* relation between *lemons* and *vitamin c* but in different representations, they are similar but different. Consequently, they are considered as coordinate to each other.

We can regard the coordinate relationship between sentences in this case as a definition in a narrow sense.

Case 2

Here, we consider the coordinate relationship between sentences in a broad sense, which is based on the coordinate relationship between term tuples.

Since a sentence can be mapped by a template and an entity tuple, we can consider another case that two sentences are similar but different to each other. That is their templates convey the same meaning (including the extreme case that they are exactly the same), but their entity tuples are different, or more precisely, coordinate to each other, defined as

$$s \parallel s' \Leftarrow \exists M(tem \text{ belongs to } M \wedge tem' \text{ belongs to } M) \ \wedge tup \parallel tup'$$

Here, M denotes a meaning and the representation of the coordinate relationship between term tuples tup and tup' is simplified.

As we introduced an example of coordinate tuples in Section 3.2, we can obtain the following sentences by mapping the coordinate tuples in the paraphrase templates:

- *Lemons are rich in vitamin c.*

- *Apples are an excellent source of pectin.*

- *Strawberries are full of potassium.*

- *Raspberries contain a high amount of fiber.*

- *Bananas are abundant in iron.*

All these sentences express the *highConcentration* relation, but in different sentences, there are different entity tuples that hold the relation. From this perspective, they are also similar but different to each other. Consequently, they are considered as coordinate.

We tackle with the coordinate relationship between sentences introduced in the Case 1 in this thesis (details in Chapter 4).

3.5 Coordinate Relationship between Documents

In this section, let us consider the coordinate relationship between documents.

We know that a document is composed of many sentences. Therefore, we can consider the coordinate relationship between documents based on that between sentences. However, a document can be also considered as a combination of many terms. In this thesis, we assume that a document, especially a news article, is a combination of subjects and actions, denoted by $d = (sub, act)$, where sub indicates the set of subjects, act indicates the set of actions, which are presented by nouns (to be specific, proper nouns) and verbs, respectively.

Take Article 1 for example. It is a combination of the following two sets:

- the subject set: {*the Oregon shooting, October 1, 2015, the UCC campus, Roseburg, Oregon, United States, Christopher Harper-Mercer*}
- the action set: {*occur, enroll, shoot, injure, wound, commit suicide*}

Article 1

The Oregon shooting ***occurred*** *on October 1, 2015 at the UCC campus near Roseburg, Oregon, United States. Christopher Harper-Mercer, a 26-year-old enrolled at the school, fatally* ***shot*** *an assistant professor and eight students in a classroom. Seven to nine others were* ***injured****. After being wounded by two police officers, the gunman* ***committed suicide*** *by shooting himself in the head.*

Article 2

Ten people were killed when a gunman opened fire at Oregon's Umpqua Community College on Thursday, forcing the nation to face yet another mass shooting. Seven other people were injured, and the shooter is dead. Earlier estimates had put the number of people hurt much higher. Multiple law enforcement officials familiar with the investigation identified the gunman as 26-year-old Christopher Harper-Mercer.

Article 3

The Virginia Tech shooting ***occurred*** *on April 16, 2007, on the campus of Virginia Polytechnic Institute and State University in Blacksburg, Virginia, United States. Seung-Hui Cho, a senior at Virginia Tech,* ***shot and killed*** *32 people and* ***wounded*** *17 others in two separate attacks, approximately two hours apart, before* ***committing suicide****.*

Based on this assumption, it is intuitive to consider two kinds of "similar but different" documents. Note that documents should be under the same topic in both of the two cases.

Case 1

Two documents describe the same event but different development phases, defined as

$$d \parallel d' \Leftarrow \exists C(d \text{ belongs to } C \wedge d' \text{ belongs to } C) \ \wedge sub \approx sub' \wedge act \neq act'$$

Here, C denotes a topic.

We know that with the development of a news event, its subjects change a little. In contrast, its actions change a lot. Therefore, in this case, two coordinate documents have similar subjects but different actions. Moreover, a document is a follow-up[2] (or a followed-up) of another document.

Given Article 1, which describes the occurrence of *the Oregon school shooting*, its similar but different documents can be the follow-ups of this article, e.g., the launch of a campus-wide search for explosives, more executive action on the subject of gun control.

Case 2

Two documents describe the same development phase but different events, defined as

$$d \parallel d' \Leftarrow \exists C(d \text{ belongs to } C \wedge d' \text{ belongs to } C) \ \wedge sub \parallel sub' \wedge act \approx act'$$

Here, C denotes a topic.

We know that similar news events have similar developments. We assume that actions can locate the development phase of an event. Therefore, in this case, two coordinate documents have similar actions but different subjects, or more precisely, their subjects are coordinate to each other.

For example, given Article 1, which describes the occurrence of *the Oregon school shooting*, its similar but different documents can be also the articles stating the occurrence of other school shooting events, such as Article 3 about *the Virginia Tech shooting*.

By carefully observing Article 1 and Article 3, we find that

1). the subjects are coordinate to each other in some way. For example, the occurrence time of the Oregon shooting, *October 1, 2015*, is coordinate to the occurrence time of the Virginia Tech shooting, *April 16, 2007*. The occurrence location of the Oregon shooting, *UCC campus*, is coordinate to that of the Virginia Tech shooting, *campus of Virginia Polytechnic Institute and State University*.

2). the actions in both these articles are similar (see terms or phrases in bold). For example, at the beginning of both these articles, it states the occurrence of the event, using exactly the same term "occurred". When it comes to the injuries and deaths of each shootings, Article 1 used the term "injured", while Article 3 used the term "wounded". Even though these two terms have different surface forms, they have the same semantic meaning.

[2]Follow-up is an article giving further information on a previously reported news event.

An image of the garden at Ryoan-ji An image of the garden at Ryoan-ji

Figure 3.4: An example of two images coordinate to each other, corresponding to the Case 1.

An image of the garden at Ryoan-ji An image of the garden at Daisen-in

Figure 3.5: Another example of two images coordinate to each other, corresponding to the Case 2.

Therefore, according to our definition, these two articles are coordinate to each other.

We tackle with the coordinate relationship between documents introduced in the Case 2 in this thesis (details in Chapter 5).

3.6 Coordinate Relationship between Images

Our concentrations are not limited to the text retrieval field, but also the image retrieval field. As a result, we also study the coordinate relationship between images in this thesis.

Before introducing the coordinate relationship between images, we clarify two terminologies at first. We denote "place" a generalized concept of "area", which means when we mention "place", we indicate a broader "area". For example, Ryoan-ji is the so-called "place", which contains the whole region of the temple. The rock garden at Ryoan-ji corresponds to the so-called "area". Therefore, an area is a part of a place.

As we mentioned before, a characteristic of the coordinate relationship between two elements is that they are "similar but different" to each other. Consequently, there are two ways when considering similar but different points between two images. Note that images should be under the same atmosphere in both of the two cases.

Case 1

Two images share the same objects, but they capture different areas of a place, viz. taken from different angles, defined as

$$\begin{aligned} i \parallel i' \Leftarrow\ & \exists C(i \text{ belongs to } C \wedge i' \text{ belongs to } C) \wedge \\ & placeName(i) = placeName(i') \wedge \\ & imageFeature(i) \approx imageFeature(i') \end{aligned}$$

Here, C denotes an atmosphere, $placeName(i)$ indicates the name of the place described in the image i, $imageFeature(i)$ indicates the image features of i.

Figure 3.4 shows an example of two images similar but different to each other in this case. From this figure, we can see that both of these two images are about the rock garden at Ryoan-ji. Therefore, they share some common objects. However, they are taken from different angles. As a result, they describe different areas of Ryoan-ji. Or more precisely, they cover slightly different parts of the famous rock garden in the temple. In a word, two coordinate images describe the same place and are visually similar to each other in this case.

Images that hold the coordinate relationship in this case represent the same atmosphere in a place. Since a landscape is an area that produces the same response from most people, we think such images as a whole can describe a certain landscape of a place.

Case 2

Two images have similar objects, but they are taken in different places, defined as

$$\begin{aligned} i \parallel i' \Leftarrow\ & \exists C(i \text{ belongs to } C \wedge i' \text{ belongs to } C) \wedge \\ & placeName(i) \parallel placeName(i') \wedge \\ & imageFeature(i) \approx imageFeature(i') \end{aligned}$$

Here, C also denotes an atmosphere.

Figure 3.5 shows an example of two images similar but different to each other in this case. From this figure, we can see that both of these two images are about the traditional Japanese dry landscape garden. Hence, there are some common features in the two images. However, they are taken in different places, e.g., the left image is taken in Ryoan-ji, the right image is taken in Daisen-in. In a word, the two images partially show the traditional Japanese dry landscape garden at different temples. Therefore, two coordinate images are visually similar to each other but describe different places in this case.

Based on images that hold the coordinate relationship in this case, we can find similar landscapes in different places.

We tackle with the coordinate relationship between images introduced in both of the Case 1 and the Case 2 in this thesis (details in Chapter 6).

CHAPTER 4

PARAPHRASING SENTENTIAL QUERIES BASED ON COORDINATE RELATIONSHIPS

4.1 Introduction

Search engines, such as Google[1] and Bing[2], provide convenience for users to obtain useful information by issuing queries based on their information needs. Therefore, they have become the major gateways to the huge amount of information on the Web. Bendersky and Croft [8] stated that Web search queries are mostly short queries whose length is less than four words on average. However, it has been reported that queries of length five words or more are becoming more common, with a year-over-year rate of 10% growth, while shorter queries, averaging those one to four words in length, are becoming less common, with a 2% decrease [25]. Several studies have proved that compared to short queries, long queries can provide more information in the form of context, consequently providing a better way for conveying complex and sophisticated information needs [29][31][32][54]. Therefore, long queries are used in many different applications, such as question answering (QA) search [64] and judgement of fact trustworthiness [66]. However, the effectiveness of retrieval for long queries is generally lower than that for short queries [25].

The expression rarity of long queries would be a conceivable reason why long queries, especially sentential queries, fail in retrieving any useful information. Take a Web search for example. Suppose users want to find more information about pectin in apples and think of a sentential query such as "apples pop a powerful pectin punch". None of the two aforementioned search engines return any matches for such a query (at the time of writing this paper).

Several studies [3][7][31][39] have concentrated on improving the retrieval effectiveness of

[1] http://www.google.com
[2] http://www.bing.com

long queries. All are based on the assumption that long queries always contain extraneous terms. Besides, they can be broadly grouped into two categories: query reduction approach and query re-weighting approach. Query reduction is aimed at improving the performance of long queries by eliminating redundancy. Therefore, a long query is reduced to a concise version by removing one or more terms. Query re-weighting is focused on identifying important or verbose terms in long queries and assigning different weights to them.

In general, long natural language queries can be divided into two categories: sentential queries and joint phrase queries. Obviously, the former are sentences in form, such as "What is the highest mountain in Africa?", while the latter are sequences generated by several separate phrases, such as "2015 Uefa Super Cup FC Barcelona Sevilla FC Pedro score", which is joint by "2015 Uefa Super Cup", "FC Barcelona", "Sevilla FC" and "Pedro score". In this study, we target at sentential queries and focus on improving their poor performance by using other queries that convey the same meaning. We call it ***sentential query paraphrasing***. Contrary to previous assumption, we argue that separate terms or phrases from a long query may lead to the missing of some information or query drift. Neither query reduction approach nor query re-weighting can exhibit an non-disappointing performance. This is because the query itself "apples pop a powerful pectin punch" is an indivisible whole. Its meaning cannot be completely expressed by any portion of it. In this case, we rewrite the original query by its paraphrases, such as "apples contain a lot of pectin" and "apples are rich in pectin". We can obtain enough Web pages with detailed information by submitting those paraphrases to the Web and aggregating their search results.

Sentential query paraphrasing is also effective in estimating the credibility of facts. Here, we define *fact* as an item of knowledge or a piece of information. It has a variety of different expressions in the surface form. Correspondingly, a certain expression of a fact is defined as a *fact statement*. For example, there is a fact about high level of pectin contained in apples. This fact can be represented in, but not limited to, the following ways:

- *Apples are rich in pectin.*
- *Apples are a great source of pectin.*
- *Apples contain a high amount of pectin.*
- *Apples are packed with pectin.*
- *Apples are abundant in pectin.*
- *Apples have high pectin content.*

Each different way that represented the fact is a fact statement. Hence, the sentence "apples are rich in pectin" is a fact statement. We assume the credibility of a fact is high if people often mention it on the Web. Based on this assumption, a naive way to judge fact credibility is to check its occurrence on the Web. However, this trial always fails. The reason is that although there is a variety of different expressions for a fact, it might be difficult to think of these expressions as many as possible. In the most extreme case, we may only think of one expression, which leads to failure of fact credibility judgement. For example, suppose we want to estimate whether a fact is credible, but can only think of a statement like "apples are abundant in pectin". Actually, this statement is seldom used on the Web. If we judge the fact credibility only based on this statement, we would draw an erroneous conclusion that apples do not have a high amount of pectin. So it is likely to draw erroneous conclusion by only observing the occurrence of a certain fact statement. However, if we also take other statements of the fact into consideration, the ones that convey the same meaning as the given statement, it is more likely for us to come to the right conclusion. For example, other fact statements, such as "apples are rich in pectin" or "apples are a great source of pectin", are widely used on the Web. If we estimate the credibility of the fact also based on these statements, we could draw the correct conclusion that apples are a high pectin fruit.

Based on the intensional-extensional relation representation, our method finds sentential paraphrases from the noisy Web instead of domain-specific corpora. Bollegala et al. [10] stated that a relation can be defined intensionally by listing all the paraphrase templates of that relation. It can be also expressed extensionally by enumerating all the instances of it. Take the **highConcentration** relation[3] for example. An intensional definition of **highConcentration** is described with templates, including but not limited to ***X** are rich in **Y*** and ***X** are an excellent source of **Y***. An extensional definition of **highConcentration** is a set of all pairs of a food and a certain nutrient in which the food is a rich source of the nutrient, including but not limited to *(lemons, vitamin c)* and *(apples, pectin)*. Given a sentential query, our method first extracts templates and entity tuples from the Web, respectively. During the extractions, several filters and limitations are added to eliminate partial inappropriate templates and entity tuples. Finally, a mutually reinforcing approach is used to identify different templates that convey the same meaning with the given template.

The remainder of the paper is organized as follows. In Section 4.2, we define the sentential query paraphrasing problem and introduce the overview of our proposed method. In Section 4.3, we describe the core of our algorithm. Section 4.4 gives more details when adapting to the fact credibility judgement application. We explain the evaluation results in Section 4.5. Finally, we

[3]We define highConcentration relation as the relation between a food and a certain nutrient such that the food contains a high amount of the nutrient.

conclude the paper in Section 6.7.

4.2 Sentential Query Paraphrasing Problem

4.2.1 Problem Definition

In this section, we give a definition to ***sentential query paraphrasing***. As we discussed in Section 5.1, our notion is to substitute a sentential query by its frequently used paraphrases to retrieve more answers. Therefore, the problem can be described as follows:

- Input: A sentence in which an entity tuple is indicated

- Output: Sentences that convey the same meaning

However, the problem slightly changes according to different applications. We introduce two applications: ***judgement of fact credibility*** and ***QA search*** in Section 4.4.

Briefly, in the former application, we estimate fact credibility by observing the occurrence of both a fact and its paraphrases on the Web. We are concerned with different expressions of entities in facts. Take the fact "Lemons are considered a high vitamin c fruit" for example. Our concern is different expressions about "lemons" and "vitamin c" with the same meaning as that in the fact. Moreover, these expressions should be practically used by people, since a fact seldom mentioned by people is unlikely to be credible. Table 4.1 lists the top 15 running results of our method for the aforementioned fact.

A bottleneck for question-answering systems is the identification of different expressions with the same meaning between a user's question and existing questions or answers. In QA search, we paraphrase users' questions in order to retrieve more answers. Here, we are only concerned with templates in questions. For example, given question "What is the highest mountain in Africa?" as the input, our method outputs its paraphrases such as "What is the highest peak in Africa?". But actually, what we do is to find other expressions of ***X*** *is the highest mountain in* ***Y***, which is the template in the above question.

4.2.2 Overview of the Proposed Method

There are three steps in our proposed method. Figure 4.1 shows its overview. Black box indicates a processing, while white box indicates the input and output for each processing. Take as an example the input of the template ***X*** *are considered a high* ***Y*** *fruit* and the entity tuple *(lemons, vitamin c)*.

In step 1, our method extracts candidate entity tuples from the Web through tuple extraction according to the input template. Similarly, our method extracts candidate templates from the Web through template extraction according to the input entity tuple. In some cases, we could not

Table 4.1: Top 15 paraphrases when given the template ***X** are considered a high **Y** fruit* and the entity tuple *(lemons, vitamin c)* as the input.

1.	lemons are an excellent source of vitamin c
2.	lemons are rich in vitamin c
3.	lemons are high in vitamin c
4.	lemons are packed with vitamin c
5.	vitamin c obtained from lemons
6.	lemons have a very high vitamin c content
7.	boosts the immune system lemons are high in vitamin c
8.	lemons contain a high amount of vitamin c
9.	lemons are a rich source of vitamin c
10.	the best know natural sources of vitamin c are the citrus fruit such as lemons
11.	lemons are also sources of vitamins and minerals other than vitamin c
12.	lemons and limes help keep your skin looking its best because they're rich in vitamin c
13.	lemons are vitamin c rich citrus fruits
14.	it is no longer news that we all need to use lemons every day because of the high amounts of vitamin c
15.	lemons contain vitamin c

obtain enough candidate entity tuples or templates. Hence, several frequently appeared tuples are used to extract more candidate templates, corresponding to mark (1) in Figure 4.1. In the same way, several frequently appeared templates are used to extract more candidate entity tuples, corresponding to mark (2) in Figure 4.1. Note that currently, such processing is taken place only once. Finally, we obtain candidate templates, such as ***X** are rich in **Y***, ***X** contain **Y***, and candidate entity tuples, such as *(apples, pectin)*, *(strawberries, fiber)*.

In step 2, we take candidate templates and candidate entity tuples as the input for the mutual reinforcement algorithm to identify paraphrase templates and coordinate tuples at the same time. For example, ***X** are rich in **Y***, ***X** are full of **Y*** are judged as paraphrase templates of the original template ***X** are considered a high **Y** fruit*.

Finally, in step 3, we combine paraphrase templates with the input entity tuple to obtain paraphrases. Hence, we have "Lemons are rich in vitamin c" and "Lemons are full of vitamin c" as the paraphrases of "Lemons are considered a high vitamin c fruit".

4.3 Mutual Reinforcement between Templates and Entity Tuples

In this section, we describe the core of our algorithm, referred to the step 2 in Figure 4.1.

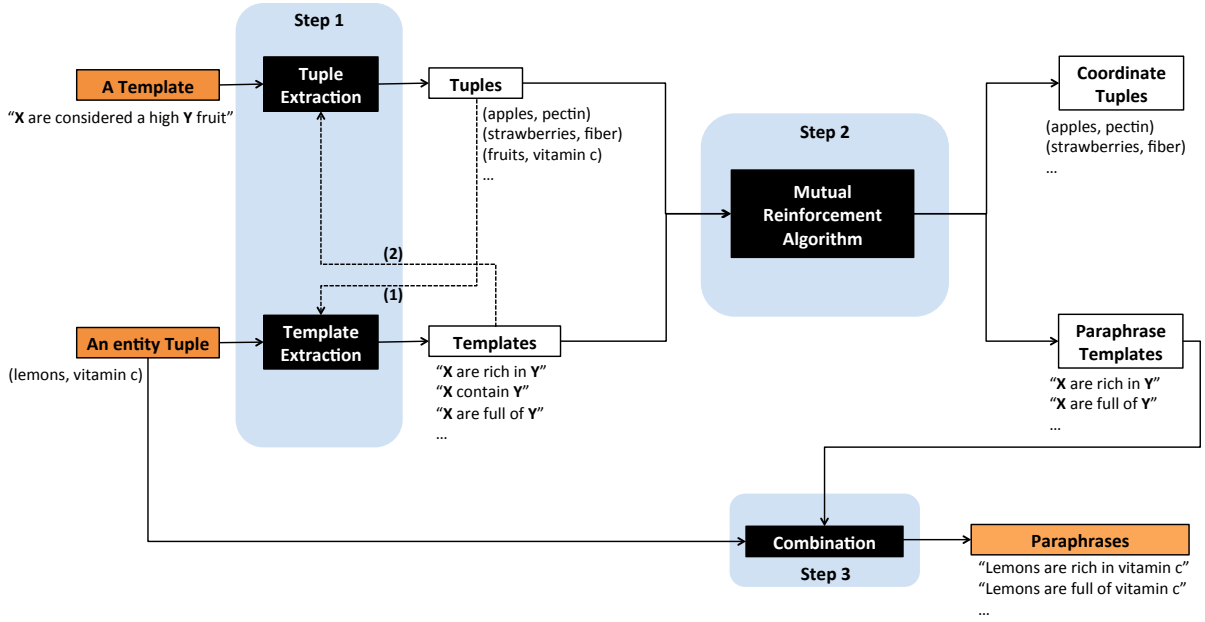


Figure 4.1: Overview of the proposed method.

4.3.1 Intensional-Extensional Representation for a Relation

Bollegala et al. [10] stated that a relation can be defined intensionally by listing all the paraphrase templates of that relation. It can be also expressed extensionally by enumerating all the instances of that relation. Take the **highConcentration** relation for example. An intensional definition of **highConcentration** is described by templates, including but not limited to

- ***X** are rich in **Y***
- ***X** are an excellent source of **Y***
- ***X** are full of **Y***

An extensional definition of **highConcentration** is a set of all pairs of a food and a certain nutrient in which the food is a rich source of the nutrient, including but not limited to

- *(lemons, vitamin c)*
- *(apples, pectin)*
- *(strawberries, potassium)*

Entity tuples holding the same relation are defined as "coordinated" to each other. Therefore, *(apples, pectin)* is a coordinate entity tuple of *(lemons, vitamin c)*. Some of the terminology used in this paper is listed in Table 4.2.

Table 4.2: Terminology.

Template	**X** are considered a high **Y** fruit
Entity tuple	*(lemons, vitamin c)*
Substitution	**X**=*lemons*, **Y**=*vitamin c*
Sentence	Lemons are considered a high vitamin c fruit.
Paraphrase templates	**X** are rich in **Y** **X** are an excellent source of **Y** **X** are full of **Y**
Paraphrases	Lemons are rich in vitamin c. Lemons are an excellent source of vitamin c. Lemons are full of vitamin c.
Coordinate entity tuples	*(apples, pectin)* *(strawberries, potassium)*

It should be noted that relations are limited to binary relations. In other words, the number of entities in an entity tuple is fixed at 2.

4.3.2 Relationship between Templates and Entity Tuples

We use Figure 4.2 to illustrate an ideal case of the mutual reinforcement between paraphrase templates and coordinate tuples. The input sentential query is "Lemons are rich in vitamin c", which is mapped by the template ***X** are rich in **Y*** and the entity tuple *(lemons, vitamin c)*. Suppose we have already obtained some paraphrase templates and coordinate tuples. They are plotted on the horizontal and vertical axes, respectively. The intersection of two dotted lines in the figure indicates a combination of the corresponding template and tuple. Moreover, a "×" signifies that people use the expression by this combination. For example, the meeting-point of the dotted lines of template ***X** are packed with **Y*** and tuple *(apples, pectin)* represents a possible expression. That is "Apples are packed with pectin". Since there is a "×" attached, we know people use this expression in daily life.

The intensional-extensional representation for a relation suggests the use of suitable tuples to represent the context of a template, and accordingly, the use of suitable templates to represent the context of a tuple. Intuitively, therefore, for template ***X** are packed with **Y***, tuples with "×"s generate its context, such as *(lemons, vitamin c)*, *(dead sea, minerals)* and so forth. On the other hand, for tuple *(apples,pectin)*, templates with "×"s generate its context, such as ***X** are packed with **Y***, ***X** are a rich source of **Y*** and so forth. The distributional hypothesis [22] has been the basis for statistical semantics. It states that words that occur in the same contexts tend to have similar meanings. We are motivated by its extended version:

- If two templates share more common coordinate tuples, they are more likely to be paraphrased to each other.

- If two tuples share more common paraphrase templates, they are more likely to be coordinated to each other.

Thus, paraphrase templates and coordinate tuples are in a mutually reinforcing relationship.

With the aid of "×"s, Figure 4.2 can be divided into two areas: dense and sparse. We use a red oblique line to symbolically separate these two areas. The closer a paraphrase template to the dense area, the better it is, and vice versa. The closer to the left, the better it is as a paraphrase template. Accordingly, the closer to the right, the worse it is as a paraphrase template. A good paraphrase template means it is more semantically similar to the original template. As a result, paraphrase templates that belong to the dense area are regarded as good paraphrase templates for ***X** are rich in **Y***, such as ***X** are high in **Y***, ***X** are an excellent source of **Y*** and ***X** are packed with **Y***. In the same way, we can identify whether templates are paraphrase templates.

4.3.3 Mutual Reinforcement Algorithm

In this section, we introduce the essential feature of our method. We start with a template set T and a tuple set E. The details of how to extract them from the Web according to a sentential query are addressed in Section 4.4. Suppose there are m templates in T and n tuples in E. At the beginning, a bipartite graph is constructed. Let $W^{TE} \in \mathbb{R}^{m \times n}$ denote the transition matrix from T to E and $W^{ET} \in \mathbb{R}^{n \times m}$ the transition matrix from E to T. The meanings of w_{ij}^{te} and w_{ij}^{et} depend on different applications.

We define the following two functions and regard them as the weight of edges between templates and the weight of edges between tuples, respectively.

- $\mathbf{Para}(\mathbf{t_i}, \mathbf{t_j})$: paraphrase degree between two templates t_i and t_j, which returns a value between 0 and 1. A high value will be returned when t_i and t_j are more likely to be paraphrased to each other.

- $\mathbf{Coord}(\mathbf{e_i}, \mathbf{e_j})$: coordinate degree between two tuples e_i and e_j, which returns a value between 0 and 1. A high value will be returned when e_i and e_j are more likely to be coordinated to each other.

There are two different situations when considering the paraphrase degree between t_i and t_j. One is exact equivalence of t_i's and t_j's suitable tuples, such as e_k in Figure 4.3(a). In other words, if we can find many tuples that are shared by two templates t_i and t_j, the paraphrase

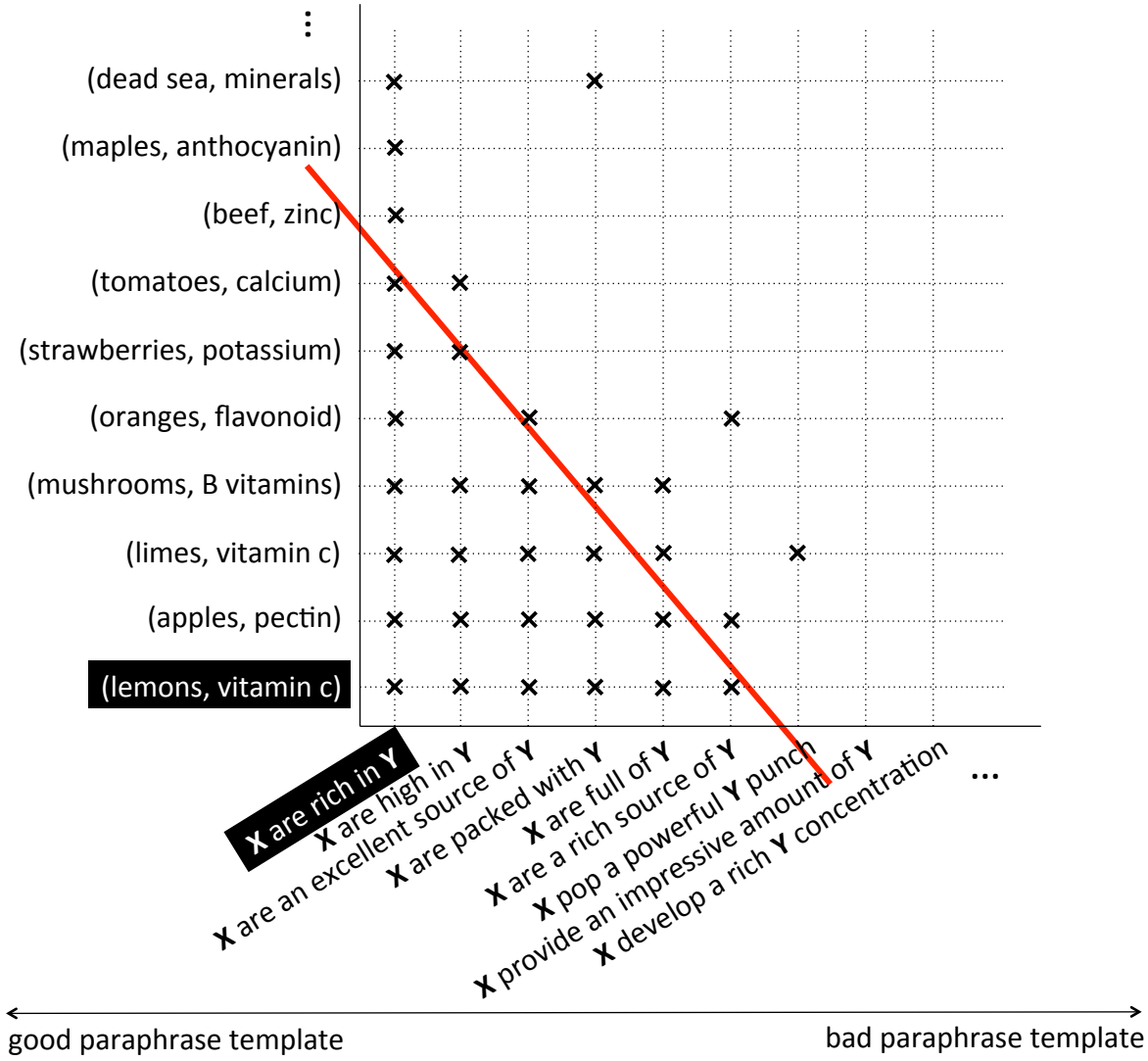


Figure 4.2: Ideal case of the mutual reinforcement between paraphrase templates and coordinate tuples, with the input "Lemons are rich in vitamin c". The mark "×" indicates that people use the expression generated by corresponding template and corresponding tuple.

degree between them is high. This situation has been often considered in previous studies of paraphrase acquisition. To improve retrieval performance, some studies involved taking semantic similarity between terms into account. This motivates us to consider semantic similarity between both templates and tuples. For example, ***X** are rich in **Y*** is semantically similar to ***X** are an excellent source of **Y***, and *(lemons, vitamin c)* is semantically similar to *(apples, pectin)*. Therefore, in addition to the former situation, we also consider the coordinate degree between e_k and e_g, which is $Coord(e_k, e_g)$ shown in Figure 4.3(b). In other words, if we can find many coordinate tuple

pairs that are shared by t_i and t_j, the paraphrase degree between them is high. As a result, the value of $Coord(e_k, e_g)$ is propagated to $Para(t_i, t_j)$ according to the transition probability. Similarly, additional values are propagated from other pairs of coordinate tuples in E to $Para(t_i, t_j)$, then the value of $Para(t_i, t_j)$ is updated. The new value is propagated to $Coord(e_k, e_g)$ in two similar situations. Since the edges between T and E are directional, we use different colors for distinguishing in Figure 4.3. For example, when considering the paraphrase degree between t_i and t_j, we consider both the routes from t_i to t_j (shown in red in Figure 4.3) and the routes from t_j to t_i (shown in blue in Figure 4.3).

Formally, the mutually reinforcing calculations are written as:

$$\begin{aligned} Para(t_i, t_j) = \frac{1}{2}(& \sum_{e_k, e_g \in E} w_{ik}^{te} w_{gj}^{et} Coord(e_k, e_g) \\ & + \sum_{e_k, e_g \in E} w_{jg}^{te} w_{ki}^{et} Coord(e_k, e_g)) \end{aligned} \tag{4.1}$$

$$\begin{aligned} Coord(e_k, e_g) = \frac{1}{2}(& \sum_{t_i, t_j \in T} w_{ki}^{et} w_{jg}^{te} Para(t_i, t_j) \\ & + \sum_{t_i, t_j \in T} w_{gj}^{et} w_{ik}^{te} Para(t_i, t_j)) \end{aligned} \tag{4.2}$$

where $i, j \in [1, m]$ and $k, g \in [1, n]$. When $i = j$, $Para(t_i, t_j) = 1$, which indicates the exactly equal case. Similarly, when $k = g$, $Coord(e_k, e_g) = 1$. After values for all pairs of templates are updated, normalization takes place. This is the same for all pairs of entity tuples. Update continues until the difference between each new value and old value is smaller than a threshold θ. Since the calculations of transition matrices W^{TE} and W^{ET} depend on applications, we discuss the details in Section 4.4.

Finally, as a result, the paraphrase degree between two templates will be high if they share many common tuples, or have many coordinate tuple pairs; the coordinate degree between two entity tuples will be high if they share many common templates, or have many paraphrase template pairs.

4.4 Application: Judgement of Fact Credibility

In this section, we give details of our paraphrasing method when handling two different applications: judgement of fact credibility and QA search, especially how to obtain candidate templates and candidate tuples from the Web (referred to the step 1 in Figure 4.1). Some adjustments are needed to adapt to these two applications.

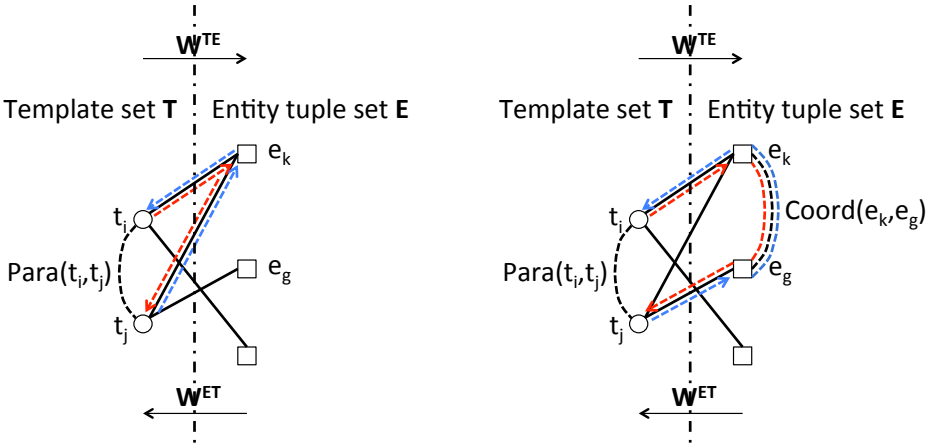


Figure 4.3: Paraphrase degree calculation. Two disjoint sets T and E (vertical chain line is used to separate them) comprise a bipartite graph, where T is composed of candidate paraphrase templates and E of candidate coordinate tuples.

4.4.1 Judgement of Fact Credibility

It is now intuitive to use the Web as a huge encyclopedia and trust information on the Web. However, the information is not always correct or true. For example, Denning et al. [15] reported that information on Wikipedia, which is regarded as the largest online encyclopedia, is not so credible. Therefore, it is necessary to understand risks of obtaining Web information and distinguish credible ones from it. We assume the credibility of a fact is high if people often mention it on the Web. Based on this assumption, a naive way to judge fact credibility is to check its occurrence on the Web. However, this trial always fails. As we stated in Section 5.1, the reason is that although there is a variety of different expressions for a fact, might be difficult to think of these expressions as many as possible. In the most extreme case, we may only think of one expression, which leads to failure of fact credibility judgement. For example, suppose we can only think of a statement like "apples are abundant in pectin", and want to know whether its represented fact is credible. We have known that the fact statement itself is rarely used. If we estimate the credibility of the fact only based on this statement, we would draw a wrong conclusion that apples do not contain much pectin. However, in fact, other fact statements, such as "apples are rich in pectin" or "apples are a great source of pectin", are widely used on the Web. These statements convey the same meaning as the given statement. Therefore, if we also take other statements of the fact into consideration, it is more likely for us to come to the right conclusion that apples are a high pectin fruit. To conclude, we judge fact credibility by observing both the given fact statement and other ones on the Web that convey the same meaning (called *paraphrases* for short hereinafter).

To run our mutual reinforcement algorithm, it is necessary to gather templates and entity tuples in advance. We now briefly introduce how to extract candidate templates and tuples from the Web by using a fact statement "Google has purchased Nest Labs", mapped by the template ***X*** *has purchased* ***Y*** and the entity tuple *(Google, Nest Labs)*, as the input.

4.4.2 Template Extraction

Since there might be many relations between entities and the Web is too large, it is necessary to limit our extraction to a certain field. We use ***context terms*** for this purpose. To obtain context terms, we prepare two kinds of queries. One is a wildcard query generated by the input template, i.e., "* has purchased *". The other is an AND query generated by nouns and verbs (excluded ones such as *be* and *has*) extracted from the given fact statement, i.e., "Google AND purchased AND Nest Labs". The context terms are chosen as the highest *tf·idf* scoring terms in the top search results of these two queries. For example, term "company" is chosen as a context term for the input. Correspondingly, we generate an AND query by the input tuple and the context term, i.e., "Google AND Nest Labs AND company". Candidate templates are extracted from the top N search results of each generated AND query. We heuristically eliminate non-essential phrases, such as additional prepositional phrases (e.g. "Google now owns Nest Labs after shelling out..." is analyzed as "Google now owns Nest Labs..."), or individual tokens, such as adverbs (e.g. "previously announced" is reduced to "announced").

Our template extraction is not limited to the text between two entities. For example, we obtain a candidate template such as ***X*** *announced the* ***Y*** *acquisition back in* *. We also assume an overlong template is more likely to contain additional information, while a too-short template is more likely to miss some information. Both situations lead to non-paraphrases. Therefore, we also exclude overlong and too-short templates.

4.4.3 Entity Tuple Extraction

We first find coordinate terms for both of the entities *Google* and *Nest Labs* using the bi-directional lexico-syntactic pattern-based algorithm [47]. As a result, we obtain *Yahoo* as a coordinate term of *Google*, *Dropcam* as a coordinate term of *Nest Labs*. Substituting each entity by its coordinate terms, we generate wildcard queries for extracting candidate tuples. For example, for *Google*'s coordinate term *Yahoo*, we generate the query "Yahoo has purchased *". For *Nest Labs*'s coordinate term *Dropcam*, we generate the query "* has purchased Dropcam". We then extract entities[4] from the corresponding asterisk part in the top M search results of the above queries. As a result, we obtain tuples such as *(Yahoo, Tumblr)* from the former query, tuple such as *(Nest, Dropcam)* from the latter.

[4]We employ the Stanford part-of-speech tagger to extract nouns or noun phrases.

We use coordinate terms for the following two reasons. First, there is a massive amount of information on the Web. If we only search by "* has purchased *" and extract entity tuples from corresponding portions of sentences, many irrelevant tuples are gathered, such as *(God,freedom)*. Hence, coordinate terms are used to reduce the number of irrelevant tuples. Second, there might be few entity tuples extracted from the Web if the binary relation between two entities is one-to-one. For example, in the sentence "The capital of Japan is Tokyo", the relation between *Japan* and *Tokyo* is one-to-one, since we can only find *Tokyo* as the answer for which city the capital of *Japan* is, and vise versa, we can only find *Japan* as the answer for *Tokyo* is the capital of which country. Thus, it is difficult to obtain other entity tuples from the wildcard query "The capital of * is Tokyo" or "The capital of Japan is *". In this case, coordinate terms are used to increase the number of entity tuples extracted from the Web.

In some cases, we could not obtain enough candidate entity tuples or templates. Hence, several frequently appeared tuples are used to extract more candidate templates. In the same way, several frequently appeared templates are used to extract more candidate entity tuples.

4.4.4 Calculations of transition matrices

Since our objective is to find frequently used paraphrases of the given fact statements, the possibility of combinations of templates and entity tuples is significant. That is, we are concerned about whether people use an expression. For example, even if template ***X** are considered a high **Y** fruit* conveys the same meaning with ***X** pop a powerful **Y** punch*, we do not obtain any search results with the query "apples are considered a high pectin fruit". Hence, such paraphrases are useless for judging fact credibility. Based on the above discussion, transition matrices W^{TE} and W^{ET}(in Section 4.3.3) are calculated in the following respective manners. Entry w_{ij}^{te} is the proportion of e_j's occurrence in t_i's top search results, while entry w_{ij}^{et} is the proportion of t_j's occurrence in e_i's top search results.

4.4.5 QA Search

Question-answering systems are frequently required to handle long queries, especially long natural language queries. Retrieval performance of QA systems can be improved if they can automatically detect the difference between a user's question and existing questions or answers. It is a good way to paraphrase templates in users' questions to widely-used ones in these systems. For example, given template ***X** has purchased **Y*** as the input, our method outputs its paraphrase templates such as ***X** buys **Y*** and ***X** finalizes acquisition of **Y***. Different from judgement of fact credibility, we have no explicit tuple at first. Take template ***X** has purchased **Y*** for further illustration. First, we have to extract some qualified tuples from the Web by a wildcard query generated from the template (e.g. "* has purchased *"). Most frequently appearing tuples are

chosen as qualified tuples. For example, we have *(Yahoo, Tumblr)* and *(CGGVeritas, Petrodata)*. These tuples are used to extract candidate templates in the same manner we mentioned in Section 4.4.2. For each tuple, we then extract candidate coordinate tuples by using the method introduced in Section 4.4.3.

Since our objective is to find frequently used paraphrase templates, we are not concerned with a certain entity tuple. For example, we know template ***X** are considered a high **Y** fruit* conveys the same meaning with ***X** pop a powerful **Y** punch*. Even though the former template is useless in retrieving *(apples,pectin)*, it is effective in finding tuples such as *(raspberries,fiber)*. Therefore, transition matrices W^{TE} and W^{ET}(in Section 4.3.3) are calculated in the following respective manners. Entry w_{ij}^{te} is determined by how many suitable tuples it has, while entry w_{ij}^{et} is determined by how many suitable templates it has.

4.5 Evaluation

In this section, we discuss the experiments we conducted to validate the main claims of the paper, which is an enhancement of what we did in [71].

4.5.1 Experimental Setting

Given a sentential query, it is costly to find all templates and all entity tuples throughout the entire Web. For our experiments, we extracted candidate templates from the top $N = 1000$ (mentioned in Section 4.4.2) search results of each AND query formed by a tuple and an additional context term, using the Bing Search API[5] . We extracted candidate tuples from the top $M = 1000$ (mentioned in Section 4.4.3) search results of each wildcard query. We fixed the value of threshold θ (mentioned in Section 4.3.3) to 0.0001 and found values of $Para(t_i, t_j)$ and $Coord(e_k, e_g)$ converging after $20 \sim 25$ updates. Note that we empirically set the above values to N, M and θ, respectively.

4.5.2 Performance for Fact Credibility Judgement

Query Data

To our knowledge, there is few widely accepted public dataset for paraphrase acquisition at sentence level. Correspondingly, it is difficult to directly use query data from evaluation part of any previous work. Therefore, we manually create our query data, containing 120 sentential queries, for evaluation. Since our proposed method, the mutual reinforcement algorithm, is based on the intensional-extensional representation for a relation, these 120 sentential queries are actually from the following six semantic relations:

1. **highConcentration:** *We define this as a food contains a high amount of a certain nutrient.*

[5]http://datamarket.azure.com/dataset/bing/search

2. **acquisition:** *We define this as the activity between two companies such that one company acquired another.*

3. **field:** *We define this as the relation between a person and his field of expertise.*

4. **majorLanguage:** *We define this as the relation between a language and an area such that the language is the major one in the area.*

5. **manufacture:** *We define this as the relation between a product and its manufacturer.*

6. **produce:** *We define this as the relation between a mineral and its producing area.*

We select last five relations by referring to some previous works [9][10][27] about acquiring paraphrases or detecting paraphrases in a corpus, and questions from TREC-8 Question-Answering Track. In addition to these five relations, we added the **highConcentration** relation, since we believed that it is an important relation and many queries in this relation cannot be broken down into joint phrase queries.

As we mentioned in Section 4.2.1, a sentence can be mapped by a template and an entity tuple. We list all templates and all entity tuples used to generate our sentential queries in Table 4.3, grouped by their semantic relations. For each relation, there are 5 templates and 4 entity tuples. Consequently, there are 20 combinations between templates and entity tuples. Thus, we have 20 sentential queries in each relation. Since each of them can be regarded as a "fact statement", we also analyze the performance for fact credibility judgement in our evaluation. Actually, for entity tuples, we manually selected 2 and also manually created 2 incredible tuples for each relation. Here, an incredible entity tuple indicates one that there does not exist the certain relation between entities in this tuple. Take the **highConcentration** relation for example. *(avocados, pectin)*, *(strawberries, protein)* are incredible tuples, since avocados do not contain much pectin and strawberries are not full of protein. Therefore, among 120 fact statements, there are 60 credible ones and 60 incredible ones. Besides, we also check the occurrence of each fact statement on the Web by Web search. We find there are 47 queries commonly used and correspondingly, 73 queries seldom used on the Web. Here, if the occurrence of a query on the Web is more than 10, we regard it as a commonly-used query, and vice versa.

Performance of Paraphrase Acquisition

As there is not much work in acquiring sentence-level paraphrases from the Web, it is difficult to directly compare against existing methods. Therefore, we constructed a baseline method for comparison, a variation of method stated in [9]. In the baseline method, we regard tuples as the context of each template, and use them to construct vector for each template. Then calculating

Table 4.3: Sentential queries for evaluation. Entity tuples shown in bold are credible ones, while the rest are incredible ones.

Template	Relation	Entity Tuple
X are rich in **Y**	**highConcentration**	***(lemons, vitamin c)***
X are a great source of **Y**		***(apples, pectin)***
X are packed with **Y**		*(avocados, pectin)*
X are abundant in **Y**		*(strawberries, protein)*
X are considered a high **Y** fruit		
X has purchased **Y**	**acquisition**	***(Google, Nest Labs)***
X bought **Y**		***(Facebook, WhatsApp)***
X has agreed to acquire **Y**		*(Twitter, Dropbox)*
X has announced plans to buy **Y**		*(Microsoft, Instagram)*
X finished its acquisition of **Y**		
X revolutionized **Y**	**field**	***(Albert Einstein, physics)***
X is popularly known as the father of **Y**		***(Euclid, geometry)***
X is known for his work in **Y**		*(Nietzsche, philosopher)*
X laid much of the foundation for **Y**		*(James Waston, biology)*
X made enormous advances in **Y**		
X is the major language of **Y**	**majorLanguage**	***(Cantonese, Hong Kong)***
X is widely spoken in **Y**		***(French, France)***
X is a **Y**-speaking city		*(French, Spain)*
X is **Y**'s official language		*(English, Taiwan)*
X is the official language of **Y**		
X manufactures **Y**	**manufacture**	***(Toyota, Lexus)***
X is planning to release **Y**		***(Nissan, Infiniti)***
Y was created by **X**		*(Honda, Vovle)*
Y is the luxury brand of **X**		*(Toyota, Mini Cooper)*
Y is a segment of **X**		
X is the biggest producer of **Y**	**produce**	***(China, tungsten)***
X dominates the primary production of **Y**		***(Russia, oil)***
X is the largest supplier of **Y**		*(Russia, aluminium)*
X has the highest **Y** reserves		*(Canada, natural gas)*
X dominates the global **Y** market		

Table 4.4: Performance of paraphrase acquisition.

Method	Fact Statement Type	# Obtained	# Correct	Precision[6]
Simple	**All**	13.4	8.4	0.464
	Widely-used	31.2	20.2	0.499
	Seldom-used	4.5	2.5	0.447
Template Reused	**All**	24.3	15.7	0.668
	Widely-used	23.7	17	0.730
	Seldom-used	24.8	14.5	0.614
Tuple Reused	**All**	28.1	19.1	0.688
	Widely-used	35.8	26.2	0.746
	Seldom-used	23.1	14.4	0.649
Complete	**All**	29.1	20.4	0.712
	Widely-used	35.6	26.2	0.751
	Seldom-used	22.9	15.1	0.680
Baseline	**All**	11.4	6.8	0.417
	Widely-used	22.4	14.9	0.498
	Seldom-used	6.2	3.0	0.380

the paraphrase degree between templates turns to be a problem to compute the cosine similarity between two vectors.

Since we mentioned in Section 4.2.2 that frequently appeared tuples are used to extract more candidate templates, and vice versa, frequently appeared templates are used to extract more candidate tuples, we investigate the employment of tuples and templates. As a result, we have 4 ways to obtain candidates: (1) Simple: extracted tuples and templates are not further used; (2) Template reused: extracted templates are further used to extract more candidate tuples; (3) Tuple reused: extracted tuples are further used to extract more candidate templates; (4) Complete: both extracted templates and extracted tuples are further used to extract more candidate tuples and candidate templates, respectively.

Table 4.4 shows the performance of our method for paraphrase acquisition, compared with the baseline method. Here we calculated the precision as how many "correct" paraphrases are in the paraphrases obtained by our method. From the table, we can point out that the employment of extracted templates and extracted tuples can make a big increase in precision, about 24.8% increased when compared the complete method with the simple method. Moreover, we can see that our complete method obtains a precision of 71.2% over all fact statements, compared to 41.7% with the baseline. Furthermore, for frequently appearing statements, our complete method gets a precision of 75.1%, compared to 49.8% with the baseline. While for infrequent ones, we also get a good result, about 68%, compared to 38% with the baseline. Besides, compared to the

Table 4.5: A comparison between baseline and our method for paraphrase acquisition.

System	Precision@5	Precision@10	Precision@15	Precision@20	Precision	Relative Recall
Baseline	0.451	0.700	0.619	0.660	0.417	0.315
Complete	0.813	0.793	0.756	0.738	0.712	0.932

baseline, our method makes a significant growth in obtaining correct paraphrases, nearly 3 times.

Take as an example the sentential query generated by the template **X** is popularly known as the father of modern **Y** and the entity tuple *(Albert Einstein, physics)*. 5 "correct" paraphrases are shown as below:

- *Albert Einstein is known as the father of physics.*
- *Albert Einstein is a prominent and legendary man acknowledged for his astounding contribution to physics.*
- *Albert Einstein is held up as a rare genius,who changed the field of theoretical physics.*
- *Albert Einstein fundamentally changed the world-view of physics.*
- *Albert Einstein laid much of the foundation for physics.*

Table 4.5 shows a comparison between the baseline and our complete method for paraphrase acquisition. Our method outperforms the baseline method no matter what value k is. Besides, significant improvement is achieved when k equals to 5. On the other hand, it is difficult to estimate the recall since we do not have a complete set of paraphrases for a certain fact statement. However, following Tague-Sutcliffe [61], we can pool the correct results (corresponding here to correct paraphrases) of each method to form the answer set. The relative recall can be calculated using the following formula:

$$\text{Relative recall} = \frac{\text{\# of correct paraphrases obtained by a method}}{\text{\# of correct paraphrases obtained by our method and baseline}}$$

From Table 4.5, we can know that our method achieved a big increase in relative recall, which indicates that it is effective to incorporate mutual reinforcement between templates and tuples to identify paraphrases. We also did a *t-test* between precision for our method and the baseline method, which yielded a p-value of 0.00281. As the p-value is smaller than 0.05, we can infer that the experimental results of our method is reliable and there is a statistically significant difference between baseline and our method.

Table 4.6: Performance for judging fact credibility by using top 10 paraphrases.

Fact Statement Type	Average HitCount of Input Statements	Average Increase HitCount	Average Increase Rate
Widely-used	7318.00	14709.18	2.01
Seldom-used	0.86	9831.75	11432.27
Credible	4212.35	22030.59	5.23
Non-credible	0	0.55	–

Table 4.1 lists the top 15 paraphrases of "Lemons are considered a high vitamin c fruit" generated by our method. By observing these top results, we find sentences, such as "lemons contain vitamin c" are misjudged as paraphrases. Such sentences are not paraphrases because we can not infer that lemons have a high amount of vitamin c from them. Our method fails to identify such sentences because templates from such sentences, we call them "general" templates, are more likely to have many suitable entity tuples. Therefore, these templates may share many common tuples with the original template, which leads them obtain high scores of paraphrase degree.

Interestingly, we find that sometimes fact statements that have the opposite meaning are misjudged as paraphrases. For example, for the fact statement "China dominates the global tungsten market", we obtain "China was the world's leading tungsten consumer" in the top 3. Therefore, how to identify the opposite or negative meaning of the original statement is a problem that needs further consideration.

Judging Fact Credibility

A fact has a variety of different expressions in the surface form, namely fact statements. For example, all the following statements represent the same fact:

- *Apples are rich in pectin.*
- *Apples are a great source of pectin.*
- *Apples are packed with pectin.*

When we say a fact statement is credible, it basically refers to its represented fact is credible. We assume that a fact is credible if people often mention it on the Web. Hence, when we judge the credibility of a fact, in the ideal case, we should consider all its possible expressions in the surface form. However, since it is difficult to obtain all possible statements, we believe part of fact statements is enough for fact credibility judgement.

Take the fact statement "apples pop a powerful pectin punch" for example. As we already know, this statement never appears on the Web. If we estimate the credibility of the fact only based on the given statement, we would draw an erroneous conclusion that apples do not have a high amount of pectin. However, if we also take other statements of the fact into consideration, the ones that convey the same meaning as the given statement, such as "apples contain a lot of pectin" or "apples are rich in pectin" which are widely used on the Web, we could draw the right conclusion that apples are a high pectin fruit.

As a result, when we judge whether a fact is credible or not, we actually check how many times its fact statements appear on the Web. Accordingly, the hitcount on the Web could be an indicator for fact credibility judgement. Besides, if a fact is credible, all its statements are credible correspondingly.

Table 4.6 shows the performance for fact credibility judgement. For each fact statement, viz. each sentential query, we also take into consideration the top 10 paraphrases obtained by our method. It means we not only estimate the hitcount of each fact statement on the Web, but also aggregate the hitcount of its paraphrases, and make a comparison between them.

The first column of Table 4.6 indicates the classification of fact statements. The second column represents the average number of how many times each fact statement occurs on the Web. The third column is the comparison of absolute value: how the occurrence increased by considering paraphrases, while the forth column is a relative value: the average increase rate.

From this table, we know that with the help of paraphrases, we can retrieve more Web pages. Especially, for fact statements that are not frequently used by people, we got a tremendous increase of retrieved Web pages, 11432.27 times increased. This finding also illustrates the effectiveness of paraphrased queries, since they solve the problem caused by the expression rarity of the original sentential queries. It is much more likely to obtain desired information, because we are able to find much more Web pages hit these paraphrased queries. When considering fact credibility judgement, we found for non-credible fact statements, even we extend them by their paraphrases, there are few Web pages returned, which results in a small increase in hitcount, only 0.55 on average. Hence, if we consider part of statements of a fact, but still cannot find enough appearances, we can figure out that the fact is incredible and its statements are correspondingly incredible. As a result, paraphrases are effective for estimating fact credibility.

Case Study

In this section, two cases are presented to demonstrate the effectiveness of paraphrases for fact credibility judgement. In each case, we take one fact statement used in our evaluation for example. Notice that all the information about both the fact and its statements are based on the Bing search engine at the time of writing this paper.

Case 1 Seldom-used & Credible

The fact statement "Facebook has announced plans to buy WhatsApp" never appears on the Web. However, its represented fact is true since Facebook did buy WhatsApp in 2014. The result of our experiment can corroborate this. Specifically, in the beginning, the hitcount of the statement is 0. After we took into consideration the top 10 paraphrases obtained by our method, such as "Facebook has announced that it is acquiring WhatsApp", "Facebook has agreed to buy WhatsApp", or "Facebook will acquire WhatsApp", we found 3600 more search results in total. As the number is not small, we can conclude that there is an acquisition between Facebook and WhatsApp, and Facebook is the buyer. Therefore, the fact statement is credible since we have already known its represented fact is credible.

Case 2 Seldom-used & Incredible

We cannot find the fact statement "Avocados are packed with pectin" on the Web. Besides, we looked up many other materials, such as Wikipedia, websites of food nutrition. We found there is little pectin contained in avocados. The result of our experiment can support this finding. In more details, in the beginning, the hitcount of the statement is 0. After we took into consideration the top 10 paraphrases obtained by our method, such as "Avocados contain high levels of pectin", "Avocados are a rich source of pectin", or "Avocados are an excellent source of pectin", we found 4 more search results in total. However, since the total occurrence is still very small, we can draw a conclusion that avocados contain little pectin. Therefore, the fact statement is incredible since we have already known its represented fact is incredible.

4.5.3 Performance for QA Search

Performance of Paraphrase Acquisition

We performed an evaluation of our method over 20 questions in the TREC-8 Quesion-Answering Track, listed in Table 4.7. We use Q_{id} to represent a question in the QA data. The second column of the table shows the templates that we extracted from the TREC-8 questions. The third column presents the precision of paraphrases for each question.

The average precision of our method is 46.9%, while the average precision for the baseline is 39.9%. Q_{35} obtains the best performance with a precision of 76.5%, while we could not acquire any paraphrases for Q_5 and Q_{23}. For Q_5, although the template itself is used by people on the Web, it is a unique template and we found they all come from the question What is the name of the managing director of Apricot Computer?, which is Q_5 itself. Hence few tuples can be extracted from the template. For Q_{23}, we found the template "X and Y started ambassadorial

Table 4.7: Performance of paraphrase acquisition for 20 questions in TREC-8.

Q_{id}	Question Template	Presicion
Q_1	**X** is the author of **Y**	0.708
Q_3	**X** manufactures **Y**	0.571
Q_4	**X** spend * on **Y**	0.429
Q_5	**X** is the name of the managing director of **Y**	0
Q_{11}	**X**'s wife is **Y**	0.368
Q_{12}	**X** spend * on **Y**	0.429
Q_{14}	**X** is the biggest producer of **Y**	0.559
Q_{18}	**X** is the first **Y** president	0.471
Q_{21}	**X** was the first **Y** in	0.625
Q_{23}	**X** and **Y** started ambassadorial relations in	0
Q_{24}	**X** visited **Y** in	0.542
Q_{25}	**X** was the lead actress in the movie **Y**	0.308
Q_{28}	**X** is best known for **Y**	0.621
Q_{29}	**X** is the brightest star visible from **Y**	0.353
Q_{32}	**X** received **Y** in	0.455
Q_{33}	**X** is the largest city in **Y**	0.667
Q_{34}	**X** is buried **Y**	0.417
Q_{35}	**X** is the highest mountain in **Y**	0.765
Q_{38}	**X** was born in **Y**	0.524
Q_{40}	**X** won **Y** in	0.615
	Average Precision	0.469
	Average Precision of baseline	0.399

relations in" is never used on the Web. As a result, no tuple can be extracted, correspondingly, no candidate templates can be extracted.

QA Search

A bottleneck of QA search is the difference between a user's question and existing questions or answers. Retrieval performance of QA search can be improved if such difference can be

Table 4.8: Performance for QA search by using top 10 paraphrases.

# Average Answers	# Average Increased Answers	Average Increase Rate
18.571	7.714	0.767

automatically detected.

Table 4.8 lists the average experimental result of TREC-8 questions by using top 10 paraphrases obtained by our method. The first column is the average number of answers found in top 100 search results of each question (still using Bing Web search). The second column is the average number of increased answers found in top 10 search results of each paraphrase[7]. The third column is the average increase rate of each question.

From the table, we can see that with the help of paraphrases, 7.714 more answers can be found within very top search results. And we got an average increase rate of 76.7%. This suggests that paraphrases are useful to find more answers in QA search.

4.6 Summary

We handle with sentential queries and aim at improving its retrieval performance. Different from previous studies, we argue that separate terms or phrases from a sentential query may lead to the missing of some information or query drift. To avoid such problems, we propose query paraphrasing for sentential queries. In sentential query paraphrasing, we use other frequently used queries that convey the same meaning to avoid returning no answers. Furthermore, we incorporate coordinate relationships between entity tuples and take a mutually reinforcing approach to identify paraphrase templates. The experimental results show that our method can acquire more paraphrases from the Web. Besides, with the help of paraphrases, more Web pages can be retrieved, especially for those sentential queries that could not find any answers with its original expression.

[7]Note that we took top 10 paraphrases in this experiment.

CHAPTER 5

STRUCTURING SEARCH RESULTS BASED ON COORDINATE RELATIONSHIPS

5.1 Introduction

Nowadays, search engines, such as Google[1] and Bing[2] play a significant role in people's daily lives. People can obtain a variety of information via search engines. However, in general, mainstream search engines return a simple, flat list. This leads to difficulties in quickly browsing and understanding search results because intrinsic connections between these search results are rarely presented and are left to the users to determine. One alternative approach is to cluster Web search results in thematic groups. Hearst and Pedersen [24] first introduced the clustering of Web search results in their Scatter-Gather system. Subsequently, multiple studies have concentrated on developing a method to organize Web search results into clusters [38][49][50][69][70]. However, results from state-of-the-art clustering methods can deviate from those derived via human cognition because human cognition is based on knowledge and experience from the real world while clustering methods are based on similarity measures.

Take two different search services for example. One is *news search*, which provides important access to the Web versions of newspapers, magazines, and news wires. Suppose a user wants to know about recent school shootings and therefore enters the keyword query "school shooting". Figure 5.1 shows the search results returned by Google News[3] (at the time of writing the document). From these results, we can see that similar news articles, such as the ones about the plans to fight gun violence or the shooting massacre threat on the Facebook, are all aggregated

[1] http://www.google.com
[2] http://www.bing.com
[3] https://news.google.com

together; however, the presentation of the news articles is a simple list. This means that the search engine mixes different events related to school shootings. Even though users can scan the list from top to bottom until they find the information they are looking for, it is difficult to pick a certain event, e.g., "Newtown school shooting". Moreover, it is difficult to survey the entire topic "school shooting". This also happens when querying on Yahoo News[4]. Moreover, Bing News[5] could not find any results for "school shooting". Another example is *academic paper search*, which provides a way for users to access academic papers indexed on the Web. Suppose one wants to survey papers on search result clustering, and correspondingly, issues a keyword query "search results clustering". We find again a similar situation. The presentation of papers is in a simple, flat list. Connections or relationships between the papers are not shown. For example, paper A is a rival of paper B. Therefore, it can be difficult to survey a research area by scanning the paper list. From the above discussion, we can see that the major problem is unstructured search results.

Previous studies have tackled this problem by applying the clustering method. However, as we mentioned before, there is a gap between clustering results and human cognition. Let us investigate the following two news articles about "school shooting" and take them as an example. They are titled "Oregon School Shooting: 1 Student Dead, Suspect Also Killed" and "Second Suspect Pleads Guilty in Frederick High School Shooting", respectively.

Oregon School Shooting: 1 Student Dead, Suspect Also Killed
At a press conference following the Oregon school shooting this morning, police say one student was gunned down at Reynolds High School in Troutdale, Ore. According to USA Today, officials later confirmed that the shooter was also deceased. At this point...

Second Suspect Pleads Guilty in Frederick High
School Shooting
The second of two men charged in a shooting that wounded two teenage boys outside Frederick High School at a basketball game pleaded guilty for his role in the incident Tuesday. Brandon Earl Tyler, 22, pleaded guilty to two counts of first-degree assault.

These two news articles are likely to be grouped in the same cluster because both of them describe injuries and deaths during shootings. However, for human beings, it is natural and intuitive to categorize these two news articles as different events. One is "Oregon School Shooting", and the other is "Frederick High School Shooting".

Therefore, in this paper, our objective is to structure the search results of a user-given query

[4]http://news.yahoo.com
[5]https://www.bing.com/news

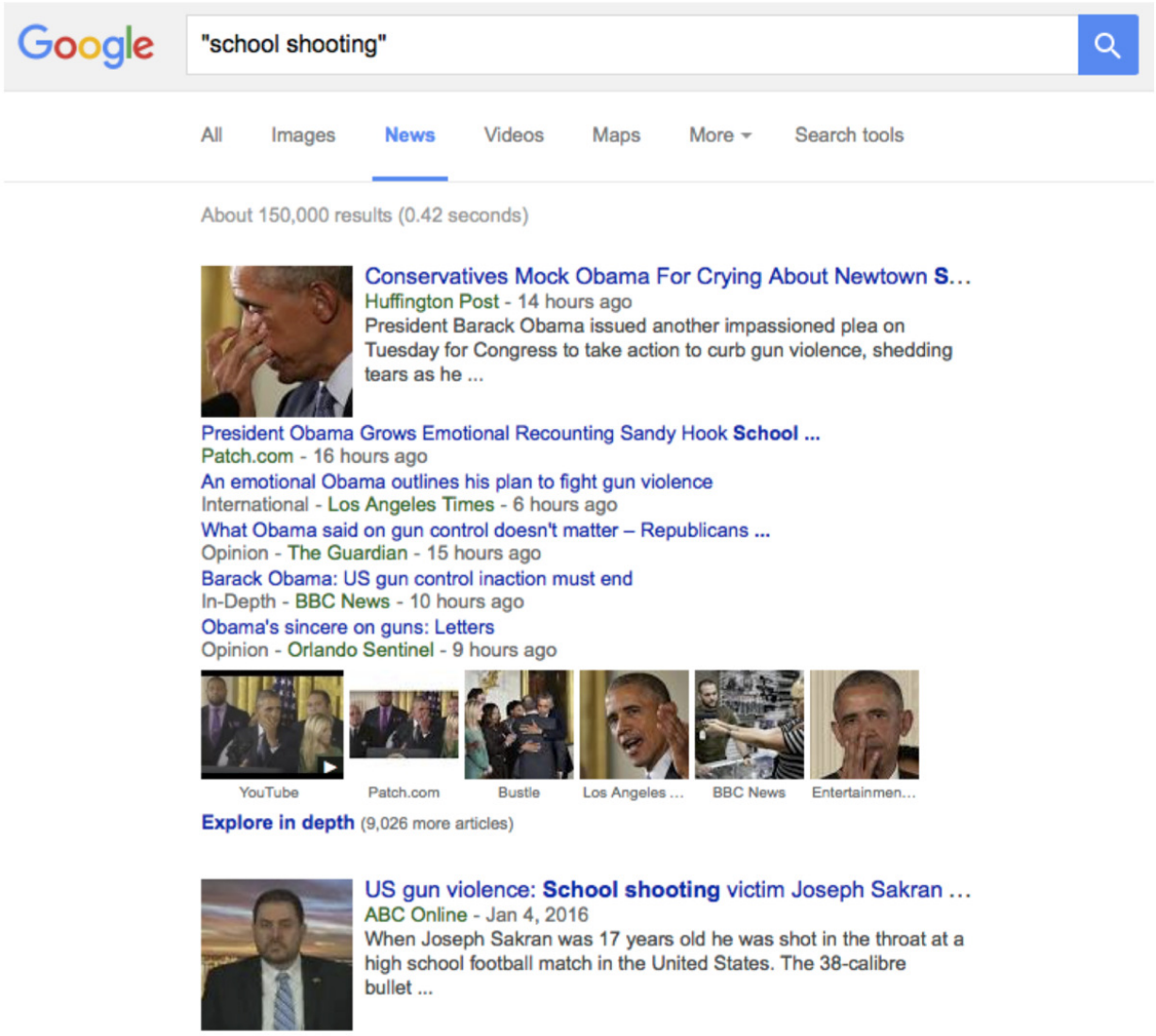


Figure 5.1: Search results for “school shooting” from Google News.

in such a way that it is intellectually easier for users to understand the aggregation result. To accomplish this goal, we introduce “coordinate relationship” between pairs of documents. In brief, two documents are coordinate to each other if they talk about the same topic and they describe similar but different events or concepts. Because they are mutually exclusive in semantics, they should not be grouped into the same cluster. In contrast, documents describing the same event should be grouped into the same cluster. Based on these two criteria, documents are clustered in a manner closer to human cognition. Further, with the help of coordinate documents, it is easier to understand the intrinsic connections between individual documents.

The remainder of the paper is organized as follows. We formalize the problem in Section 5.2. In Sections 5.3 and 5.4, we describe the details of our proposed method. In Section 5.5, we show

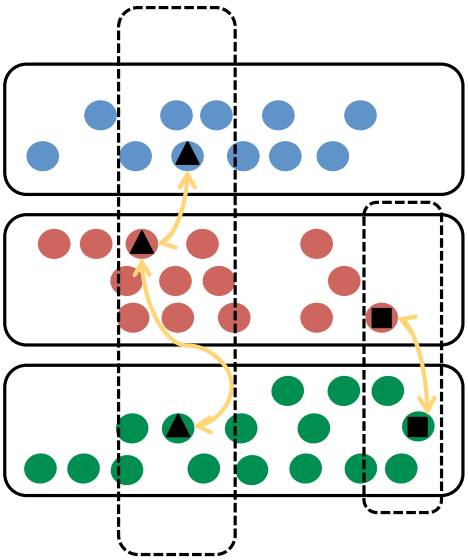


Figure 5.2: Example of the output.

experimental results which were conducted on the New York Times Annotated Corpus. Finally, we conclude the paper in Section 6.7.

5.2 Problem Statement

We can define the problem in this study as follows. Given a set of documents $D = \{d_1, ..., d_N\}$, we want to compute an assignment $C = \{C_i | C_i$ is a subset of D and $C_i \cap C_j = \phi\}$, such that C_i is a thematic group, and obtain intrinsic relations between the documents $CD = \{(d_i, d_j) | d_i$ and d_j are coordinate documents of each other$\}$.

In other words, given the input of a set of documents about a certain topic, the output contains two parts: the partition of these documents and their intrinsic relationships. In particular, we focus on the relationships that are mutually exclusive, referred to as the ***coordinate*** hereafter.

Figure 5.2 gives an example of our output. Each circle represents a document and circles with the same color in a solid frame consist of a thematic group. Circles with the same marker inside represent their represented documents describe similar but different events or concepts, so-called coordinate documents, e.g., news articles stating the occurrence of different school shootings. We use the dash frame to represent the area of coordinate documents across the thematic groups. Pairs of circles connected by the yellow arrows indicate the pairs of coordinate documents. Therefore, the output is the thematic groups, indicated by the solid frames, and coordinate document pairs, indicated by the yellow arrows in Figure 5.2.

It is intuitive to think that documents that hold a coordinate relationship should not be grouped together. Wagstaff et al. introduced the concept ***cannot link*** in [62] as two instances must not be

placed in the same cluster. Following their definition, we can infer that a pair of documents that is coordinate to each other form a cannot link. In contrast, there are relationships that are mutually inclusive and documents that hold such relationships should be grouped together. We borrow the naming in [62] and call such pairs of documents ***must links***. On the basis of these two types of constraints, documents are clustered in a manner closer to that of human cognition. For example, news articles are organized according to events they describe.

Based on the discussion, we divide the problem into three sub-problems.

1. Coordinate Document Detection

Given a set of documents $D = \{d_1, ..., d_N\}$, we want to detect coordinate relationships between the documents $CD = \{(d_i, d_j)|d_i \text{ and } d_j \text{ are coordinate documents of each other}\}$.

2. Must Link Detection

Given a set of documents $D = \{d_1, ..., d_N\}$, we want to detect mutually inclusive pairs in the documents $ML = \{(d_i, d_j)|d_i \text{ and } d_j \text{ are mutually inclusive}\}$.

3. Constrained Clustering

Given (i) a set of documents $D = \{d_1, ..., d_N\}$, (ii) a set of cannot links $CL = \{(d_i, d_j)\}$, and (iii) a set of must links $ML = \{(d_i, d_j)\}$, we want to compute an assignment $C = \{C_i|C_i \text{ is a subset of } D \text{ and } C_i \cap C_j = \phi\}$, such that C_i is a thematic group.

5.3 Coordinate Documents

In this section, we address the key concept "coordinate relationship" and discuss it at the document level.

Coordinate relationships exist at different levels, such as term, sentence and document levels. Here, we use the symbol "$\|$" to express the coordinate relationship between elements. Previous studies [43][47][60] concentrate on coordinate relationship at the term level and define it between terms as below:

$$t \parallel t' \Leftarrow \exists C(t \text{ belongs to } C \wedge t' \text{ belongs to } C)$$

where C denotes a concept. In other words, two terms are coordinate to each other if they share any common hypernym. For example, because both "Umpqua Community College" and "Virginia Polytechnic Institute and State University" belong to the *school* category, they are coordinate terms. Here, "school" is their common hypernym.

When considering coordinate relationship at the document level, we assume that a document, especially a news article, is a combination of subjects and actions, which are presented by nouns (to be specific, proper nouns) and verbs, respectively. Based on this assumption, it is intuitive to consider two kinds of "similar but different" documents: (1) documents with similar subjects but

different actions; (2) documents with similar actions but different subjects. Note that documents should be under the same topic in both of the two situations. We know that with the development of a news event, its subjects change a little. In contrast, its actions change a lot. Therefore, in the former situation, a document is a follow-up[6] (or a followed-up) of another document. On the other hand, we know that similar news events have similar developments. We assume that actions can locate the development phase of an event. Therefore, in the latter situation, similar but different documents denote documents stating different events but in the same development phase.

Article 1

The Oregon shooting ***occurred*** *on October 1, 2015 at the UCC campus near Roseburg, Oregon, United States. Christopher Harper-Mercer, a 26-year-old enrolled at the school, fatally* ***shot*** *an assistant professor and eight students in a classroom. Seven to nine others were* ***injured****. After being wounded by two police officers, the gunman* ***committed suicide*** *by shooting himself in the head.*

Article 2

Ten people were killed when a gunman opened fire at Oregon's Umpqua Community College on Thursday, forcing the nation to face yet another mass shooting. Seven other people were injured, and the shooter is dead. Earlier estimates had put the number of people hurt much higher. Multiple law enforcement officials familiar with the investigation identified the gunman as 26-year-old Christopher Harper-Mercer.

Article 3

The Virginia Tech shooting ***occurred*** *on April 16, 2007, on the campus of Virginia Polytechnic Institute and State University in Blacksburg, Virginia, United States. Seung-Hui Cho, a senior at Virginia Tech,* ***shot and killed*** *32 people and* ***wounded*** *17 others in two separate attacks, approximately two hours apart, before* ***committing suicide****.*

Given Article 1, which describes the occurrence of *the Oregon school shooting*, its similar but different documents can be the follow-ups of this article, e.g., the launch of a campus-wide search for explosives, more executive action on the subject of gun control, corresponding to the first situation in the above discussion. Its similar but different documents can be also the articles stating the occurrence of other school shooting events, such as Article 3 about *the Virginia Tech shooting*, corresponding to the second situation in the above discussion. In this paper, we target at the latter one.

[6]Follow-up is an article giving further information on a previously reported news event.

By carefully observing Article 1 and Article 3, we find that (1) the subjects are coordinate to each other in some way. For example, the occurrence time of the Oregon shooting, *October 1, 2015*, is coordinate to the occurrence time of the Virginia Tech shooting, *April 16, 2007*. The occurrence location of the Oregon shooting, *UCC campus*, is coordinate to that of the Virginia Tech shooting, *campus of Virginia Polytechnic Institute and State University*. (2) The actions in both these articles are similar (see terms or phrases in bold). For example, at the beginning of both these articles, it states the occurrence of the event, using exactly the same term "occurred". When it comes to the injuries and deaths of each shootings, Article 1 used the term "injured", while Article 3 used the term "wounded". Even though these two terms have different surface forms, they have the same semantic meaning.

Based on the finding, we assume that two documents are coordinate to each other if they have coordinate subjects and similar actions. Therefore, we give the definition of coordinate relationship between documents in this paper as follows:

$$\begin{aligned} d \parallel d' \Leftarrow \exists C(&d \text{ belongs to } C \wedge d' \text{ belongs to } C \wedge \\ &Subject(d) \parallel Subject(d') \wedge \\ &Action(d) \coloneqq Action(d')) \end{aligned}$$

where C denotes a topic. $Subject(d)$ and $Action(d)$ represent the subject set and the action set of d, respectively.

5.3.1 Finding Coordinate Documents

As we mentioned above, in this paper, we target at documents with similar actions but different subjects, viz. articles describing the same development phase but different events. We can tackle this problem from these two aspects. We define two functions as follows.

$CoordSub(d_i, d_j)$

Given two documents d_i and d_j, this function returns the coordinate subject degree between them, which is a value between $[0, 1]$. Note that when $i = j$, its value is 0.

$SimAct(d_i, d_j)$

Given two documents d_i and d_j, this function returns the similar action degree between them, which is a value between $[0, 1]$.

Based on the assumption discussed above, we can combine the coordinate subject degree and the similar action degree via a weighted harmonic mean to compute the coordinate degree between

the two documents. This formula is given below.

$$Coord(d_i, d_j) = \frac{1}{\alpha \cdot \frac{1}{CoordSub(d_i, d_j)} + (1-\alpha) \cdot \frac{1}{SimAct(d_i, d_j)}} \tag{5.1}$$

where α is the weight. When $\alpha = 1$, only the subjects are considered when calculating the coordinate score; and when $\alpha = 0$, only the actions are considered.

We know that the problem turns to be two sub-problems: a comparison between subject sets and a comparison between action sets. Because subjects are presented by proper nouns and actions are presented by verbs, the two sub-problems are:

1). Given two proper noun sets N_i and N_j that are extracted from d_i and d_j, respectively, we want to compute the coordinate degree between these two sets.

2). Given two verb sets V_i and V_j that are extracted from d_i and d_j, respectively, we want to compute the similarity between these two sets.

To make a comparison between sets, we first introduce the comparison between terms.

5.3.2 Term Comparison

Here, we use two methods to compare nouns based on their coordinate relationship and verbs based on their similarity relationship. $IsCoord(n_i, n_j)$ denotes the coordinate degree between the two nouns. $IsSynonym(v_i, v_j)$ denotes the similarity degree between the two verbs. The details are addressed as below.

The WordNet method

WordNet[43] is a large English lexical database. It groups English words into sets of synonyms called synsets, and records a number of relations between these synonym sets or their members. For example, noun synsets are arranged into hierarchies that indicate the super-subordinate relation (also called hyperonymy or hyponymy). Therefore, we know the hypernyms of a term via WordNet.

As introduced in Section 5.3, two terms are coordinate if they share any common hypernyms. Therefore, given two nouns n_i and n_j, we can look up their hypernyms in WordNet and check whether they have any in common. The result is a boolean value. Therefore, $IsCoord(n_i, n_j)$ returns a value of 0 or 1.

Because WordNet groups words into synsets that have similar meaning, given two verbs v_i and v_j, we can look up them in WordNet and check whether they are in the same synset. The result is a boolean value. Therefore, $IsSynonym(v_i, v_j)$ returns a value of 0 or 1.

The word2vec method

Mikolov et al. introduced word2vec, a group of related models, such as the skip-gram model and the continuous bag of word (CBOW) model, to learn vector representations of words from a text corpus [41][42]. The learned word vectors can satisfactorily capture relationships between words in semantics.

In addition to the definition of the coordinate terms introduced in Section 5.3, we also consider that, if two terms are coordinate to each other, they should appear in the same, or similar contexts. Because word2vec learns the distributed representation of words by considering the surrounding words of each word, we use the cosine similarity of learned vectors to represent the coordinate degree between words. However, synonyms have high similarities. Therefore, to distinguish coordinate terms from synonyms, we empirically set a threshold θ ($\theta = 0.9$ in the experiments) and treat words whose similarity is greater than the threshold as synonyms. In this case, $IsCoord(n_i, n_j) = 0$.

Because the word2vec model can detect semantically similar words, we simply use the cosine similarity of learned vectors to compute the similarity between two verbs. Hence, $IsSynonym(v_i, v_j)$ returns a value between 0 and 1.

5.3.3 Calculating the coordinate subject degree

Let us consider how to calculate the coordinate degree between two noun sets. Because we already know whether two given terms are coordinate, it is easy to know the number of pairs of coordinate terms between the two sets. However, to reduce the computational complexity, we do not do pairwise comparison. Instead, we think for each term in one noun set, there is at most one coordinate term in another set. If we already find a pair of coordinate terms, e.g., *the Oregon school shooting* in d_i coordinate to *the Virginia Tech shooting* in d_j, other pairs containing one term in the found pair, such as *the Oregon school shooting* in d_i coordinate to *Virginia* in d_j, are just an enhancement, because we already find a term that plays a similar role as *the Oregon school shooting* in d_j. Therefore, we keep on removing pairs of coordinate terms that have already found and making term comparison within the rest terms in both of the two sets. As a result, we can get pairs of coordinate terms and exactly the same terms. The calculation of coordinate subject degree is based on these pairs.

It is easy to understand that if two documents have many shared proper nouns, they are more likely to describe the same event rather than different ones. Consequently, we should not assign a high coordinate degree to such documents. Based on this consideration, we introduce the following three methods listed below to calculate $CoordSub(d_i, d_j)$, called ***minS1***, ***minS2***, ***minS3***,

respectively.

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{min\{|N_i|, |N_j|\} + |N_i \cap N_j|} \tag{5.2}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{min\{|N_i|, |N_j|\}} \cdot (1 - \frac{|N_i \cap N_j|}{min\{|N_i|, |N_j|\}}) \tag{5.3}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{min\{|N_i|, |N_j|\} + e^{|N_i \cap N_j|}} \tag{5.4}$$

The *minS1* method penalizes sets that have the same proper nouns, the *minS2* method adds the ratio of the same ones to eliminate their effects, and the *minS3* method enhances the penalty of the same ones.

In addition, we find that, if the size difference between N_i and N_j is very large, the larger set tends to contain more common proper nouns by chance. Consequently, we should take the size difference into consideration. As a result, compared to the above methods considering the smaller size of the two sets , we introduce methods considering the average size, or the larger size as follows.

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{avg\{|N_i|, |N_j|\} + |N_i \cap N_j|} \tag{5.5}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{avg\{|N_i|, |N_j|\}} \cdot (1 - \frac{|N_i \cap N_j|}{avg\{|N_i|, |N_j|\}}) \tag{5.6}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{avg\{|N_i|, |N_j|\} + e^{|N_i \cap N_j|}} \tag{5.7}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{max\{|N_i|, |N_j|\} + |N_i \cap N_j|} \tag{5.8}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{max\{|N_i|, |N_j|\}} \cdot (1 - \frac{|N_i \cap N_j|}{max\{|N_i|, |N_j|\}}) \tag{5.9}$$

$$CoordSub(d_i, d_j) = \frac{\sum_{n_m \in N_i, n_n \in N_j} IsCoord(n_m, n_n)}{max\{|N_i|, |N_j|\} + e^{|N_i \cap N_j|}} \tag{5.10}$$

where Formula (5)-(7) denote methods that are called *avgS1*, *avgS2* and *avgS3*, respectively. Formula (8)-(10) denote methods that are called *maxS1*, *maxS2* and *maxS3*, respectively.

5.3.4 Calculating the similar action degree

Let us consider how to calculate the similarity degree between two verb sets. Because we already know whether two given verbs are similar, it is easy to know the number of verbs that are the same or similar in the two sets[7]. Similarly, we do not do pairwise comparison. Instead, we keep on removing synonym pairs that have already found and making term comparison with the rest terms in both of the two verb sets. The similarity degree between the two verb sets are computed based on the number of verbs that are the same and similar in the two sets.

$$SimAct(d_i, d_j) = \frac{|V_i \cap V_j| + \sum_{v_m \in V_i, v_n \in V_j} IsSynonym(v_m, v_n)}{avg\{|V_i|, |V_j|\}} \tag{5.11}$$

5.4 Constrained Clustering

In this section, we introduce a constrained clustering algorithm to cluster documents in a manner closer to that of human cognition. We take coordinate documents into account and assume that such documents should not be grouped into the same cluster. Such two documents hold a **cannot link** constraint [62]. Correspondingly, we also consider documents that are mutually inclusive in semantics and assume that such documents should be grouped into the same cluster. Similarly, such two documents hold a **must link** constraint [62]. Therefore, we first introduce the method to detect these two types of constraints, respectively.

5.4.1 Cannot Link Detection

Because pairs of coordinate documents can be regarded as cannot links, we simply use the coordinate degree between two documents as their cannot link degree, written as below.

$$Cannot(d_i, d_j) = Coord(d_i, d_j) \tag{5.12}$$

5.4.2 Must Link Detection

We also focus on subjects and actions to detect documents that hold must links. Compared to coordinate documents that describe different events, documents that hold must links describe the same event. Therefore, it is not difficult to think up that such documents should have the same or similar subjects and similar actions. We define another function called $SimSub(d_i, d_j)$ as follows.

***SimSub*(d_i, d_j)**

Given two documents d_i and d_j, this function returns the similar subject degree between them, which is a value between $[0, 1]$. When $i = j$, its value is 1.

[7] Note that verbs are compared in their base form.

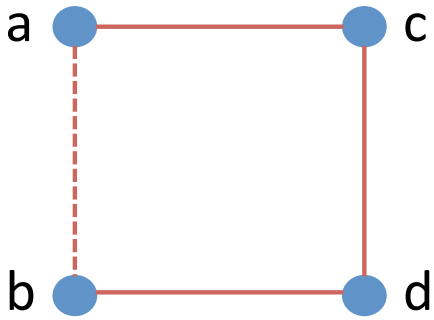


Figure 5.3: A situation of failure in COP-KMeans algorithm. The instance is represented by circle. The solid line between two instances indicates the must link constraint, while the dash line indicates the cannot link constraint.

The problem is given two noun sets N_i and N_j that are extracted from d_i and d_j, respectively, we want to compute the similarity between these two sets. Similar to what we did in Section 5.3.4, we use the two methods, WordNet and word2vec, to estimate the similarity degree between the two proper nouns. Then we compute the similar subject degree based on the number of proper nouns that are the same and similar in the two sets.

$$SimSub(d_i, d_j) = \frac{|N_i \cap N_j| + \sum_{n_m \in N_i, n_n \in N_j} IsSynonym(n_m, n_n)}{avg\{|N_i|, |N_j|\}} \tag{5.13}$$

Based on the assumption that documents that hold a must link should have the same or similar subjects and similar actions, we can combine the similar subject degree and the similar action degree via a weighted harmonic mean to compute the must link degree between the two documents. This formula is given below.

$$Must(d_i, d_j) = \frac{1}{\beta \cdot \frac{1}{SimSub(d_i, d_j)} + (1-\beta) \cdot \frac{1}{SimAct(d_i, d_j)}} \tag{5.14}$$

where β is the weight. When $\beta = 1$, only the subjects are considered when calculating the must link score; and when $\beta = 0$, only the actions are considered. To simplify, let β equal to the weight α in Formula (1).

5.4.3 The Constrained Algorithm

Several studies have introduced the clustering algorithms with some constraints [62][5][35]. COP-KMeans [62] is well known among these algorithms. However, COP-KMeans fails if any specified constraints are violated, which leads the empty partition ($\{\}$) returned. Here, we suppose the cluster number $k = 2$. Figure 5.3 shows a situation in which the algorithm fails. Fom the figure, we know that the instances a and c must be in the same group, so are the pair of c and

Table 5.1: Illustration for 5 datasets.

Topic	Event
school shooting	Red Lake shooting, Virginia Tech shooting, West Nickel Mines School shooting
earthquake	Bam earthquake, Indian Ocean earthquake, Kashmir earthquake
heat wave	2003 European heat wave, 2006 North American heat wave, 1995 Chicago heat wave
terrorist attack	Bali bombings, Madrid train bombings, Mumbai train bombings
accounting scandal	Enron scandal, WorldCom scandal, Freddie Mac scandal

d, the pair of b and d. Because the must link satisfies transitivity, we can infer that the instances a and b must be in the same group. However, there is a cannot link between them, which leads a violation. Actually, when the number of constraints is large and the cluster number k is small, the COP-KMeans algorithm will frequently fail. Therefore, COP-KMeans is not suitable for our problem because the constraints of cannot links and must links are automatically generated and may contain some mistakes.

As a result, we propose another constrained clustering algorithm for solving our problem, shown in the following frame. The basic idea is that if there is a cannot link between two clusters, their distance should be increased; if there is a must link between two clusters, their distance should be decreased.

Algorithm: Constrained Clustering

Input: Set of documents $D = \{d_1, d_2, ..., d_N\}$,
set of cannot links $CL = \{(d_i, d_j)\}$,
set of must links $ML = \{(d_i, d_j)\}$,
weight for a cannot link w_C,
weight for a must link w_M,
maximum distance to join clusters d_{max},
cluster number k (optional).

Output: Disjoint partitioning $C = \{C_1, C_2, ...\}$ of D.

Method:

1. Remove the constraint (d_i, d_j) in both CL and ML.
2. Create the cannot point set $CP = \{p_1, p_2, ...\}$ from CL.
3. Initialize cluster labels for each point d_i in D in a manner such that
   - for each pair of points in CL, assign different cluster labels to the two points;
   - for each pair of points in ML, assign the same cluster label to the two points.
   - for other points, assign no labels.
4. For each point d_i in D except for those in CP, assign it to the closest p_j in a manner such

that

- if d_i is not a point in ML, calculate its distance to each p_l. Assign the cluster label of the closest p_j to it.

- if d_i is a point in ML, find other points that consists of a must link with it. Then calculate their distance to each p_l and assign the cluster label of the closest p_j to all of them.

5. Calculate the distance between any two clusters as follows.

$$\begin{aligned} d(C_i, C_j) =& max(dist(C_i[m], C_j[n])) \\ &+ w_C \cdot cannot\_links(C_i, C_j) \\ &- w_M \cdot must\_links(C_i, C_j) \end{aligned}$$

6. Sort pairs of clusters by distance.

7. Traverse the sorted result and join clusters if their distance $< d_{max}$, to keep their cluster labels the same.

8. Iterate between (4) and (6) until convergence or the current cluster number $< k$.

9. Return $\{C_1, C_2, ...\}$.

Here, $cannot\_links(C_i, C_j)$ indicates the number of cannot link constraints between the two clusters. Similarly, $must\_links(C_i, C_j)$ indicates the number of must link constraints between the two clusters. $dist(C_i[m], C_j[n])$ computes the cosine distance between the $tf \cdot idf$ vectors of $C_i[m]$ and $C_j[n]$. Moreover, to simplify, let $w_C = w_M = d_{max}$.

However, because at the beginning of the clustering, points are assigned to their closest cannot points, our algorithm will get a bad result when both the number of constraints and the cluster number k are small. Suppose there is only one cannot link and the cluster number $k = 3$. In this case, all points are assigned to either of the two cannot points. It leads to a result of 2 clusters.

In contrast, when the number of constraints is large enough, there is no need to indicate the cluster number k because the algorithm will stop until no more clusters can be joined together.

5.5 Experiments

In this section, we outline the details of our experiments.

5.5.1 Datasets

Although there are several well-known public datasets for clustering, the granularity is large. In other words, they are composed of documents from different topics, even though there may be groups on similar topics. However, in this study, we aim at grouping documents at a smaller granularity, e.g., grouping news articles according to events they describe.

Table 5.2: Combination of methods used in the experiments.

Methods for term comparison	weight α	Methods to calculate $CoordSub(d_i, d_j)$
WordNet	0.7	minS3, avgS3, maxS3
word2vec	0.5	minS3, avgS2, maxS2

Therefore, we manually created our datasets, containing 5 datasets (shown in Table 5.1), for evaluation. In details, each dataset corresponds to a topic and in each dataset, there are 3 different events. For each event in each topic, we surveyed on the Web and extracted related news articles from the New York Times Annotated Corpus. Because the corpus only contains news articles published by the New York Times, there are not so many articles found for each event. Besides, we manually removed noises from each dataset, e.g., news articles summarized the news in the past period. Finally, we got $60 \sim 90$ news articles for each topic.

We regard the name of each topic as a user-given query and aim at clustering news articles to revert to the 3 different events. For example, we devote to separate articles in the topic "terrorist attack" in three main groups, while each group describes an event listed in Table 5.1.

The datasets used for the evaluation include the correct label for each article. In other words, the event each article describe is known. Therefore, we use the labels for evaluating performance.

5.5.2 Experimental Setting

The word2vec model was trained on the New York Times Annotated Corpus with article publication time from 2002 to 2007.

Because of the space limitation and numerous combinations of the methods, we cannot show all experimental results. We did a preliminary experiment to choose the combinations. We found when using the WordNet to compare two terms, a good choice for the weight α of harmonic mean in Formula (5.1) is 0.7. And the methods *minS3*, *avgS3* and *maxS3* perform better. When using the word2vec model, a good choice for α is 0.5 and the methods *minS3*, *avgS2* and *maxS2* perform better. The combinations of methods used in the experiments are listed in Table 5.2.

5.5.3 Experimental Results for Finding Coordinate Documents

Two baseline methods are employed for better comparison.

sim We employ vector space model [58] and represent each article by a feature vector. Similarity between two articles is computed by cosine similarity of their feature vectors. We denote the *sim* method a ranking based on the similarity to the query article.

MMR Maximal Marginal Relevance (MMR), a document summarization method proposed by Carbonell and Goldstein[12], is widely used when diversity needs to be considered. Since we aim at finding documents with the same topic but talking about different events, it is

also a diversity problem. When the method is used to select desired documents, it considers both relevance of each selected document and diversity of the whole output. That is, the output should not contain similar documents even if they are relevant.

We randomly select 10 news articles in each dataset and regard each of them as an input. Therefore, we evaluate the proposed methods on 50 queries.

It is a time-consuming and expensive process involving human beings to assess relevance of each article in the ranking list by given one as the input. In information retrieval, it is usual for relevance to be assessed only for a subset. *Pooling* is the most standard approach to reducing the burden. Therefore, we first take a subset of the collection that is formed from the top k ($k = 10$) articles returned by the proposed methods and two baseline methods. Then we manually evaluate articles in the pooled set under the criterion that they state the same development phase of different news events (compared with the query article).

We use precision and recall, two traditional evaluation metrics in information retrieval, for evaluation.

Table 5.3 shows the performances of the proposed methods and two baseline methods, using precision for evaluation. It details the precision for each method at different k. From the table, we know that

(1) using the WordNet method to compare terms can get a better precision score than using the word2vec method. Besides, all the three methods *WordNet+minS3*, *WordNet+avgS3* and *WordNet+maxS3* outperform both of the two baseline methods (*sim* and *MMR*), no matter what k is. Especially, the methods *WordNet+avgS3* and *WordNet+maxS3* significantly improve the precision when $k \leq 5$.

(2) the performance of the method *WordNet+minS3* is quite stable, which means the precision does not change a lot with the increase of k.

(3) using the word2vec method to compare terms is less than satisfactory. Only *word2vec+minS3* is competitive with the baseline method *sim*.

(4) the precision for the *sim* method does not decrease a lot with the increase of k, while the precision for the *MMR* method changes differently.

Table 5.4 shows the performances of the proposed methods and two baseline methods, using recall for evaluation. It also details the recall for each method at different k. We can find similar phenomenon to that of the precision. Using the WordNet method to compare terms can get a better recall score than using the word2vec method. Especially, the three methods *WordNet+minS3*,

Table 5.3: Precision at different k.

@k	WordNet			word2vec			sim	MMR
	minS3	avgS3	maxS3	minS3	avgS2	maxS2		
1	0.400	**0.800**	0.400	0.600	0.200	0.200	0.400	0.340
2	0.400	**0.500**	**0.500**	0.400	0.300	0.300	0.300	0.310
3	0.400	0.400	**0.467**	0.400	0.267	0.267	0.333	0.267
4	0.400	0.350	**0.450**	0.400	0.250	0.250	0.300	0.295
5	**0.400**	0.360	0.400	0.320	0.240	0.240	0.280	0.200
6	**0.400**	0.30	0.367	0.267	0.233	0.267	0.300	0.167
7	**0.371**	0.314	0.314	0.229	0.229	0.257	0.314	0.143
8	**0.375**	0.325	0.325	0.225	0.200	0.225	0.300	0.250
9	**0.422**	0.356	0.3556	0.200	0.200	0.244	0.267	0.222
10	**0.440**	0.380	0.360	0.220	0.220	0.240	0.260	0.296

Table 5.4: Recall at different k.

@k	WordNet			word2vec			sim	MMR
	minS3	avgS3	maxS3	minS3	avgS2	maxS2		
1	0.029	**0.065**	0.040	0.052	0.012	0.012	0.037	0.040
2	0.059	**0.081**	**0.081**	0.065	0.050	0.050	0.051	0.058
3	0.083	**0.096**	0.108	0.091	0.075	0.075	0.077	0.080
4	0.112	0.111	**0.138**	0.117	0.087	0.087	0.091	0.093
5	0.142	0.137	**0.151**	0.117	0.099	0.099	0.106	0.117
6	**0.168**	0.137	0.163	0.117	0.112	0.124	0.131	0.146
7	**0.181**	0.165	0.163	0.117	0.125	0.137	0.158	0.167
8	**0.208**	0.190	0.189	0.128	0.125	0.137	0.183	0.176
9	**0.273**	0.241	0.239	0.128	0.137	0.163	0.183	0.188
10	**0.323**	0.292	0.276	0.165	0.174	0.175	0.197	0.231

WordNet+avgS3 and *WordNet+maxS3* significantly improve the recall when $k \geq 9$. And unfortunately, using the word2vec method to compare terms cannot make any improvement in recall, compared to either the baseline method *sim* or the baseline method *MMR*.

F-measure is one of the other traditional evaluation metrics in information retrieval, which considers both the precision and the recall to compute the score. It can be interpreted as a weighted harmonic mean of the precision and the recall, where an F-measure score reaches its best value at 1 and worst at 0. Figure 5.4 shows the performance of our proposed methods and the two baseline methods, evaluated by F-measure. From this figure, we can point out that all methods using WordNet for term comparison outperform both of the two baseline methods *sim* and *MMR*. However, when using the word2vec model to compare terms, the performance turns to be bad. The method *word2vec+minS3* performs better than both *sim* and *MMR* only when $k \leq 5$.

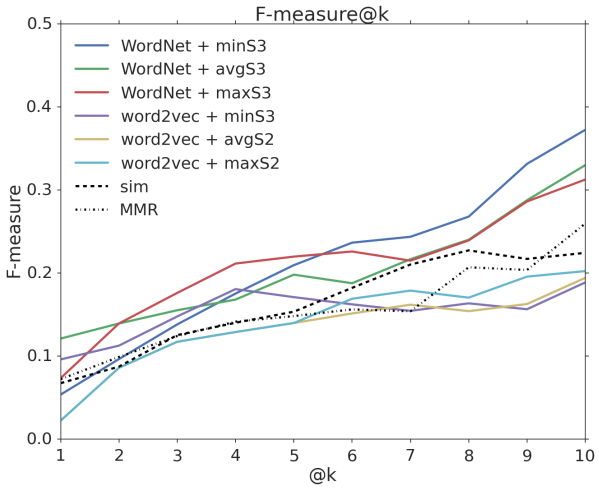


Figure 5.4: Performance evaluated by F-measure.

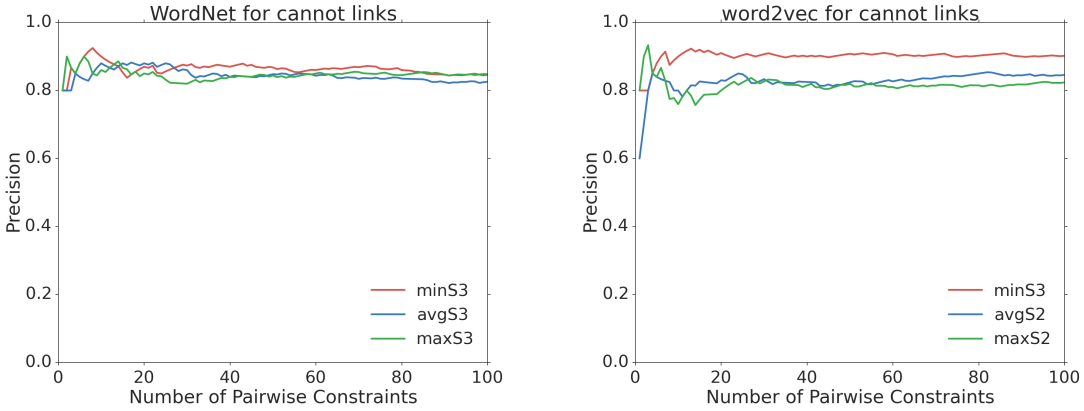


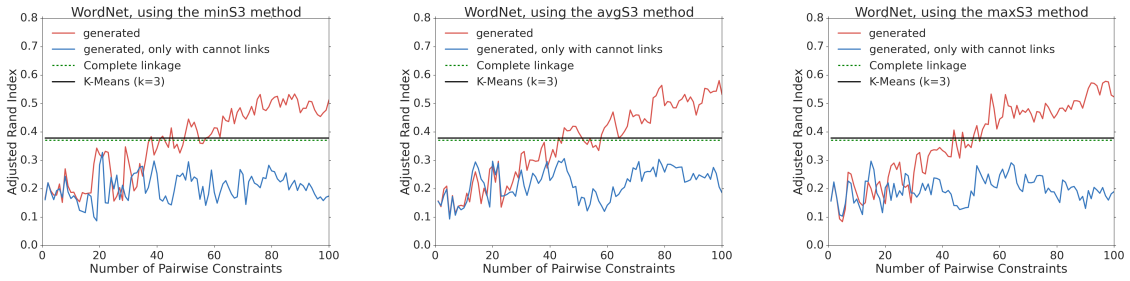
Figure 5.5: Performance of detecting cannot links.

To conclude, compared to the two baseline methods only considering similarity, our proposed methods employing WordNet for term comparison can significantly improve both precision and recall, which illustrates the importance of coordinate relationship to find similar but different documents.

5.5.4 Experimental Results for Detecting Constraints

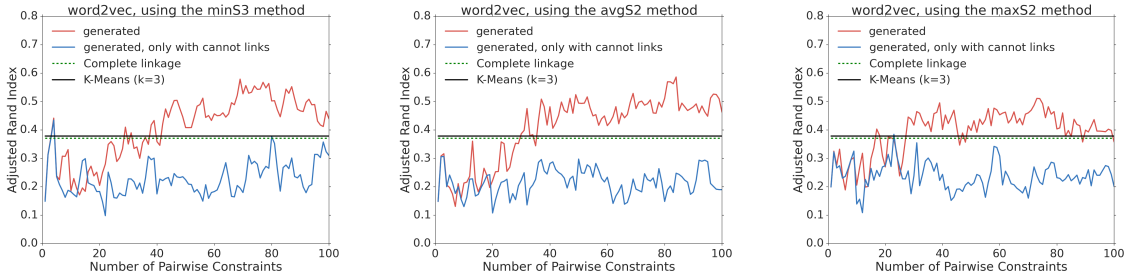
Cannot Link Detection

For each dataset, we calculate pairwise coordinate degree of any two articles and sort them in descending order according to the calculation. Because pairs of coordinate documents can be regarded as cannot links, those with high coordinate degrees are more likely to be cannot links.



(a) (b) (c)

Figure 5.6: Performance of the clustering, using the WordNet method to compare terms. ($\alpha = 0.7$)



(a) (b) (c)

Figure 5.7: Performance of the clustering, using the word2vec method to compare terms. ($\alpha = 0.5$)

A pair of coordinate document form a cannot link. However, documents in a cannot link are not always coordinate to each other. Here, we evaluate top pairs in the sorted result by considering whether the two documents in a pair belong to different events. Therefore, they could be at the different development phases, viz., not coordinate according to our definition in Section 5.3.

Figure 5.5 shows the performance of detecting cannot links, averaged on the 5 datasets. The x axis indicates the number of constraints we took from the top of the above sorted result. The y axis indicates the precision. Figure 5.5(a) is the performance when using the WordNet to compare terms, while Figure 5.5(b) is the performance when using the word2vec model.

From the figure, we know that for the WordNet method, there is no big difference among the three methods *minS3*, *avgS3* and *maxS3* to calculate $CoordSub(d_i, d_j)$ and all of them are stable. Even in the worst case, it can find cannot links with a precision about 0.8.

However, for the word2vec method, the method *avgS2* cannot find correct cannot links within the top. However, the three methods perform well in lower position, e.g., behind the top 20 constraints. Besides, *minS3* achieves a precision near 0.9.

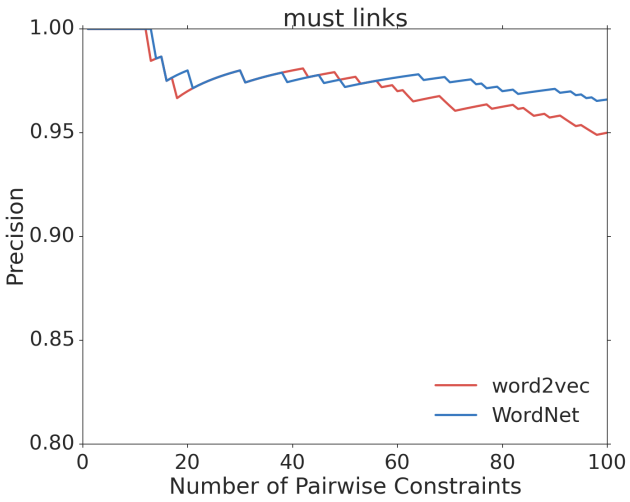


Figure 5.8: Performance of detecting must links.

Must Link Detection

Similarly, for each dataset, we calculate pairwise must link degree of any two articles and sort them in descending order according to the calculation. Those with high scores are more likely to be must links. We evaluate top pairs in the sorted result by considering whether the two documents in a pair belong to the same event.

Figure 5.8 shows the performance of detecting must links, making a comparison between the methods WordNet and word2vec. We know that the proposed methods to find must links are excellently good. Both the two methods obtain the precision over 0.95.

5.5.5 Experimental Results for Clustering

Evaluation Method

We used the metric adjusted Rand index (ARI), proposed by Hubert and Arabie [26], for cluster evaluation.

The ARI is the corrected-for-chance version of the Rand index (RI) [57]. Given a set of n elements, if C is a ground truth class assignment and K the clustering, let us define a and b as the number of pairs of elements that are in the same set in C and in the same set in K and the number of pairs of elements that are in different sets in C and in different sets in K, respectively. The RI is computed by

$$RI = \frac{a+b}{\binom{n}{2}}$$

However the RI does not guarantee that random label assignments will get a value close to 0. To counter this effect we can discount the expected RI of random labelings by defining the ARI

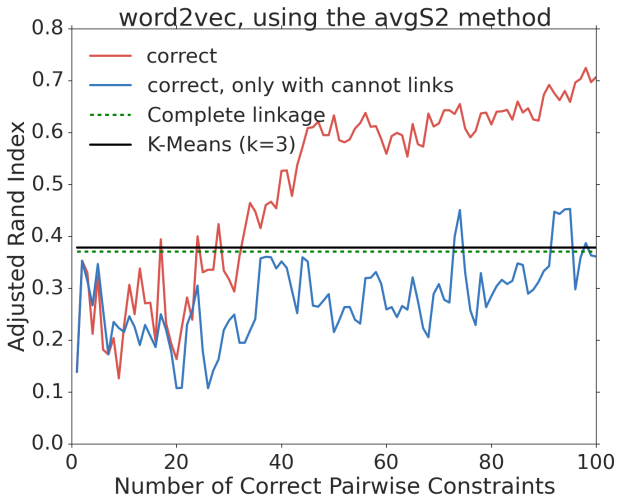


Figure 5.9: Performance of the clustering, using only correct constraints. ($\alpha = 0.5$)

as follows.

$$ARI = \frac{RI - ExpectedRI}{max(RI) - ExpectedRI}$$

It yieldes a value between $[-1, 1]$. Negative values are bad (independent labelings), similar clusterings have a positive ARI.

Constrained Clustering

Because our objective is to group news articles according to events they describe, it is a clustering problem. We use K-Means [37] and complete linkage clustering as the baselines. All the evaluations here take the average on the 5 datasets.

Figure 5.6 shows the performance of clustering, using the WordNet method to compare terms and the weight of harmonic mean in Formula (1) and (11) is 0.7. We consider two situations here: (i) use both cannot links and must links detected by our method and make the number of cannot links equal to that of must links (named "generated" for short). (ii) use only cannot links detected by our method as constraints (named "generated, only with cannot links" for short). The performance of the former situation is indicated by the red solid line, while the performance of the latter situation is indicated by the blue solid line in Figure 5.6.

Figure 5.7 shows the performance of clustering, using the word2vec method to compare terms and the weight of harmonic mean is 0.5. We also consider two situations, "generated" and "generated, only with cannot links".

Compared Figure 5.7 with Figure 5.6, we know that (i) only taking the cannot links as constraints is useless to improve the clustering result, which suggests us to use both cannot links and must links when doing clustering. (ii) The ARI scores are not ideal, even worse than the two

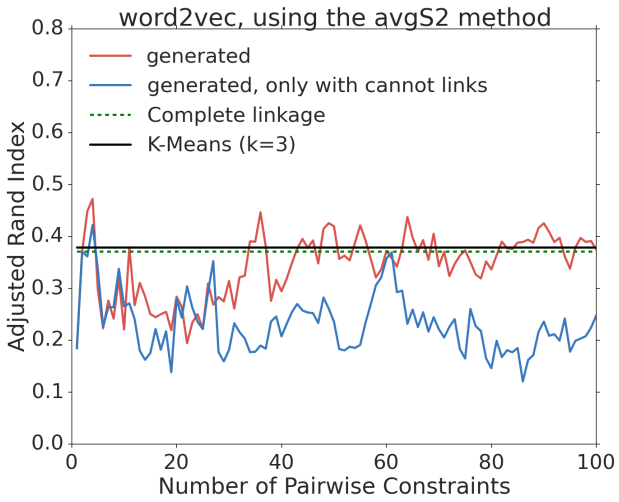


Figure 5.10: Performance of the clustering, using the word2vec method to compare terms. ($\alpha = 1.0$)

baselines, when the number of constraints is not large enough, no matter which method we use to compare terms. (iii) For the WordNet method, only when the number of constraints is greater than 50, the "generated" outperforms both K-Means and complete linkage clustering. (iv) For the word2vec method, when the number of constraints is greater than 30, the "generated" outperforms the two baselines. (v) With the increase of the number of constraints, the performance of the WordNet method keeps growth, while the performance of the word2vec method has a trend of rise first then fall. Because for different methods, e.g., *WordNet, using the minS3 method* vs. *word2vec, using the minS3 method*, the top generated cannot links or must links are different. As a result, when comparing the WordNet with the word2vec, we can infer that the quality of constraints generated by the WordNet is better than that of the word2vec.

Figure 5.9 shows the performance of the clustering, using only correct constraints. The difference compared to Figure 5.7(b) is that we remove the incorrect cannot links or must links. We find that even for the "correct, only with cannot links", its performance could be better than the baselines. Even the "correct" still performs badly when the number of correct constraints is smaller than 36, with the increase of the number of correct constraints, the performance keeps in growth and when the number of correct constraints is 100, we obtain the ARI score of 0.7, dramatically better than the two baselines (both are around 0.37). This illustrates the effectiveness of our constrained clustering algorithm.

Figure 5.10 shows the performance of the clustering, using the word2vec method and the weight of harmonic mean is 1.0. It means that in this case, only proper nouns are considered to calculate the coordinate degree and the must link degree. Verbs are ignored. Compared to Figure

5.7(b), we find that the overall performance turns worse. This confirm the assumption in Section 5.3 that coordinate documents should have coordinate subjects and similar actions.

To conclude, we automatically generate constraints for clustering and achieve 0.1 improved in the ARI score. While in previous studies [62][5][35], constraints are artificially generated which makes the cost of the clusterings very high.

5.6 Summary

We propose a method to structure search results of a user-given query and show intrinsic connections between individual documents. To accomplish this goal, we introduce the concept of "coordinate documents" as documents talk about the same topic and they describe similar but different events or concepts. We consider coordinate documents are mutually exclusive in semantics. Consequently, they should not be grouped into the same cluster. Based on this criterion, news articles are organized according to events they describe. Experiments conducted on the New York Times Annotated Corpus verify the effectiveness of our method and illustrate the importance of coordinate relationships in finding coordinate documents and clustering search results.

CHAPTER 6

PANORAMIC IMAGE SEARCH BASED ON SIMILARITY AND ADJACENCY

6.1 Introduction

In recent years, the increasing development of the Internet and its related technologies, has led to the appearance of a variety of multimedia content formats on the Internet, such as images, audio, and video. Various search technologies are often used to retrieve desired contents. To improve user satisfaction in multimeadia retrieval, content-based image retrieval has been studied extensively and incorporated into retrieval frame. Currently, it is possible to search for images based on their contents, irrespective of features such as their color, brightness, and texture. As a result, some recent studies have focused on the development of search systems that try to detect images similar to the query image. An example is TinEye [1], a Web search service, which returns images that are very similar to a query image and there is no need to enter a keyword.

Among a variety of search intents, we believe that attempts to identify locations similar to a location, to which a user is familiar, are common. For example, users may imagine a scene in which there is a garden around the Kyokochi pond , which reflects Shari-den of the Golden Pavilion temple on its surface. The task is to find locations with scenes that are similar to those in users' imaginations. In this case, the desired results should be different from the location in users' mind. It is straightforward to search by a keyword that indicates a location in the user's imagination, or an image that represents the scene imagined by the user, but it is still difficult to realize this goal.

Even if each photo is attached with a geo-tag, it is still challenging. Because it is only useful

[1] http://www.tineye.com/

for short-range views, but for intermediate or distant views, the spot where a photo is taken may be distant from the place described in the photo. For example, we can take photos of Mt. Fuji in Tokyo, Chiba prefecture, or Saitama prefecture. However, the geo-tags attached to these photos are obviously different from where Mt. Fuji locates. Besides, it is hard to find boundaries of landscapes and photos are likely to appear somewhat different even belonging to the same landscape.

In this paper, we propose a novel kind of image search that given an image or a few images of a location, images which is atmosphere-similar to query image(s), especially, those of other locations, will be returned. In general, atmosphere-similar images in a location compose a representation of a "landscape". Notice there is no need for images to be visually similar to each other. As images in the same landscape always have some overlaps with each other, we incorporate adjacency (See details in Sec.6.3.2) into our scheme. We propose an image ranking method called PanoramaRank: a combination of image similarity and image adjacency to discover a certain landscape in a location. In Section 6.2, we describe a simple survey and give our definition of "landscape". In Section 6.3, the basic idea and the problems are addressed. Section 6.4 explains our proposed method PanoramaRank and Section 6.5 presents how to discover similar landscapes. Our evaluation experiments are addressed in Section 6.6. Finally, conclusions and possible directions for future work are stated in Section 6.7.

6.2 Brief Introduction to "Landscape"

6.2.1 Landscape

The word "landscape" is widely used in many fields, such as geography, urban engineering, social technology, and architecture. In 2004, the Landscape Act was enacted in Japan, but there is no precise legal definition of what constitutes a landscape. According to our survey, there are two main arguments. The main difference between them is whether or not the components that constitute a landscape should contain human elements.

In [44], Nakamura et al. determined that the concept of a landscape can be summarized in the following five points. First, different objects such as buildings or structures exist simultaneously and are associated with each other. Second, a certain space has a specific form. Third, there is a hierarchy where the size of the space is concerned. Fourth, the landscape is a type or a model. Finally, a landscape will change over time. For example, a bridge cannot by itself be referred to as a bridge landscape, but it is an element of a bridge landscape. It must therefore be considered in relation to other elements such as the river, traffic network and community.

On the other hand, in architecture, landscape can be divided into its scenery, which is viewed by people, and feeling, which is felt by people viewing the landscape. Furthermore, a scene

may be subdivided into locality, integration, and the extent of exposure to the public. A feeling is subdivided into diversity, lifestyle and participation. In particular, even without seeing an actual landscape, visualization is possible from past experiences and information obtained. In this way, images conjured by persons are referred to as landscape images (different with the below-mentioned "landscape images"). Considering the types of appearance pertaining to landscapes, even if the same objects are seen, the impression on the viewer may be different because of the distance (whether it is a short-range view, a distant view, or an intermediate view). Likewise, our impression of a landscape would vary according to the season, or time at which it was seen.

In this paper, we exclude factors that refer to feelings, i.e. psychological factors, which include but are not limited to differences that result from personal experiences, background, education, and religion. We prefer to define "landscape" as an area that produces the same response from most people. Moreover, since a landscape is a specific area, an image set, a set of images of a particular area, is necessary to integrally and factually represent a landscape.

6.2.2 Landscape Images

As mentioned in Section 6.2.1, due to changes in season, or time, there may be diverse landscapes at the same location. Furthermore, what we refer to as an area is not an exact geographical area. For example, the Golden Pavilion Temple does not refer to the golden reliquary hall, or the entire area belonging to the temple. It is actually a subarea of an exact geographical area, or sometimes even includes an extended surrounding area. Fig.6.1 is an example of a landscape of the Golden Pavilion Temple, mainly describing the environment of the golden reliquary hall and the surrounding Kyokochi pond. Fig.6.2 reflects the Japanese style in a quiet and calm atmosphere. Although the area changes little in Fig.6.3 compared to that in Fig.6.1, it is a different landscape because of the seasonal change. We consider these three landscapes at the same location as distinct ones. From these three examples, we observe that

(a) A whole space can be divided into several distinct landscapes, as each landscape image set introduces a somewhat different impression.

(b) Even if the area represented in an image does not change much, it can become another landscape as time progresses.

(c) These three distinct landscapes, together with others not listed here, generate an upper-level landscape of the Golden Pavilion Temple, which means there exists a hierarchy related to the word "landscape".

Figure 6.1: An example of a landscape surrounding the Golden Pavilion Temple, mainly describing the environment of the golden reliquary hall and the surrounding Kyokochi pond (mirror pond).

Figure 6.2: Another example of a landscape surrounding the Golden Pavilion Temple: this is not a typical landscape. This photo set reflects the Japanese style in a quiet and calm atmosphere.

6.3 Basic Idea

Similar landscapes are geographically separated but can invoke similar impressions in most people. A simple example is provided in Fig.6.4. Thus, the problem described in this paper is as follows:

- **Input**: At least one image, which partly indicates the desired landscape
- **Output**: Ranked landscape image sets that represent different landscapes similar to the

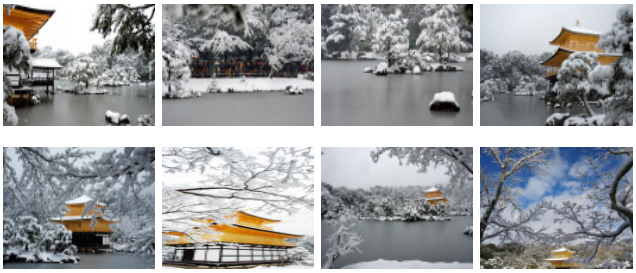


Figure 6.3: Example of a landscape surrounding the Golden Pavilion Temple, describing mainly the environment of the golden reliquary hall and the surrounding Kyokochi pond (mirror pond), in an area that is similar to that of Figure 6.1. However, this set is for a winter landscape, which illustrates the seasonal changes that occur.

(a) A landscape of Ryoan-ji

(b) A landscape of Tenryu-ji

Figure 6.4: Both Fig.6.4(a) and Fig.6.4(b) are typical representations of a traditional Japanese dry landscape garden. From this point of view, we consider landscapes described by those two photo sets to be similar.

given landscape

Based on our hypothesis that a specific area usually cannot be represented by just a single image, we consider the following two distinct relations between images.

6.3.1 Image Similarity

This problem is as follows, given an image as our input, other images that are similar to the input image are expected to be flagged based on their similarity to the input image. Suppose there is an image dataset I_n as follows, I_n=$\{img_1, img_2, ..., img_i, ..., img_n\}$, where n is the total number of images in this dataset. When any one image img_i in the dataset is given as an input, the expected output is similarities between other images in the dataset and the input image. The functional form representing this problem is as follows,

$$Sim(img_i, img_j)$$

where img_j is any other image in the dataset, $1 \leq i \leq n$, $1 \leq j \leq n$. It is desired that this function returns a real number, which has a high value when the content of img_j is highly similar to that of img_i, and vice versa.

6.3.2 Image Adjacency

In this section, we will explain the concept of image adjacency in details and describe how it is applied to image search. Fig.6.5(a)-6.5(c) is the left, middle, right part of the clock tower, respectively. Note that there is no overlap between Fig.6.5(a) and Fig.6.5(c). However, there are common features in the photo pairs of Figs.6.5(a) and 6.5(b), and Figs.6.5(b) and 6.5(c). For example, both Figs.6.5(a) and 6.5(b) contain the top of the clock tower and a small part of the camphor tree. Therefore, these two photos are referred to as adjacent images. The same applies

(a) Left (b) Middle (c) Right

Figure 6.5: Three photos taken in front of the main gate of Kyoto University

to Figs.6.5(b) and 6.5(c). Notice that we don't aim to recognize each object, but only consider overlap between two images.

The problem is, then similar to that of image similarity. Given an image as our input, it is expected that other images adjacent to this image will be highlighted in terms of adjacency with the input image. When any one image img_i in the dataset is given as an input, the expected output is adjacencies between other images in the dataset and the input image. The functional form representing this problem is as follows,

$$Adj(img_i, img_j)$$

$1 \leq i \leq n$, $1 \leq j \leq n$. A high value will be returned when img_j is highly adjacent to img_i, in other words, when it is most likely that img_j and img_i can generate a panorama. And vice versa.

We combine image similarity and image adjacency together to find images roughly similar and roughly adjacent to query image(s) no matter whether directly or indirectly.

6.4 PanoramaRank

Here we explain our graph-based image ranking method PanoramaRank in detail. Roughly speaking, we employ PanoramaRank to discover a landscape in a location. Image similarity and image adjacency is combined in our method to find images both similar and adjacent to chosen image(s).

6.4.1 Calculating Image Similarity

In this section, we apply existing methods to calculate image similarity.

Search through a vocabulary tree:In [46], a descriptor vector is retrieved to propagate down the hierarchical vocabulary tree, in which each node is assigned a weight, so that we get the searching image query vector. Then, the relevance score assigned to the database image can be defined based on the normalized difference between the searching image query vector img_i and the database image vector img_j, indicated by $NisterScore(img_i, img_j)$.

Color:We compare images by employing color coherence vectors(CCV), a histogram-based

method to compare images that incorporate spatial information, which was proposed by Pass et al. in 1996 [53]. For each color, the number of coherent pixels relative to the number of incoherent pixels is stored. The method used to compare the difference between two images is based on the differences between coherent pixels for each discretized color in both images as well as the differences between incoherent pixels. We use CCVScore(img_i, img_j) to indicate the difference estimated by this method.

Our image similarity between two images img_i and img_j is considered as

$$Sim(img_i, img_j) =$$

$$w_1 NisterScore(img_i, img_j) + w_2 CCVScore(img_i, img_j)$$

where w_1 and w_2 are weight factors of different scores, and $w_1 + w_2 = 1$.

6.4.2 Calculating Image Adjacency

In this section, we apply the panorama construction algorithm [11] to find corresponding pixels that can generate a panorama using two images.

Given two images img_i and img_j, first we extract SURF descriptors from each image, separately. Then we find correlation points in these two images. Finally, we find geometrically consistent feature matches using RANSAC to solve for the homography between these two images. We treat the inner-most points as the corresponding points between these two images. Here we use N_{cp} to refer to the number of corresponding points in query image img_i and database image img_j, and N_{img_i} to the number of extracted SURF descriptors in the query image, N_{img_j} to the number of extracted SURF descriptors in any database image img_j, respectively. Then image adjacency is defined as

$$Adj(img_i, img_j) = \frac{N_{cp}}{(N_{img_i} + N_{img_j})/2}$$

6.4.3 Similarity/Adjacency Graph and PanoramaRank

In [28], VisualRank, which is an inferred visual similarity graph-based ranking model for image-ranking problems, was introduced. In this model, edge weights have also been considered when estimating the score associated with a vertex in the graph. The random walk algorithm is employed to rank images based on the visual hyperlinks among the images. It is assumed that if a user is viewing an image, other related(similar) images may also arouse the user's interest. Similar to PageRank algorithm, if image u is visually hyperlinked to image v, such hyperlink is treated as a vote of confidence, which means it is possible that the user will go from viewing u to viewing v. As a result, images related (similar) to the query image will have many other images pointing to them and will therefore be viewed often.

Here, let $G_S = (V, E_S)$ be an undirected graph with a set of vertices V and a set of edges E_S, where E_S is a subset of $V \times V$, $E_S = \{e = (u, v)|\text{u is similar to v}\}$. Likewise, let $G_A = (V, E_A)$ be an undirected graph with a set of vertices V and a set of edges E_A, where E_A is a subset of $V \times V$, $E_A = \{e = (u, v)|\text{u is adjacent to v}\}$. Then the final similarity/adjacency graph(SA graph for short below) for ranking is defined as $G = G_S \cup G_A$, where $E_S \cap E_A \neq \phi$.

In our case, we apply the above-mentioned random walk algorithm to the defined SA graph G. Thus, after iterative calculation, images roughly similar and roughly adjacent to query image(s) will be deemed important, since those are viewed as images composing the same landscape according to our hypothsis. Given n images, our proposed PanoramaRank (PR) is defined as follows:

$$PR = dS^* \times PR + (1 - d)p$$

where p_i is the initial value of V_i, and we refer to vertex V_i of an image img_i. d is a damping factor, and we set it to 0.8 in the following experiments empirically.

$$p_i = \begin{cases} \frac{1}{|SI|}, img_i \in SI \\ 0, img_i \notin SI \end{cases}$$

where SI is the image set whose element is selected by the user, and S^* is the column normalized adjacency matrix S, but here $S_{u,v}$ denotes the combination of visual similarity and adjacency between image u and v. Since both similarity and adjacency should be considered, the geometric mean and harmonic mean are employed to weigh edges between images in G.

$$S_{u,v} = Sim(u, v)^{\alpha} * Adj(u, v)^{1-\alpha}$$

or

$$S_{u,v} = \frac{1}{\frac{\alpha}{Sim(u,v)} + \frac{1-\alpha}{Adj(u,v)}}$$

where α is the weight factor.

6.5 Discovering Similar "Landscapes"

We assume each image attached with a tag that indicates its geographic location. This tag can be the name of a certain location, such as "Tokyo Tower", or a geo-tag containing both longitude and latitude information. According to this information, images are separated into different sets.

When at least one image is selected, our PanoramaRank will be applied limited in the set to which selected image(s) belong. Given an example in Fig.6.6. An image of Location A is chosen as the query image q. Therefore, PanoramaRank is employed in the image set of Location A. Only images whose score is above the threshold (empirically set to 0.075 in the following experiments), are preserved and referred to as a supplement that represents a certain landscape

together with the selected one(s). This step is also considered as query formulation, as user intent is about searching for a certain landscape but it is difficult to point out all images described that landscape. In a word, our proposed method is to discover a certain landscape in a location according to the given image example(s). In the above-mentioned example, after PanoramaRank is employed, some images, with a cross, are discarded. The rest ones, with some diagonals, are considered to form a representation of a landscape of Location A.

Meanwhile, similar images taken in different locations are retrieved, which is only based on image similarity mentioned in Sec.6.4.1 and graph-based ranking algorithm is not employed. For the top K (K=20) images, ones belonging to the same location are treated as an insufficient representation of a similar landscape to the original one. Images with the same geographic tag in every distinct location are regarded as the given image examples. Then PanoramaRank is applied in every set found before to find a whole landscape for every distinct location. Similar to what happened in the image set of Location A, PanoramaRank is then applied in the image set of Location B, C and so on. As a result, some images with a cross are also discarded and landscapes of Location B, C (shown in Fig.6.6) turn to be similar landscapes of that of Location A. For each image in the final result set, similarity between each one in the original landscape is calculated. The maximum (used in experiments), or minimum, or average similarity score is referrd to as its final score. We believe this score can reflect landscape similarity between images. As a result, a ranked image list will be returned. Moreover, each image in every landscape is compared with each one in the original landscape. The maximum (used in experiments), or minimum, or average similarity score is treated as landscape similarity between different landscapes. Therefore, a ranked landscape list will also be returned. In our example, the landscape of Location B is the most similar landscape, compared to the original landscape of Location A. The landscape of Location C is the second-most similar one.

6.6 Experiments and Evaluations

We created a small dataset having 6 categories, with 180 photos in total, shown in Fig.6.9. The performance of our proposed method is evaluated by

- comparing the resulting landscape image ranking list to the list obtained by three evaluators
- examining how well the expected landscape image sets are ranked at the top.

6.6.1 Similar “Landscape Image” Search

In detail, our evaluation is divided into two steps. In the first instance, ranking results, referred to the listed-up landscape images, are estimated in comparision with the original selected query image(s). Especially, here we only chose one image of a certain location as the query so that

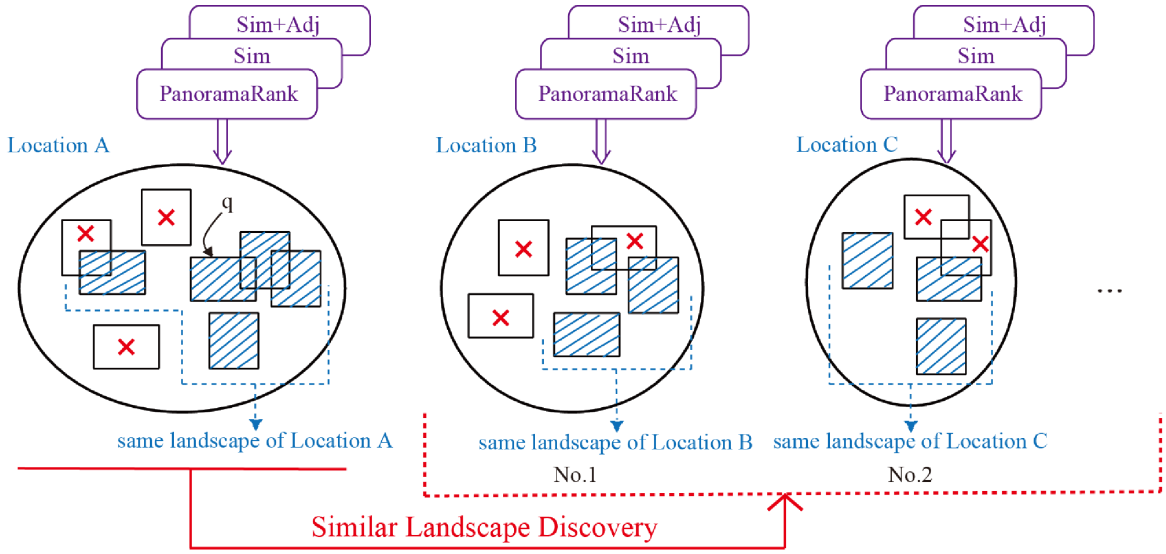


Figure 6.6: Illustration of our approach

all we need to do is to evaluate ranking results from the very begining to the end according to their relations with the query image. Relation here indicates whether this pair of images looks landscape-similar or not. Since it will fairly vary based on diverse personality, three evaluators are employed for estimation with the purpose of reducing bias. Besides, five-step evaluation is applied, then the average score for each image is calculated. As a result, we got the ideal ranked list for each query.

We find after combining image similarity and adjacency, performance will be improved as the ranked list is closer to the human judgment, above-mentioned ideal ranked list. Fig.6.7 shows our PanoramaRank performance for each query under harmonic mean of image similarity and adjacency, evaluated by nDCG. The final score of an image is the maximum similarity score among the similarity scores between this image and each one in the original landscape. When $\alpha = 0$, it means search only based on image adjacency, likewise, when $\alpha = 1$, it means search only based on image similarity. Queries such as the image of the Eikan-do temple, shown in Fig.6.9, are good examples to support the effectiveness of our proposed method. When only applying image similarity-based search ($\alpha = 1$), the nDCG score is 0.84, and only applying image adjacency-based search ($\alpha = 0$), the nDCG score is down to 0.75. However, after the combination of image similarity and adjacency, we can get a higher nDCG score, which is 0.91 when $\alpha = 0.3$ or $\alpha = 0.5$. From this, we can point out that it has been significantly improved with the purpose of discovering similar landscape images compared to search based on either image similarity or image adjacency. On the other hand, queries such as the image of Jiuzhaigou didn't go well. No matter adjacency is incorporated or not, there is little change in performance. The

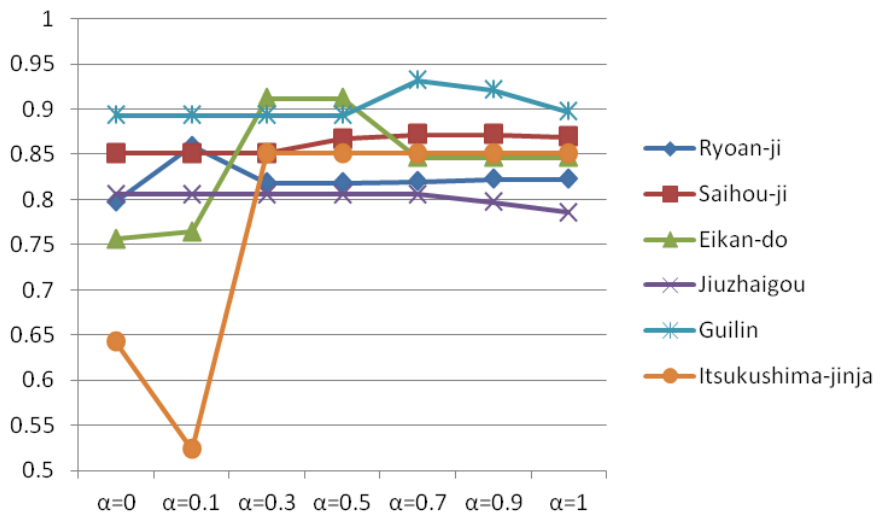


Figure 6.7: nDCG scores for each query when α is set to different values, using harmonic mean of image similarity and adjacency.

reason is that there are not so many images adjacent to each other in the image set of Jiuzhaigou as those in others. Thus, after taking harmonic mean of similarity and adjacency, the weight for each edge will turn smaller instead, which leads that discovered landscape images vary little so that final results hardly vary.

6.6.2 Similar "Landscape" Discovery

In the second step, evaluation is not based on a single image, but rather an intergral landscape image set. In Fig.6.9, every location in the same category is a candidate similar landscape of the rest, such as given an image of the rock garden in the Ryoan-ji temple, images about similar rock garden in the Daisen-in or Manshu-in are deemed to be similar landscape images. Thus, it can be said for rock garden landscape, the Daisen-in and Manshu-in are similar to the Ryoan-ji temple. We prepared two baseline methods as follows.

- **Sim**: Instead of applying our proposed PanoramaRank, only image similarity with the query image(s) is taken into account to find images belonging to the same landscape. Pay attention that graph-based ranking method is not employed here, which is the essential difference with the case when $\alpha = 1$. See Fig.6.6, Sim is employed to discover a landscape in a certain location, for example, in the image set of Location A, B, C and so forth.

- **Sim+Adj**: Instead of applying our proposed PanoramaRank, images belonging to the same landscape are found according to harmonic mean of image similarity and image adjacency ($\alpha = 0.5$) with the query image(s), shown in Fig.6.6.

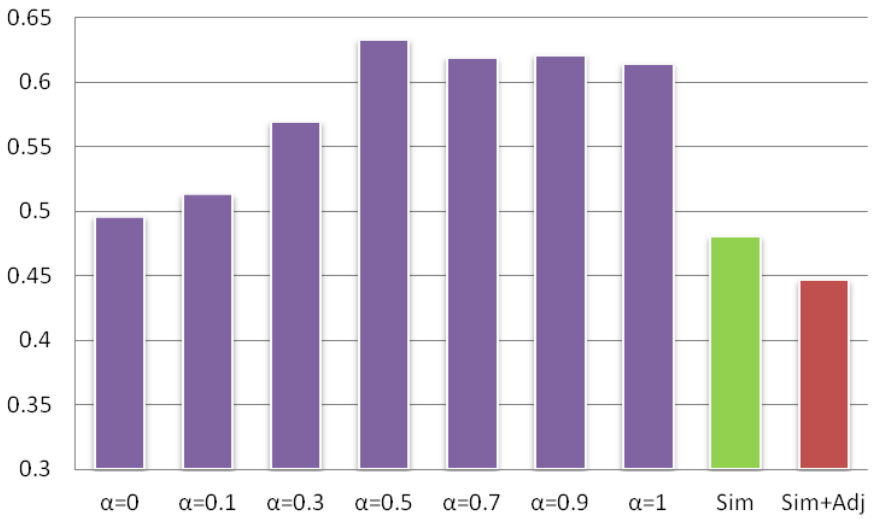


Figure 6.8: MAP scores when α is set to different values shown above, using harmonic mean of image similarity and adjacency.

Harmonic mean is taken and the performance is shown in Fig.6.8. We find that when $\alpha = 0.5$, 0.02 has been improved compared to search only based on image similarity in MAP, and more than 0.1 has been improved compared to search only based on image adjacency, and also ourperforms two baseline methods, whose MAP score is 0.48 and 0.44, respectively. This strongly illustrates that it is useful to incorporate image adjacency to discover similar landscapes.

6.7 Summary

In this paper, we proposed a method that can be used to find a landscape that is similar to the one whose image is given.

To discover a certain landscape, we applied a biased-VisualRank method, called Panorama-Rank. In addition, we incorporated image similarity and image adjacency to weigh each edge between two vertices in the SA graph constructed for ranking. Experiments are presented to show the effectiveness of our proposed method.

Our future works will be as follows. First, we will incorporate other image features into our scheme to improve the performance of our proposed method, and then employ a machine learning method to determine weight factors in related formulas. Second, for tags that are attached to photos, we will explore the relationships that exist among similar landscape image sets. Finally, we will extend the categories and generate a larger dataset to objectively evaluate our proposed method.

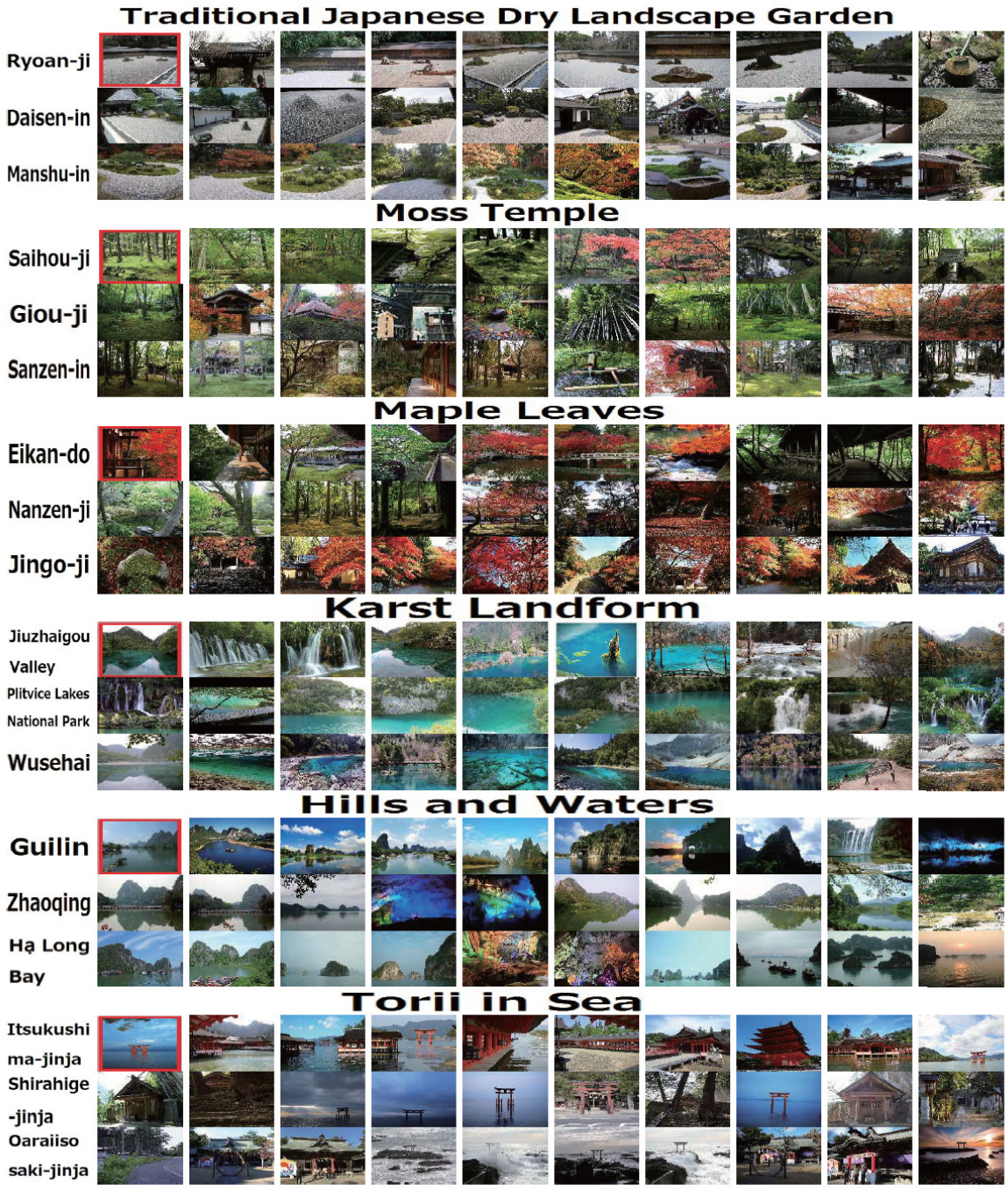


Figure 6.9: All images in our dataset. Actually, we assume six categories: traditional Japanese dry landscape garden, moss temple, maple leaves, karst landform, hills and waters, and torii in sea, respectively. For each category, we chose three different locations, with each containing ten images about that location. The leftmost in each row shows the exact location of the rest of the images in the same row. Meanwhile, category information is also shown at the top of the three locations belonging to the same category. Images with red frames are used as inputted query images in our experiments.

CHAPTER 7

CONCLUSIONS

7.1 Summary

This thesis discussed methodologies to complement the deficiencies out of reach to similarity-based search by taking into account coordinate relationships. We incorporated the coordinate relationships between tuples, documents, and images to acquire paraphrases from the Web, find similar but different documents and locations. Three research topics addressed in this thesis are summarized as follows:

- **Paraphrasing Sentential Queries based on Coordinate Relationships**
  The effectiveness of retrieval decreases with the increase in query length. We target at sentential queries, a type of long queries, and propose a method called *sentential query paraphrasing* for improving their retrieval performance, especially on recall. Briefly, given a sentential query, our method acquires paraphrases from the noisy Web and uses them to avoid returning no answers. We are motivated by the assumption that a relation can be represented either intensionally (referred to as *paraphrase templates*) or extensionally (referred to as *coordinate tuples*) and propose a mutual reinforcement algorithm based on it. Experimental results show that our method can acquire more paraphrases from the noisy Web. Besides, with the help of paraphrases, more Web pages can be retrieved, especially for those sentential queries that could not find any answers with its original expression.

- **Structuring Search Results based on Coordinate Relationships**
  We propose a method to structure search results of a user-given query to distinguish coordinate relationships from similarity relationships in the documents. We take into account documents that are mutually exclusive in semantics (called *coordinate documents*) and as-

sume that such documents should not be grouped into the same cluster. Correspondingly, we also consider documents that are mutually inclusive in semantics and assume that such documents should be grouped into the same cluster. Therefore, on the basis of these two types of constraints, documents are clustered in a manner closer to that of human cognition, e.g., news articles are organized according to events they describe. Experimental results show the effectiveness of our method and illustrate the importance of coordinate relationships in finding coordinate documents and structuring search results.

- **Panoramic Image Search based on Similarity and Adjacency**
  We introduce a new image search method, called *panoramic image search*, which is a trial in image retrieval, and show its application to similar landscape discovery. Briefly, taken an image or a few images of a place as the input, the output is images of other places that are similar to the query place from a certain perspective, referred to as "landscape". We believe that a single image cannot completely exhibit a landscape. Therefore, we also consider the surroundings. That is the physical surroundings around the spot captured in the single image. Consequently, a set of images is used to describe a landscape. In order to find such images, we consider not only similarity relationship between images but also adjacency relationship between images, and propose an image ranking algorithm called PanoramaRank. Experimental results show the effectiveness of our method to find similar landscapes.

Finally, technical and social contributions of my researches are summarized as follows:

Technical contributions

- developing a method to complement the deficiencies of similarity-based search.
- modeling the coordinate relationships between tuples, documents, and images for Web search.

Social contributions

- reducing the search difficulty of issuing suitable keyword queries and enabling people to search by natural language queries.
- supporting people understanding an unfamiliar topic.
- enabling people to find similar locations with the same atmosphere without knowledge of the locations.

7.2 Future Directions

There still remain several research questions that need to be further explored in future work. First, at the present, when considering the coordinate relationships between tuples, we only focus on binary relations. In other words, we only take two entities into account. In the future, we would like to extend the coordinate relationships between tuples to multiple relations so that there is no limit to the number of entities in the natural language queries. This brings new challenges: How should we model the multiple relations among entities? How can we detect the correspondence of entities from different tuples without doing dependency parsing? Are the current template extraction useful for this complicated case? Second, we would like to extend our approach of finding similar but different documents to all kinds of genres (not limited to news articles). This suggests us to design a general model to fit different types of documents. We believe these research directions will lead to the development of Web search based on coordinate relationships, which can make searches more efficient and effective.

BIBLIOGRAPHY

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of DL*, pages 85–94, 2000.

[2] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of SIGIR*, pages 37–45, 1998.

[3] N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Exploring reductions for long web queries. In *Proceedings of SIGIR*, pages 571–578, 2010.

[4] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 501–510, 2007.

[5] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of SDM*, volume 4, pages 333–344, 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[7] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proceedings of SIGIR*, pages 491–498, 2008.

[8] M. Bendersky and W. B. Croft. Analysis of long queries in a large scale search log. In *Proceedings of WSCD*, pages 8–14, 2009.

[9] R. Bhagat and D. Ravichandran. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL2008:HLT*, pages 674–682, 2008.

[10] D. T. Bollegala, Y. Matsuo, and M. Ishizuka. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of WWW*, pages 151–160, 2010.

[11] M. Brown and D. G. Lowe. Recognising panoramas. In *Proceedings of Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1218–1225, 2003.

[12] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, 1998.

[13] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision (ICCV2007)*, 2007.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[15] P. Denning, J. Horning, D. Parnas, and L. Weinstein. Wikipedia risks. *Communication of ACM*, 48(12):152–152, 2005.

[16] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[17] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *ARTIFICIAL INTELLIGENCE*, 165:91–134, 2005.

[18] A. Feng and J. Allan. Finding and linking incidents in news. In *Proceedings of CIKM*, pages 821–830, 2007.

[19] A. Feng and J. Allan. Incident threading for news passages. In *Proceedings of CIKM*, pages 1307–1316, 2009.

[20] F.Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26:354–359, 1983.

[21] G.N.Lance and W.T.Williams. A general theory of classificatory sorting strategies. *The Computer Journal*, 9:373–380, 1967.

[22] Z. S. Harris. Distributional structure. *Word*, 10:146–162, 1954.

[23] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, 15(4):784–796, 2003.

[24] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR*, pages 76–84, 1996.

[25] Hitwise Intelligence. Google received 72 percent of u.s. searches in january 2009. `http://image.exct.net/lib/fefc1774726706/d/1/SearchEngines_Jan09.pdf`, 2009.

[26] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[27] I. S. Idan, H. Tanev, and I. Dagan. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*, pages 41–48, 2004.

[28] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1877–1890, November 2008.

[29] R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *Proceedings of WWW*, pages 477–486, 2006.

[30] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of SIGIR*, pages 297–304, 2004.

[31] G. Kumaran and J. Allan. A case for shorter queries, and helping users create them. In *Proceedings of HLT*, pages 220–227, 2007.

[32] T. Lau and E. Horvitz. Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of UM*, pages 119–128, 1999.

[33] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proceedings of the 10th European Conference on Computer Vision (ECCV2008)*, pages 427 – 440, 2008.

[34] Z. Li, B. Wang, M. Li, and W.-Y. Ma. A probabilistic model for retrospective news event detection. In *Proceedings of SIGIR*, pages 106–113, 2005.

[35] Y. Liu, R. Jin, and A. K. Jain. Boostcluster: Boosting clustering by pairwise constraints. In *Proceedings of KDD*, pages 450–459, 2007.

[36] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[37] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[38] G. Mecca, S. Raunich, and A. Pappalardo. A new algorithm for clustering search results. *Data Knowl. Eng.*, 62(3):504–522, Sept. 2007.

[39] Q. Mei, H. Fang, and C. Zhai. A study of poisson query generation model for information retrieval. In *Proceedings of SIGIR*, pages 319–326, 2007.

[40] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411, 2004.

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshop*, 2013.

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[43] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[44] K. Nakamura, H. Ishii, and A. Tezuka. *Region and Landscape*. Kokin Shoin, Tokyo, 1991.

[45] R. Nallapati, A. Feng, F. Peng, and J. Allan. Event threading within news topics. In *Proceedings of CIKM*, pages 446–453, 2004.

[46] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[47] H. Ohshima, S. Oyama, and K. Tanaka. Searching coordinate terms with their context from the web. In *Proceedings of the 7th International Conference on Web Information Systems*, WISE'06, pages 40–47, 2006.

[48] C. F. Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21(8):1313–1325, 1995.

[49] S. Osinski, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of IIPWM*, pages 359–368, 2004.

[50] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, May 2005.

[51] M. Paşca and P. Dienes. Aligning needles in a haystack: Paraphrase acquisition across the web. In *Proceedings of IJCNLP*, pages 119–130, 2005.

[52] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 113–120, 2006.

[53] G. Pass and R. Z. J. Miler. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM international conference on Multimedia*, pages 65–73, 1996.

[54] N. Phan, P. Bailey, and R. Wilkinson. Understanding the relationship of information need specificity to search query length. In *Proceedings of SIGIR*, pages 709–710, 2007.

[55] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007)*, 2007.

[56] J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. In *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP 2008)*, pages 738–745, 2008.

[57] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[58] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.

[59] Y. Shinyama, S. Sekine, and K. Sudo. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*, pages 313–318, 2002.

[60] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL'44, pages 801–808, 2006.

[61] J. Tague-Sutcliffe. The pragmatics of information retrieval experimentation, revisited. *Inf. Process. Manage.*, 28(4):467–490, 1992.

[62] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of ICML*, pages 577–584, 2001.

[63] S. Wubben, A. van den Bosch, E. Krahmer, and E. Marsi. Clustering and matching headlines for automatic paraphrase acquisition. In *Proceedings of ENLG*, pages 122–125, 2009.

[64] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *Proceedings of SIGIR*, pages 475–482, 2008.

[65] Y. Yamamoto and K. Tanaka. Towards web search by sentence queries: Asking the web for query substitutions. In *Proceedings of DASFAA*, pages 83–92, 2011.

[66] Y. Yamamoto, T. Tezuka, A. Jatowt, and K. Tanaka. Supporting judgement of fact trustworthiness by considering temporal and sentimental aspects. In *Proceedings of WISE*, pages 206–220, 2008.

[67] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proceedings of SIGIR*, pages 28–36, 1998.

[68] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: Open information extraction on the web. In *Proceedings of NAACL*, pages 25–26, 2007.

[69] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. In *Proceedings of WWW*, pages 1361–1374, 1999.

[70] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of SIGIR*, pages 210–217, 2004.

[71] M. Zhao, H. Ohshima, and K. Tanaka. Sentential query rewriting via mutual reinforcement of paraphrase-coordinate relationships. In *Proceedings of iiWAS*, December 2015.

PUBLICATIONS

Journal Papers

1. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Paraphrasing Sentential Queries by Incorporating Coordinate Relationship"
   IPSJ Transactions on Databases, Vol.9, No.2, pp. 1–11, June 2016.

International Conference Papers

1. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Finding Coordinate Documents Based on Coordinate Relationships"
   In *Proceedings of the 18th International Conference on Asia-Pacific Digital Libraries (ICADL2016)*, submitted.

2. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Sentential Query Rewriting via Mutual Reinforcement of Paraphrase-Coordinate Relationships"
   In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS2015)*, pp. 499-508, December 2015.

3. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Panoramic Image Search by Similarity and Adjacency for Similar Landscape Discovery"
   In *Proceedings of the 13th international conference on Web Information Systems Engineering (WISE2012)*, pp. 284-297, November 2012.

International Workshop Papers

1. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Finding Paraphrase Facts Based on Coordinate Relationships"
   In *Proceedings of the 20th International Conference on Database Systems for Advanced Applications (DASFAA2015) International Workshops, SeCop*, pp. 135-151, April 2015.

2. Tomohiro Manabe, Kosetsu Tsukuda, Kazutoshi Umemoto, Yoshiyuki Shoji, Makato P. Kato, Takehiro Yamamoto, Meng Zhao, Soungwoong Yoon, Hiroaki Ohshima, Katsumi Tanaka
   "Information Extraction based Approach for the NTCIR-10 1CLICK-2 Task"
   In *Proceedings of the 10th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-10)*, pp.243-249, June 2013.

3. Makoto P. Kato, Meng Zhao, Kosetsu Tsukuda, Yoshiyuki Shoji, Takehiro Yamamoto, Hiroaki Ohshima, Katsumi Tanaka
   "Information Extraction based Approach for the NTCIR-9 1CLICK Task"
   In *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-9)*, pp. 202–207, December 2011.

Domestic Symposium and Workshops

1. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Finding Coordinate Relationships in Search Results"
   The 8th Forum on Data Engineering and Information Management (DEIM2016), A7-3, March 2016.

2. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Query Paraphrasing Towards Better Search by Incorporating Coordinate Relationship"
   The 7th Forum on Data Engineering and Information Management (DEIM2015), A3-3, March 2015.

3. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Paraphrase Acquisition from the Web Using Coordinate Information"
   The 6th International Workshop with Mentors on Databases, Web and Information Management for Young Researchers (iDB2014) (Closed workshop), July 2014.

4. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Discovery of Surrounding Fact Information based on Fact Adjacency Relationships"
   The 6th Forum on Data Engineering and Information Management (DEIM2014), A8-2, March 2014.

5. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Neighborhood-based Search: Incorporating Page Adjacency to Reach Serendipitous Information"
   The 5th International Workshop with Mentors on Databases, Web and Information Management for Young Researchers (iDB2013) (Closed workshop), July 2013.

6. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Text Landscape for Exploratory Search"
   The 5th Forum on Data Engineering and Information Management (DEIM2013), A3-4, March 2013.

7. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Similar Landscape Discovery based on Image Similarity and Adjacency"
   The 4th International Workshop with Mentors on Databases, Web and Information Management for Young Researchers (iDB2012) (Closed workshop), July 2012.

8. Meng Zhao, Hiroaki Ohshima, Katsumi Tanaka
   "Image Search based on Similarity and Adjacency"
   The 4th Forum on Data Engineering and Information Management (DEIM2012), B5-3, March 2012. (in Japanese)