Semantic image similarity based on deep knowledge for effective image retrieval

Yuanxi Li
Hong Kong Baptist University

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation
Li, Yuanxi, "Semantic image similarity based on deep knowledge for effective image retrieval" (2014). Open Access Theses and Dissertations. 99.
https://repository.hkbu.edu.hk/etd_oa/99

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at HKBU Institutional Repository. It has been accepted for inclusion in Open Access Theses and Dissertations by an authorized administrator of HKBU Institutional Repository. For more information, please contact repository@hkbu.edu.hk.
Semantic Image Similarity Based on Deep Knowledge for Effective Image Retrieval

LI Yuanxi

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Principal Supervisor: Prof. LEUNG Clement H. C.

Hong Kong Baptist University

August 2014
Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at Hong Kong Baptist University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications. Some contents of this thesis have already been included in my publications in Journals, Proceedings of International Conference, Book Chapters and etc..

Signature:__________________

Date: August 2014
A flourishing World Wide Web dramatically increases the amount of images uploaded and shared, and exploring them is an interesting and challenging task. While content-based image retrieval, which is based on the low level features extracted from images, has grown relatively mature, human users are more interested in the semantic concepts behind or inside the images. Search that is based solely on the low level features would not be able to satisfy users requirements and not effective enough. In order to measure the semantic similarity among images and increase the accuracy of Web image retrieval, it is necessary to dig the deep concept and semantic meaning of the image as well as to overcome the semantic gap.

By exploiting the context of Web images, knowledge base and ontology-based similarities, through the analysis of user behavior of image similarity evaluation, we established a set of formulas which allows efficient and accurate semantic similarity measurement of images. When jointly applied with ontology-based query expansion approaches and an adaptive image search engine for deep knowledge indexing, they are able to produce a new level of meaningful automatic image annotation, from which semantic image search may be performed. Besides, the semantic concept can be automatically enriched in MPEG-7 Structured Image Annotation approach.

The system is evaluated quantitatively using more than thousands of Web images with associated human tags with user subjective test. Experimental results indicate that this approach is able to deliver highly competent performance, attaining good
precision efficiency. This approach enables an advanced degree of semantic richness to be automatically associated with images and efficient image concept similarity measurement which could previously only be performed manually.

**Keywords:** Image Index, Image Retrieval, Semantic Similarity, Relevance Feedback, Knowledge Base, Ontology, Query Expansion, MPEG-7 . . .
Acknowledgements

In the first place I would like to record gratitude to my principal supervisor, Prof. Clement Ho Cheung LEUNG, for giving me the opportunity to work with his supervision. His feedback often gave very insightful perspectives on the aspects to the research. More importantly, the mentoring I have received from him extends well beyond academic research. I am thankful for his continued guidance and encouragement. I am really proud of being a student of Prof. LEUNG, who is such an excellent and knowledgeable professor. I also want to express sincerely thanks to visiting professor Prof. Alfredo MILANI for his kind guidance, great inspirations and generous help.

I would like to thank Dr. Li CHEN for serving as my co-supervisor. Also, many thanks to my research committee, Prof. Jiming LIU and Dr. William CHEUNG, who took time out of their busy schedules to provide valuable feedback and their constructive suggestions on the thesis. Besides, I would like to express my sincere gratitude to Prof. Pong Chi YUEN, Dr. Jian FENG and Prof. Yiu Wing LEUNG for their guidance and great supports.

In the course of my doctoral study, I have had the opportunity to collaborate with brilliant fellow students, senior brother apprentice: Roger Chun Fan WONG, senior sister apprentice: Alice Wing Sze CHAN, colleagues and friends: Valentina FRANZONI, Jie DENG, Sheung Wai CHAN, Paolo MENGONI, Marco MENCACCI, Yanrong CHEN and among others. I would like to thank them for their contributions,
valuable discussions and continued collaboration.

I want to give my special thanks to supporting staff from general office and technical team of Computer Science Department, as well as other academic/non-academic units of Hong Kong Baptist University. We cannot complete the research smoothly without the resources and supports from our University.

I would also like to express my gratefulness to my parents and grandparents for being there to share thoughts with me and guide me through the various stages. Their unquestioned and uninterrupted encouragement over the years has been a constant source of sustenance. All their love is treasured deeply in my heart.

Last but not least, I want to thank God for giving me wisdom and guiding me all the way in the research life, especially in this thesis writing.
Table of Contents

Declaration i

Abstract ii

Acknowledgements iv

Table of Contents vi

List of Tables xi

List of Figures xiii

Chapter 1 Introduction 1

1.1 Current Challenges in Image Retrieval 3

1.2 Motivations and Significance 8

1.3 Outline of this Thesis 10

Chapter 2 Literature Review 13

2.1 Content-Based Image Retrieval 14

2.1.1 Feature Extraction 18

2.1.2 Relevance Feedback 23

2.2 Concept-Based Image Retrieval 29

2.3 Summary 35
## Chapter 3  Semantic-Based Concept Similarity of Image

3.1 Image Semantic Concept Similarity Measurements .......................... 37
   3.1.1 WordNet Distance .................................................. 37
   3.1.2 Wikipedia Distance ................................................ 41
   3.1.3 Flickr Distance ....................................................... 43
   3.1.4 Confidence ........................................................... 46
   3.1.5 Normalized Google Distance ....................................... 46
   3.1.6 PMI Similarity ....................................................... 48
   3.1.7 PMING Similarity ................................................... 49

3.2 Semantic Space and Context-based Image Similarity ....................... 52

3.3 Summary .......................................................................... 63

## Chapter 4  Ontological Query Expansion Based on Similarity

4.1 Introduction ..................................................................... 65

4.2 Semantic Query Expansion Using Ontology Architectures ............. 70
   4.2.1 SIM-CYC Distance .................................................... 70
   4.2.2 Framework of CYC Based Query Expansion Image Search Model 71
   4.2.3 CYC-WordNet Based Query Expansion System Work ........... 72
   4.2.4 User Iterative Selection and Pruning .............................. 73

4.3 Optimization of Ontological Expansion .................................... 76

4.4 Extended ROC Analysis .................................................... 78

4.5 Summary .......................................................................... 80

## Chapter 5  An Adaptive Image Search Engine for Deep Knowledge Indexing

5.1 Relevance Feedback .......................................................... 83

5.2 Evolutionary Self-Organizing Search Engine ............................. 84
   5.2.1 Adaptive Search Engine Architecture ............................ 87
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.2</td>
<td>Query Processing</td>
<td>88</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Feedback Processing</td>
<td>95</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Introducing New Terms and New Objects</td>
<td>97</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Term Insertion</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>Summary</td>
<td>101</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Image Retrieval Based on MPEG-7 Description Structure</td>
<td>103</td>
</tr>
<tr>
<td>6.1</td>
<td>Analyze and Match MPEG-7 Descriptors</td>
<td>104</td>
</tr>
<tr>
<td>6.2</td>
<td>Exploiting the Relationship between Image Metadata and Semantic Content</td>
<td>107</td>
</tr>
<tr>
<td>6.3</td>
<td>Architecture for image Semantic Information Extraction</td>
<td>111</td>
</tr>
<tr>
<td>6.4</td>
<td>Automatically Structural Casting to MPEG-7 Representation and Index Building</td>
<td>113</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Scheme of Information Extraction and Standardization</td>
<td>113</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Real user participative interface</td>
<td>116</td>
</tr>
<tr>
<td>6.5</td>
<td>Automatic Tags Generation for MPEG-7 Descriptions</td>
<td>118</td>
</tr>
<tr>
<td>6.6</td>
<td>Summary</td>
<td>121</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Experiments and Evaluation</td>
<td>122</td>
</tr>
<tr>
<td>7.1</td>
<td>Experiments of Semantic-based Image similarity</td>
<td>123</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Semantic Image Similarity Measurements</td>
<td>123</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Comparison of Different Concept Distance Measurements</td>
<td>127</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Context-based Group Similarity</td>
<td>128</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Summary</td>
<td>139</td>
</tr>
<tr>
<td>7.2</td>
<td>Experiments of Knowledge-based Query Expansion</td>
<td>140</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Experimental Results</td>
<td>140</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Summary</td>
<td>143</td>
</tr>
<tr>
<td>7.3</td>
<td>Experiments of Evolutionary Self-Organizing Search Engine</td>
<td>144</td>
</tr>
</tbody>
</table>
7.3.1 Simulated User Test Model .................................. 145
7.3.2 Real User Test Design ........................................ 149
7.3.3 Performance Evaluation ....................................... 151
7.3.4 Test Results and Discussions ................................. 152
7.3.5 Summary .......................................................... 164
7.4 Experiments of Image Retrieval Accuracy Based on MPEG-7 Descriptors ............................................. 165
  7.4.1 Experimental Results ......................................... 165
  7.4.2 Summary .......................................................... 166
7.5 Summary .............................................................. 170

Chapter 8 Summary and Future Research Directions 171
  8.1 Summary and Contributions ..................................... 172
  8.2 Future Research Directions ..................................... 174
  8.3 Conclusions .......................................................... 176

Bibliography 177

Appendices 207
  A.1 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: Conference) - Table 1 ........................................ 207
  A.2 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: NGD) - Table 2 ........................................ 207
  A.3 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: PMI) - Table 3 ........................................ 207
  A.4 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: Confidence) - Table 4 ........................................ 207
A.5 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: NGD) - Table 5 . . . . . . . . . . . . . . . . . 207
A.6 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: PMI) - Table 6 . . . . . . . . . . . . . . . . . 207

Publications 214

Curriculum Vitae 217
## List of Tables

3.1 Links that connect the terms (directed edges) in Wikipedia. .......... 41

5.1 Illustration of the relevance indexing function $I$ ................... 86

7.1 Performance of semantic relatedness measures of WD algorithms with their standard deviations. ................................. 125

7.2 Performance of semantic relatedness measures of WLVM algorithms with their standard deviations. ......................... 126

7.3 A Subset of Experimental Results Confidence – Max ................ 136

7.4 A Subset of Experimental Results NGD – Max ...................... 137

7.5 A Subset of Experimental Results PMI – Max ...................... 138

7.6 Sample Plain Queries and Precision Comparison of Image Search with Plain Query and Plain Query + Expanded Query. .......... 141

7.7 Real user tests: experimental setting .............................. 150

7.8 Results Comparison between Two Sets of Runs ..................... 154

7.9 Global Relative Relevance ........................................ 160

7.10 Coverage and noise in real user tests ............................. 163

7.11 Image search precision rate comparison between unaided approach and proposed approach ................................ 168

1 A Subset of Experimental Results Conference – Avg .................. 208

2 A Subset of Experimental Results NGD – Avg ...................... 209
3 A Subset of Experimental Results $PMI - Avg$ . . . . . . . . . . . . . . . . . . 210
4 A Subset of Experimental Results $Confidence - Min$ . . . . . . . . . . . 211
5 A Subset of Experimental Results $NGD - Min$ . . . . . . . . . . . . . . . . . . 212
6 A Subset of Experimental Results $PMI - Min$ . . . . . . . . . . . . . . . . . . 213
List of Figures

1.1 Semantic gap on human perception and Content Based Image Retrieval. 5

2.1 General framework of Content Based Image Retrieval. ............. 14
2.2 General process of Content-based image retrieval. ................. 16
2.3 Example of JSEG segmentation. ..................................... 17
2.4 Example of image retrieval by average color and dominant color: (a) original region; (b) average color; (c) dominant color. ................. 19
2.5 Example of image retrieval by texture feature. .................... 21
2.6 Example of image retrieval by Shape feature: 20 images for the guitar class. ......................................................... 22
2.7 Example of Content Based Image Retrieval (CBIR). .............. 24
2.8 Methodology of the relevance feedback based CBIR by using SVM. 26
2.9 General framework of Semantic Based Image Retrieval. ............ 29
2.10 Example of generation of the visual dictionary. ..................... 31
2.11 The indexing structure is represented by a pyramid. ............... 32
2.12 Example of Bridging the gap: the semantic extraction and represen-
tation of images. ......................................................... 34

3.1 Example of concept expansion based on WordNet Distance. ........ 38
3.2 Example demonstrate the way WordNet Distance defines two words are similar if nearby in thesaurus hierarchy. ......................... 40
6.2 Image Category according to Image Metadata Each column shows the different categories of Micro, Night Scenes, Indoor Activities, Day Scenes, Portraits and Outdoor Activities .......................... 108

6.3 An example of Image distribution in 3D space of some Flickr images . 109

6.4 Sample rules of Automatic Semantic Annotation ......................... 110

6.5 Conceptual aspects descriptions ........................................... 111

6.6 Overall system workflow ..................................................... 112

6.7 The prototype of the ontology-based and MPEG-7 SAD image retrieval architecture .......................................................... 115

6.8 Automatic Generation of Semantic MPEG-7 Descriptions for Image from Metadata ............................................................. 119

6.9 An example of automatic generation of semantic MPEG-7 descriptions for image from metadata and low level features ............... 120

7.1 Performance of relatedness rates between concept downtown and other concepts computed by the WordNet distance and image searching precision .......................................................... 124

7.2 Performance of relatedness rates between concept downtown and other concepts computed by the Wikipedia distance and image searching precision .......................................................... 124

7.3 Correctly classified pairs with respect to the human perception evaluation model .............................................................. 128

7.4 Concept Distance Comparison .................................................. 129

7.5 The user interface of image similarity test experiment. .................. 129

7.6 Sample set 1 of image pairs similarity experiment results - comparing user score and proposed Group Similarity. ............................. 131

7.7 Sample set 2 of image pairs similarity experiment results - compares user score and proposed Group Similarity. ............................. 132
7.8 Sample set 1 of image pairs similarity experiment results - a graph comparing user score and proposed Group Similarity. .......................... 134
7.9 Sample set 2 of image pairs similarity experiment results - a graph compares user score and proposed Group Similarity. .......................... 134
7.10 Precision Comparison of Image Search with Plain Query and Plain Query + Expanded Query .......................... 142
7.11 User Model Testing Architecture .............................................. 145
7.12 Parameters Tuning on Weight Parameters $c_1$, $c_2$, and $c_3$. ................ 155
7.13 Parameters Tuning on Weight Parameters $c_4$. .......................... 156
7.14 Increasing the Total Number of Objects. ................................. 157
7.15 Comparison of Elitism Strategies. .......................................... 159
7.16 Relative Relevance for Real User Tests. .................................. 161
7.17 Image search precision rate comparison between searching by original tags and searching by MPEG-7 Descriptors .................. 167
Chapter 1

Introduction

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others [47]. Diverse types of Web images are increasingly prevalent, and their effective sharing is often inhibited by limitations in their search and discovery mechanisms [110], which are particularly restrictive for images that do not lend themselves to automatic processing or indexing. Web image search is often limited by inaccurate or inadequate tags, and many raw images are constantly uploaded with little meaningful clue of their semantic contents, thus limiting their search and discovery. While content-based image retrieval, which is based on the low-level features extracted from images, has grown relatively mature, human users are more interested in the semantic concepts behind or inside the images. Search that is based solely on the low level features would not be able to satisfy users’ requirements. In order to increase the accuracy of Web image retrieval, it is necessary to enrich the concept index and semantic meaning of the images as well as to overcome the semantic gap. With semantic search, meaning can exist at different levels. In general, three levels of visual information retrieval may be distinguished [58]:

1. Level 1 The lowest level is based on primitive features such as colour, texture,
shape, spatial location of image elements, or a combination of these.

2. Level 2 This level comprises retrieval by derived attributes or semantic content and corresponds to Panofsky’s pre-iconographic level of picture description. Search requests on this level include the retrieval of objects of a given type or class as well as the retrieval of individual objects or persons.

3. Level 3 This level comprises retrieval by abstract attributes and includes search requests for named events or types of activity, corresponding with iconography, and those for pictures with emotional or symbolic significance, corresponding with iconology.

In this Work, we mainly focus on the automatic semantic enrichment, to enhance the accuracy and efficiency of image index and retrieval. The following methods and mechanism have been proposed and discussed in this work: semantic space and context-based image similarity measurements, ontological query expansion based on similarity mechanism, an adaptive image search engine for deep knowledge, as well as MPEG-7 descriptor semantic enrichment.
1.1 Current Challenges in Image Retrieval

The number of Web images is increasing at a rapid rate, and searching them semantically presents a significant challenge. Many raw images are constantly uploaded with little meaningful direct annotation of semantic content, limiting their capacity to be searched and discovered. Unlike in a traditional database, information in an image database is in visual form, which requires more space for storage, is highly unstructured and needs state-of-the-art algorithms to determine its semantic content.

As Web images tend to grow into unwieldy proportions, their retrieval systems must be able to handle multimedia annotation and retrieval on a Web scale with high efficiency and accuracy. With the exception of systems that can identify or detect music, words, faces, irises, smiles, people, pedestrians, or cars, matching is not usually directed toward object semantics. Recent research studies show a large disparity between user needs and technological capabilities.

Vast numbers of Web images are continuously added with few meaningful direct annotations of semantic content, limiting their search and discovery. While some sites encourage tags or keywords to be included manually, such is far from universal and applies to only a small proportion of images on the Web. Research in image search has reflected the dichotomy inherent in the semantic gap [182, 72, 212], and is divided between two main categories: concept-based image retrieval and content-based image retrieval. The former focuses on retrieval by image objects and high-level concepts, while the latter focuses on the low-level visual features of the image. In order to determine image objects, the image often has to be segmented into parts. Common approaches to image segmentation include segmentation by region and segmentation by image objects. Segmentation by region aims to separate image parts into different regions sharing common properties. These methods compute a general
similarity between images based on statistical image properties, \([116, 6, 191]\) and common examples of such properties are texture and colour where these methods are found to be robust and efficient. Some systems use colour, texture, and shape \([184, 39, 67, 231, 46, 211, 128]\) as attributes and apply them for entire image characterization, and some studies include users in a search loop with a relevance feedback mechanism to adapt the search parameters based on user feedback, while various relevance feedback models and ranking methods for Web search have been developed \([37, 126, 1]\). Segmentation by object, on the other hand, is widely regarded as a hard problem, which if successful, will be able to replicate and perform the object recognition function of the human vision system; although progress on this front has been slow, some advances in this direction have nevertheless been made \([214, 168, 178]\). In \([117, 145, 124]\), semantic annotation of images combined with a region-based image decomposition is used, which aims to extract semantic properties of images based on the spatial distribution of colour and texture properties. Such techniques have drawbacks, primarily due to their weak disambiguation and limited robustness in relation to object characterization. However, an advantage of using low-level features is that, unlike high-level concepts, they do not incur any indexing cost as they can be extracted by automatic algorithms. In contrast, direct extraction of high-level semantic content automatically is beyond the capability of current technology. Although there has been some effort in trying to relate low-level features and regions to higher level perception, these tend to be for isolated words, and they also require substantial training samples and statistical considerations \([117, 56, 14, 13, 18, 91]\). These methods, however, have limited success in determining semantic contents in broad image domains. There are some approaches which exploit surrounding and associated texts in order to correlate and mine these with the content of accompanying images \([193, 22, 65, 96, 96, 215, 220]\). Text-based retrieval is often limited to the processing of tags, and no attempt is made to extract a thematic description of
the picture. Some research focuses on implicit image annotations which involves an implicit, rather than an explicit, indexing scheme and, in consequence, augments the original indexes with additional concepts that are related to the query [8, 152, 120], necessitating the use of some probabilistic weighting schemes.

With semantic search, meaning can exist at different levels. In [61], it has been suggested that image search requests may be categorized into unique and non-unique queries. Unique queries are those which can be satisfied by the retrieval of a unique person, object or event while non-unique ones cannot. Panofsky [160], based on picture identification in fine arts, categorizes images based on who, what, where and when.

Some of the most important notions in image retrieval are keywords, terms or concepts. Terms are used both from humans to describe their information need and from the system as a way to represent images. However, current image search systems, such as Yahoo! and Google, use some surrounding text description provided by humans in order to infer semantics. These techniques ignore the meaningful image features which can be extracted via image processing analysis. Further, as...
most of these images come without explicit semantic tags and those inferred from surrounding text are often unreliable, these models have limited effectiveness at present and they need further development and refinement. There lies the need to automatically infer semantics from raw images to facilitate semantics-based searches.

The effectiveness of image retrieval depends on meaningful indexing; the key problem of image retrieval is to organize them based on semantics. The word 'semantic', which frequently appears in the content of this chapter, is the linguistic interpretation of multimedia objects, such as images and video clips, and is closely associated with the nature and meaning of the underlying objects.

From the viewpoint of image annotation and retrieval models, semantics can be textual descriptions attached to images [121]. Moreover, it can be high-level concepts describing the scene or relationships between images in a group that have a particular meaning for a user [205]. The process of assigning semantics to images is referred as image annotation, indexing or tagging in general for multimedia objects. The approach of manual annotation is followed by a number of commercial platforms and collaborative communities of multimedia, such as YouTube, Flickr and Pbase. However, the manual annotation of images is a laborious and error-prone task, as annotations can be biased towards the annotator’s perspective, and it is difficult to define a strategy to ensure annotative consistency. This implicit connection is usually called as the 'semantic gap'. The concept of the 'semantic gap' is proposed and formalized by Smeulders et al. [182]:

“The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.”

Smeulders et al

They also conclude that:
“A critical point in the advancement of content-based retrieval is the semantic gap, where the meaning of an image is rarely self-evident. The aim of content-based retrieval systems must be to provide maximum support in bridging the semantic gap between the simplicity of available visual features and the richness of the user semantics.”

Smeulders et al

Despite continuous research efforts in developing and exploring new models, the gap between the expressive power of image features and semantic concepts is still a fundamental barrier. In simpler terms, the capability of current technology has no ability to fully understand the semantics of multimedia objects and direct extraction of semantic content automatically is not possible. Although there has been some effort in trying to relate low-level features and regions to higher-level perception, the success of such systems is limited. Although there has been some effort in trying to relate low-level features and regions to higher level perception, these are limited to isolated words. They also require substantial training samples and statistical considerations in order to, partially or completely, bridge of these gaps.
1.2 Motivations and Significance

A flourishing World Wide Web dramatically increases the amount of images uploaded, and exploring them from Big Data is an interesting and challenging task. To explore multi-levels of visual information and the semantic concepts behind the images.

Current image search systems, such as Bing and Google, use some surrounding text description provided by humans in order to infer semantics [63]. These techniques ignore the meaningful image features which can be extracted via image processing analysis. Besides, manual annotation is impossible for a large dynamic database and not accurate to express the content and concept of an image as “A picture is worth a thousand words” (proverb that contributed by Confucius, the Chinese thinker and social philosopher, 2500 years ago).

Content Based Image Retrieval (CBIR) allows exploring digital image by their visual content by low level feature extraction, such as color, texture, and shape information. Feature extraction may be done from region or an entire image [50, 140]. The similarities of images can be calculated based on the similarities between features. However, as for the semantic gap, the semantic concept hidden in the images could not be explored only by CBIR.

To harness the considerable user judgment and human evaluation of media objects in the course of users’ normal interaction with the search system, which collectively over the community and cumulatively over time can amount to highly substantial efforts [102]. The basic idea of this work is that information derived by the queries issued by a community of users can be used to drive and adapt the future systems answers to similar queries. In this work, we shall focus on a single-click feedback model, i.e. we assume that the only form of feedback the user can provide is to click on a single object link, or not clicking at all when the answer lists are
not sufficiently relevant to the query. While the analysis of a more specific session model is beyond the scope of the paper, the single click feedback model is quite universally applicable to most search engine interfaces. The increasing relevance of privacy concerns and legal issues makes user profile and client side monitoring of user task increasingly difficult. A single click anonymous user feedback model is then a quite realistic hypothesis.

In order to overcome the disadvantages of current techniques, and improve the semantic concepts exploration in images, this work enables (1) measuring the image semantic similarity based on context; (2) using knowledge base and ontology to enrich high-level concepts of images; (3) involving user’s relevance feedback (RF) into retrieval loop so that the image searching engine will make evolution continuously and automatically; (4) generating MPEF-7 descriptions to further enrich and standardize the semantic concept of images to support higher level indexing.
1.3 Outline of this Thesis

- **Chapter 2 - Literature Review:** We review the recent related techniques of image retrieval and indexing by exploring the key theoretical and empirical contributions of these techniques related to low level feature extraction of Content Bases Image Retrieval (CBIR), user interactive relevance feedback, Concept Based Image Retrieval. The advantages and drawbacks of current techniques have also been compared. This also highlight significant challenges and open issues in effective image indexing scheme.

- **Chapter 3 - Semantic-Based Concept Similarity of Image:** We exploit several similarity/aproximity measurements and different concept ontology - WordNet Distance, Wikipedia Distance, Flickr Distance, Confidence, Normalized Google Distance (NGD) and Pointwise Mutual Information (PMI). Based one the comparison, a new similarity type: PUMING Similarity, which combine the advantages of individual concept similarity measurements, has been proposed. Furthermore, in order to calculate the similarity of images, we proposed a scheme to discover Context-based Image Similarity. Comparing with Content Based Image Retrieval (CBIR), which measure the image content similarities by low level features, the proposed Context-based Semantic Image Similarity outperformed CBIR in measuring the deep concept similarity and relationship of images.

- **Chapter 4 - Ontological Query Expansion Based on Similarity:** We discuss methods that combine the advantages of the image semantic similarity measurements and knowledge based scheme, propose a ontology-based query expansion method for effective image indexing and retrieval. We study several semantic concept-based query expansion and re-ranking scheme and compare different ontology-based expansion methods in image search and retrieval. To
improve the query expansion efficiency and accuracy, we employ the CYC knowledge base to generate the expansion candidate concepts, then filter and rank the expansion results by calculating concept similarities using the Semantic Relatedness Metrics. Using our knowledge-based query expansion in image retrieval, the efficiency and accuracy has been significantly improved.

- **Chapter 5 - An Adaptive Image Search Engine for Deep Knowledge Indexing:** We developed an adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time. In the course of normal usage, the underlying index structure and contents are gradually and dynamically re-organized. This approach realizes a form of machine learning which can be put into relationship with reinforcement learning [93] and genetic algorithms based approaches [69] where the feedback from the environment, here represented by the user community, is used to evolve the system behavior. A major difference from typical supervised learning scheme is the absence of separate training phase. Here the learning phase is a continuous interactive process since the object repository, the community of users, and user judgment of relevance of objects to search terms dynamically evolve over the time. For this reason an adaptive algorithm has been devised to reflect the continuous change in the term/objects index. The adaptive engine exploits some typical evolutionary techniques such as mutation, random tournament and elitism. Systematic simulation experiments on a large number of objects and search queries have been held to tune parameters and to evaluate adaptive ability, performance and scalability of the approach. In addition, experiments on real data with actual users have been performed.

- **Chapter 6 - Image Retrieval Based on MPEG-7:** We present a com-
prehensive fully automated approach based on the analysis of image metadata in conjunction with image analysis techniques. In this chapter, we propose an automatic image semantic enrichment approach, which could inject deeper semantics to MPEG-7 Description for effective image search.

- **Chapter 7 - Experiments and Evaluation:** We illustrate the experiment design, results of four aspects in this work: semantic-based concept similarity of image, ontological query expansion based on similarity, an adaptive image search engine for deep knowledge, and image retrieval based on MPEG-7. The evaluation and discussion have also been presented.

- **Chapter 8 - Conclusion and Future Research Directions:** We conclude the overall results and contributions from the thesis presented in previous chapters. The chapter ends with some pointers to the future work regarding chapters 3 to 7 and an overview of the future of semantic image similarity based on deep knowledge for effective image retrieval.
Chapter 2

Literature Review

This chapter is to review the recent related techniques of image retrieval and indexing by exploring the key theoretical and empirical contributions of these techniques related to low level feature extraction of Content Bases Image Retrieval (CBIR), user interactive relevance feedback, Concept Based Image Retrieval. The advantages and drawbacks of current techniques have also been compared. This also highlight significant challenges and open issues in effective image indexing scheme.
2.1 Content-Based Image Retrieval

Content-based image retrieval (CBIR) \cite{48, 27, 35, 151, 118, 147, 226, 54, 78} is a technique used for retrieving similar images from an image database. CBIR operates on retrieving stored images from a collection by comparing features automatically extracted from the images themselves \cite{110}. The commonest current CBIR systems, whether commercial or experimental, operate at level 1. A typical system allows users to formulate queries by submitting an example of the type of image being sought, though some offer alternatives such as selection from a palette or sketch input. The system then identifies those stored images whose feature or signature values match those of the query most closely, and displays thumbnails of these images on the screen. The most challenging aspect of CBIR is to bridge the gap between low-level feature layout and high-level semantic concepts.

The general framework of content-based image retrieval is shown in Fig. 2.1 \cite{213}.

Colour, texture and shape features have been used for describing image content. Different CBIR systems have adopted different techniques. Few of the techniques have used global colour and texture features \cite{54, 78, 162, 189} whereas few others
have used local colour and texture features [27, 35, 151, 118]. The latter approach segments the image into regions based on colour and texture features. The regions are close to human perception and are used as the basic building blocks for feature computation and similarity measurement. These systems are called region based image retrieval (RBIR) systems and have proven to be more efficient in terms of retrieval performance. Few of the region based retrieval systems, e.g., [27], compare images based on individual region-to-region similarity. These systems provide users with rich options to extract regions of interest. But precise image segmentation has still been an open area of research. It is hard to find segmentation algorithms that conform to the human perception. For example, a horse may be segmented into a single region by an algorithm and the same algorithm might segment horse in another image into three regions. These segmentation issues hinder the user from specifying regions of interest especially in images without distinct objects. To ensure robustness against such inaccurate segmentations, the integrated region matching (IRM) algorithm [118] proposes an image-to-image similarity combining all the regions between the images. In this approach, every region is assigned significance worth its size in the image. A region is allowed to participate more than once in the matching process till its significance is met with. The significance of a region plays an important role in the image matching process. In either type of systems, segmentation close to human perception of objects is far from reality because the segmentation is based on colour and texture. The problems of over segmentation or under segmentation will hamper the shape analysis process. The object shape has to be handled in an integral way in order to be close to human perception. Shape feature has been extensively used for retrieval systems [68, 88]. Fig. 2.2 is the general process of Content-based image retrieval.

Image retrieval based on visually significant points [135, 148] is reported in literature. In [133], local colour and texture features are computed on a window of regular
Figure 2.2: General process of Content-based image retrieval.
Figure 2.3: Example of JSEG segmentation.

geometrical shape surrounding the corner points. General purpose corner detectors [73] are also used for this purpose. In [12], fuzzy features are used to capture the shape information. Shape signatures are computed from blurred images and global invariant moments are computed as shape features. The retrieval performance is shown to be better than few of the RBIR systems such as those in [35, 118, 170].

An example image segmentation [132] is shown in Fig. 2.3.

The studies mentioned above clearly indicate that, in CBIR, local features play a significant role in determining the similarity of images along with the shape information of the objects. Precise segmentation is not only difficult to achieve but is also not so critical in object shape determination. A windowed search over location and scale is shown more effective in object-based image retrieval than methods based on inaccurate segmentation [80]. The objective of this paper is to develop a technique which captures local colour and texture descriptors in a coarse segmentation framework of grids, and has a shape descriptor in terms of invariant moments computed on the edge image. The image is partitioned into equal sized non-overlapping tiles. The features computed on these tiles serve as local descriptors of colour and texture. In [79] it is shown that features drawn from conditional co-occurrence histograms using image and its complement in RGB colour space perform significantly better. These features serve as local descriptor of colour and texture in the proposed method. The
grid framework is extended across resolutions so as to capture different image details within the same sized tiles. An integrated matching procedure based on adjacency matrix of a bipartite graph between the image tiles is provided, similar to the one discussed in [118], yielding image similarity. A two level grid framework is used for colour and texture analysis. Gradient Vector Flow (GVF) fields [221] are used to compute the edge image, which will capture the object shape information. GVF fields give excellent results in determining the object boundaries irrespective of the concavities involved. Invariant moments are used to serve as shape features. The combination of these features forms a robust feature set in retrieving applications. The experimental results are compared with [35, 118, 12, 170] and are found to be encouraging.

2.1.1 Feature Extraction

Colour Feature

Several methods for retrieving images on the basis of colour similarity have been described in the literature [58], but most are variations on the same basic idea. Each image added to the collection is analysed to compute a colour histogram which shows the proportion of pixels of each colour within the image. The colour histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each colour (75% olive green and 25% red, for example), or submit an example image from which a colour histogram is calculated. Either way, the matching process then retrieves those images whose colour histograms match those of the query most closely. The matching technique most commonly used, histogram intersection, was first developed by Swain and [194]. Variants of this technique are now used in a high proportion of current CBIR systems. Methods of improving on Swain and Ballard’s original technique include the use of cumulative
Figure 2.4: Example of image retrieval by average color and dominant color: (a) original region; (b) average color; (c) dominant color.

colour histograms [188], combining histogram intersection with some element of spatial matching [188], and the use of region-based colour querying [26]. The results from some of these systems can look quite impressive.

Fig. 2.4 shows the example of image retrieval by average color and dominant color [132].

**Texture Feature**

The ability to retrieve images on the basis of texture similarity may not seem very useful. But the ability to match on texture similarity can often be useful in distinguishing between areas of images with similar colour (such as sky and sea, or leaves and grass). A variety of techniques has been used for measuring texture similarity; the best-established rely on comparing values of what are known as second-order statistics calculated from query and stored images. Essentially, these calculate the relative brightness of selected pairs of pixels from each image. From these it is possible to calculate measures of image texture such as the degree of contrast, coarseness, directionality and regularity [196], or periodicity, directionality and randomness [129]. Alternative methods of texture analysis for retrieval include the use of Gabor filters [141] and fractals [95]. Texture queries can be formulated in a similar manner to colour queries, by selecting examples of desired textures from a palette, or by supplying an example query image. The system then retrieves images with texture measures most similar in value to the query. A recent extension of the tech-
nique is the texture thesaurus developed by [139], which retrieves textured regions in images on the basis of similarity to automatically-derived codewords representing important classes of texture within the collection.

Fig. 2.5 is an example of image retrieval by texture feature [171].

Shape Feature

The ability to retrieve by shape is perhaps the most obvious requirement at the primitive level. Unlike texture, shape is a fairly well-defined concept - and there is considerable evidence that natural objects are primarily recognized by their shape [17]. A number of features characteristic of object shape (but independent of size or orientation) are computed for every object identified within each stored image. Queries are then answered by computing the same set of features for the query image, and retrieving those stored images whose features most closely match those of the query. Two main types of shape feature are commonly used - global features such as aspect ratio, circularity and moment invariants [89] and local features such as sets of consecutive boundary segments [143]. Alternative methods proposed for shape matching have included elastic deformation of templates [162, 49], comparison of directional histograms of edges extracted from the image [88, 3], and shocks, skeletal representations of object shape that can be compared using graph matching techniques [98, 202]. Queries to shape retrieval systems are formulated either by identifying an example image to act as the query, or as a user-drawn sketch [77, 28].

As shown in Fig. 2.6, images are retrieved by Shape feature [154].

Shape matching of three-dimensional objects is a more challenging task - particularly where only a single 2-D view of the object in question is available. While no general solution to this problem is possible, some useful inroads have been made into the problem of identifying at least some instances of a given object from different viewpoints. One approach has been to build up a set of plausible 3-D models from
Figure 2.5: Example of image retrieval by texture feature.
Figure 2.6: Example of image retrieval by Shape feature: 20 images for the guitar class.

the available 2-D image, and match them with other models in the database [34]. Another is to generate a series of alternative 2-D views of each database object, each of which is matched with the query image [51]. Related research issues in this area include defining 3-D shape similarity measures [181], and providing a means for users to formulate 3-D shape queries [81].

Retrieval by Other Types of Primitive Feature

One of the oldest-established means of accessing pictorial data is retrieval by its position within an image. Accessing data by spatial location is an essential aspect of geographical information systems, and efficient methods to achieve this have been around for many years (e.g. [40, 169]). Similar techniques have been applied to image collections, allowing users to search for images containing objects in defined spatial relationships with each other [29, 30]. Improved algorithms for spatial retrieval are still being proposed [70]. Spatial indexing is seldom useful on its own, though it has proved effective in combination with other cues such as colour [188, 183] and shape
Several other types of image feature have been proposed as a basis for CBIR. Most of these rely on complex transformations of pixel intensities which have no obvious counterpart in any human description of an image. Most such techniques aim to extract features which reflect some aspect of image similarity which a human subject can perceive, even if he or she finds it difficult to describe. The most well-researched technique of this kind uses the wavelet transform to model an image at several different resolutions. Promising retrieval results have been reported by matching wavelet features computed from query and stored images \cite{85, 123}. Another method giving interesting results is retrieval by appearance. Two versions of this method have been developed, one for whole-image matching and one for matching selected parts of an image. The part-image technique involves filtering the image with Gaussian derivatives at multiple scales \cite{165}, and then computing differential invariants; the whole-image technique uses distributions of local curvature and phase \cite{164}. The advantage of all these techniques is that they can describe an image at varying levels of detail (useful in natural scenes where the objects of interest may appear in a variety of guises), and avoid the need to segment the image into regions of interest before shape descriptors can be computed. Despite recent advances in techniques for image segmentation \cite{23}, this remains a troublesome problem.

Fig.2.7 gives and example of state-of-the-art Content Based Image Retrieval\cite{138}.

\subsection{Relevance Feedback}

Relevance feedback \cite{102}, originally developed for information retrieval is an online learning technique used to improve the effectiveness of the information retrieval systems. Since its introduction into image retrieval around the mid 1990s, it has attracted tremendous attention in the Content-Based Image Retrieval (CBIR) community. A particularly noteworthy work is \cite{173}, which develops a relevance feed-
Figure 2.7: Example of Content Based Image Retrieval (CBIR).
back based interactive retrieval approach focusing on content-based aspects such as color, texture and shape, and relevance feedback has since been shown to provide dramatic performance improvement [229, 38]. Recent works that apply relevance feedback techniques to image retrieval systems include [7], which makes use of kernel machines and selective sampling that adaptively modify the similarity measures, and [197], which develops kernel convex machines and exploits the idea that negative samples for relevance feedback form several sub-clusters while positive ones group only in one cluster. In [198], an orthogonal complement component analysis method was used that captures the concepts in all positive samples and demonstrates favourable comparison with those of linear and kernel principal component analysis method. Here, we apply the user-feedback media ranking in a different way. Generally, most of the user-feedback media ranking techniques are applied to the CBIR only. CBIR focuses on retrieval based on the visual feature of the image (e.g. color, texture). However, our approach supports the searching of multimedia resources such as images, videos, and audios, and it has the advantage that it is able to focus on arbitrarily higher-level human properties and perceptive details which are not extractable by machines.

The methodology of the relevance feedback based CBIR by using SVM [177] is shown in Fig. 2.8.

Particularly significant studies of relevance feedback as applied to cross-media retrieval are given in [225, 230]. In [230], a uniform cross-media correlation graph (UCCG) is constructed for evaluating the correlation among media objects of different modalities, with the media objects represented as vertices and their correlations as weights of the weighted edges. Unlike the relevance feedback used in [230] which adopts a heuristic approach that modifies the edge weights of an underlying graph model, the algorithm used in [225] adopts a statistical approach and refines the cross-media indexing space (CMIS) directly, where the mechanism of long-term relevance
Figure 2.8: Methodology of the relevance feedback based CBIR by using SVM.
feedback is formulated as a minimization problem which includes making use of data periodically extracted off-line from a log-file for producing overall improvement in cross-media retrieval performance. Unlike their approach, which makes use of the concept of a multimedia document (MMD) consisting of a set of heterogeneous multimedia objects of the same semantics, our approach focuses on each media object separately, and our index update mechanism is carried out continuously online in the course of normal usage rather than done periodically offline. In addition, with respect to experimentation, we base our measurements mostly on concepts rather than content-based features and semantic categories. Rocchio’s similarity-based relevance feedback algorithm [167, 36] is one of the classical query reformation methods in information retrieval. It is essentially an adaptive supervised learning algorithm from examples. Rocchio’s formula is also used in [228] in the context of image retrieval in which the query point vector moving strategy for relevance feedback is based. Van Uden in [208] comments that the Rocchio algorithm aims at finding a request that best suits the user’s information need (i.e. user input query) by using relevance feedback. The ranking of this algorithm relies heavily on the iterative explicit user feedback. On the other hand, our approach not only takes into account explicit user feedback, but also implicit feedback incorporating both positive and negative feedback forms. Lin’s web image retrieval re-ranking process [127] based on relevance model utilizes global information from the image’s HTML document to evaluate the relevance of the image. This approach seems promising when text-based HTML documents associated with the web images are available. However, this approach can only perform well when the web images are rich in related text-based HTML information. Our proposed re-ranking approach focuses on each type of media separately and does not rely on any extra embedded information, and the media resources can be retrieved even though there is no supplementary information on the object. This work focuses on the user community feedback as the main
source for ranking relevance; on the other hand classical search methods, such as query expansion [59] and pseudo relevance feedback [20], can be easily integrated in the search engine. These methods have proven to be particularly effective when additional textual information (e.g. annotations, abstracts, dialogue transcripts extracted from videos etc.) is available on the multimedia objects [222, 203, 172].
2.2 Concept-Based Image Retrieval

Neither a single features nor a combination of multiple visual features could fully capture high-level concept of images [213]. Besides, due to the performance of image retrieval based on low level features are not fully satisfactory, there is a need for retrieval based on semantic meaning by trying to extract the cognitive concept of a human to map the low level image features to high-level concept (semantic gap).

In addition, representing image content with semantic terms allows users to access images through text query which is more intuitive, easier and preferred by the users to express their mind compared with using images. For example, users’ queries may be 'Find an image of sunset rather than 'find me an image contains red and yellow colours'.

The general framework of concept-based image retrieval is shown in Fig. 2.9 [213].

Although CBIR and visual similarity techniques, discussed in the previous section, are important and widely applicable in image retrieval and annotation domain, these studies indicate that the accuracy is subject to refinement. An advantage of using low-level features is that, unlike high-level concepts, they do not incur any indexing cost as they can be extracted by automatic algorithms. Some recent studies [33, 41, 223] find that text descriptors, such as time, location, events, objects,
formats, and topical terms are most helpful to users. In contrast, direct extraction of high-level semantic content automatically is beyond the capability of current technology. Although there has been some effort in trying to relate low-level features and regions to higher-level perception [56, 55], these are limited to isolated words, and they also require substantial training samples and statistical considerations [14, 19, 18, 91, 117, 158, 24].

The use of image-based analysis techniques is still not very accurate or robust and, after years of research, their retrieval performance is still far from users’ expectations. Furthermore, the paramount challenge of semantic image retrieval and annotation, the semantic gaps, are not fully presentable in low-level features. It is inappropriate to fill and bridge the semantic gap by only the image pixel, but the effort should be made together with high-level semantic content. Thus concept-based image retrieval is still preferable in general commercial products.

The tagging process involves interpretation of the visual information given some context, either the context of the image or the context of the annotation or retrieval. For example, in the case that the image itself has no special meaning, we may still tag the place or the event of the image if it is available. The scenes of images, like indoor or outdoor scenes, are useful for the user-retrieval process as well.

Fig. 2.10 gives an example of generation of the visual dictionary [227].

Some studies [86, 76, 92, 45, 130, 82] unify concepts from the literature of diverse fields such as cognitive psychology, library sciences, art, and the more recent concept-based image retrieval. Then they present multiple-level structures for visual and non-visual information.

In [86], they begin by defining the distinction between visual content and non-visual content:

1. The visual content of an image corresponds to what is directly perceived when the image is observed (i.e., descriptors stimulated directly by the visual content
2. The non-visual content corresponds to information that is closely related to the image, but that is not explicitly given by its appearance.

This is followed by the definition between percept and concept:

1. The percept refers to what our senses perceive - in the visual system these are light patterns. These patterns of light produce the perception of different elements such as texture and colour. No interpretation process takes place when referring to the percept - no knowledge is required.

2. A concept, on the other hand, refers to an abstract or generic idea generalised from particular instances. As such, it implies the use of background or prior knowledge and an inherent interpretation of what is perceived. Concepts can be
very abstract in the sense that they depend on an individual’s knowledge and interpretation - this tends to be very subjective.

Their visual structure (Fig. 2.11) [87] contains ten levels: the first four refer to syntax, and the remaining six refer to semantics. In addition, levels one to four are directly related to percept, and levels five through ten to visual concept. Some of these division may not be strict, but these are highly related to what the user is searching for and how they are finding images in a dataset.

Similarly, another study [84] defines a knowledge-based type of abstraction hierarchy with a three-layered image model to integrate the image representation. Meanwhile, some studies propose the thesaurus-based search model [32] to fulfill semantic image search.

Although, keywords may be content-dependent, they are appropriate to express descriptive metadata [182]. In [14], two ways to link tags or concepts with images are suggested:

1. to predict annotations of entire images using all information present; we refer to this task as annotation.

2. to associate particular words with particular image substructures, that is, to infer correspondence.
Concept-based retrieval or semantic similarity is often limited to the processing of tags, and generally no attempt is made to extract description of the picture and ignore the meaningful image features which can be extracted via image processing analysis. More importantly, it does not normally address the interaction of text and image processing in deriving semantic descriptions of a picture.

In Fig.2.12, an example of semantic extraction and representation of images illustrates how to bridge the semantic gap. [213]

In current commercial image platforms, such as Flickr, concept-based retrieval approaches are always incorporated with user-friendly user interface with suggestions of some similar keywords and terms that entered by user. Furthermore, such models may extract and recommend keywords and terms by extracting raw data from camera metadata.

Some research focuses on implicit image annotation which involves an implicit, rather than an explicit, indexing scheme and, in consequence, augments the original indexes with additional concepts that are related to the query [9, 101, 216, 153], necessitating the use of some probabilistic weighing schemes.
Figure 2.12: Example of Bridging the gap: the semantic extraction and representation of images.
2.3 Summary

In this chapter, techniques and schemes have been summarized in two main image retrieval systems - Content Based Image Retrieval and Concept Base Image Retrieval. Based on the drawbacks of the existing systems, we proposed mainly four directions to improve efficiency of image indexing and retrieval. The proposed schemes will be introduced in the following chapters.
Chapter 3

Semantic-Based Concept Similarity of Image

To measure the semantic similarity among image concepts, we exploit several similarity/aproximity measurements and different concept ontology - WordNet Distance, Wikipedia Distance, Flickr Distance, Confidence, Normalized Google Distance (NGD) and Pointwise Mutual Information (PMI). Based on the comparison, a new similarity type: PUMING Similarity, which combine the advantages of individual concept similarity measurements, has been proposed.

Furthermore, in order to evaluate image similarity/proximity in terms of the associated groups of concepts, we proposed a scheme to discover Context-based Semantic Image Similarity. Comparing with Content Based Image Retrieval (CBIR), which measures the image content similarities by low level features, the proposed Context-based Image Similarity outperformed CBIR in measuring the deep concept similarity and relationship of images.
3.1 Image Semantic Concept Similarity Measurements

In order to defined the group concept proximity measures, the characteristics of basic image Semantic Concept Proximity Measures has been studied. Here we introduce several basic concept proximity measures first.

3.1.1 WordNet Distance

WordNet [149], is one of these applications of semantic lexicon for the English language and is a general knowledge base and commonsense reasoning engine [104].

The purpose of the work is both to produce a combination dictionary-and-thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

WordNet is Taxonomy/lexicon based and reflects universal knowledge - it is built by human experts. The limitation of WordNet is that it is only for nouns and verbs in WordNet itself and it is not dynamically updated.

WordNet Similarity is a freely available software package that makes it possible to measure the semantic similarity and relatedness between a pair of concepts (or synsets). It provides six measures of similarity, and three measures of relatedness, all of which are based on the lexical database WordNet. These measures are implemented as Perl modules which take as input two concepts, and return a numeric value that represents the degree to which they are similar or related [161].

Several methods for determining semantic similarity between terms have been proposed in the literature and most of them have been tested on WordNet5. Similarity measures apply only for nouns and verbs in WordNet (taxonomic properties for adverbs and adjectives do not exist). Semantic similarity methods are classified into four main categories:
Figure 3.1: Example of concept expansion based on WordNet Distance.

- Edge Counting Methods
- Information Content Methods
- Feature based Methods
- Hybrid methods
- Single Ontology
- Cross Ontology

For example [104], by using WordNet, ‘downtown’ has been expanded to ‘business district’, ‘commercial district’, ‘city centre’ and ‘city district’, while ‘city district’ has been expanded to ‘road’, ‘building’, ‘architecture’, ‘highway’ and ‘hotel’. The semantic knowledge is hierarchically expansible from the query terms and concepts and knowledge can be expanded extensively. The more extensive and complete such hierarchies, the greater the scope for rich semantic manipulation. Fig. 3.1 visualize the example of concept expansion using WordNet.
Recent research [21] on the topic in computational linguistics has emphasized the perspective of semantic relatedness of two lexemes in a lexical resource, or its inverse, semantic distance.

The first line of research [166], which brings together ontology and corpus, tries to define the similarity between two concepts $c_1$ and $c_2$ lexicalized in WordNet, named WordNet Distance (WD). It indicates by the information content of the concepts that subsume them in the taxonomy. Formally, define:

$$sim(c_1, c_2) = \max (c \in S(c_1, c_2)) [-\log p(c)]$$

where $p(c) = \frac{\sum_{n \in \text{words}(c)} \text{count}(n)}{N}$ and $N$ is the total number of nouns observed. And $S(c_1, c_2)$ is the set of concepts that subsume both $c_1$ and $c_2$. Moreover, if the taxonomy has a unique top node, then its probability is 1. In practice, we often measure word similarity rather than concept similarity. Using $s(w)$ to represent the set of concepts in the taxonomy that are senses of word $w$, define

$$wsim(c_1, c_2) = \max (c \in S(c_1, c_2)) [sim(c_1, c_2)]$$

where $c_1$ ranges over $s(w_1)$ and $c_2$ ranges over $s(w_2)$.

It defines two words as similar if near to one another in the thesaurus hierarchy. For example, refer to Fig. 3.2, ‘entity’ can expand to ‘inanimate-object’. Then ‘inanimate-object’ can expand to both ‘natural-object’ followed by ‘geological-formation’. It then expands to both ‘natural-elevation’ and ‘shore’. The former can expand to ‘hill’ while the latter expands to ‘coast’.

Beside the original work, researchers [125] propose a similarity measure between arbitrary objects. It uses the same elements but in a different fashion.

Resnik defined the similarity between two concepts lexicalized in WordNet to be the information content of their lowest super-ordinate (most specific common subsumer) $lso(c_1, c_2)$:
Figure 3.2: Example demonstrate the way WordNet Distance defines two words are similar if nearby in thesaurus hierarchy.
Table 3.1: Links that connect the terms (directed edges) in Wikipedia.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Arches</th>
<th>Memory</th>
<th>Max degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,345,655</td>
<td>300,481,664</td>
<td>264 gigabyte</td>
<td>813,407</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
    d(c_1, c_2) &= \frac{2 \times \log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)} \\
    &\quad \text{(3.1.3)}
\end{align*}
\]

where \( p(c) \) is the probability of encountering an instance of a synset \( c \) in some specific corpus. \( P(c) = \frac{\sum_{n \in \text{words}(c)} \text{count}(n)}{N} \) and \( N \) is the total number of nouns observed. And \( lso(c_1, c_2) \) is the lowest super-ordinate of the two observed concepts.

### 3.1.2 Wikipedia Distance

Wikipedia is the world’s largest collaboratively edited source of encyclopedic knowledge [104]. In spite of its utility, its contents are barely machine-interpretable. Each article in Wikipedia describes a single topic; its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus. Meanwhile, each article must belong to at least one category of Wikipedia. Hyperlinks between articles keep many of the same semantic relations as defined. Table 3.1 [146] shows the links that connect the terms (directed edges) in Wikipedia.

WikiRelate [190] was the first to compute measures of semantic relatedness using Wikipedia. Their approach took familiar techniques that had previously been applied to WordNet and modified them to suit Wikipedia. Implementation of WikiRelate follows hierarchical category structure of Wikipedia.

Wikipedia Distance reflects the relationships as seen by the user community and it dynamically changes as links and nodes are changed by the users’ collaborative effort. However, it only can apply to knowledge base organized as networks of
The Wikipedia Link Vector Model (WLVM) [150] uses Wikipedia to provide structured world knowledge about the terms of interest. Their approaches are using the hyperlink structure of Wikipedia rather than its category hierarchy or textual content [210]. Fig. 3.3 illustrates an example of semantic nodes and links [144].

Probability of WLVM is defined by the total number of links to the target article over the total number of articles. Thus if $t$ is the total number of articles within Wikipedia, then the weighted value $w$ for the link $a \rightarrow b$ is:

$$w(a \rightarrow b) = |a \rightarrow b| \times \log \left( \sum_{x=1}^{t} \frac{t}{|x \rightarrow b|} \right)$$  \hspace{1cm} (3.1.4)$$

$$w(a \rightarrow b) = |a \rightarrow b| \times \log \left( \sum_{x=1}^{t} \frac{t}{|x \rightarrow b|} \right)$$  \hspace{1cm} (3.1.5)$$

where $a$ and $b$ denotes the search terms.

For example, calculate the similarity between Israel and Jerusalem, one would consider only the nation and its capital city. The commonness of a sense is defined by the number of times the term is used to link to it: e.g. 95% of Israel anchors link to the nation, 2% to the football team, 1% to the ancient kingdom, and a mere 0.1%
to the Ohio township. According to Equation 3.1.5, WLVM value of both terms Israel and Jerusalem is 0.994, which is completely reasonable.

### 3.1.3 Flickr Distance

Flickr distance (FD) [218] is another model for measuring the relationship between semantic concepts in visual domain [107]. For each concept, a collection of images are obtained from Flickr, based on which the improved latent topic-based visual language model is built to capture the visual characteristic of this concept. The Flickr distance between concepts $c_1$ and $c_2$ can be measured by the square root of Jensen-Shannon divergence [119, 175, 176] between the corresponding visual language models as follows:

$$D(C_1, C_2) = \sqrt{\sum_{i=1}^{K} \sum_{j=1}^{k} \frac{D_{JS}(P_z c_1 | P_z c_2)}{K^2}}$$  \hspace{1cm} (3.1.6)

where

$$D_{JS} (P_z c_1 | P_z c_2) = \frac{1}{2} D_{KL} (P_z c_1 | M) + \frac{1}{2} D_{KL} (P_z c_2 | M)$$  \hspace{1cm} (3.1.7)

$K$ is the total number of latent topics, which is determined by experiment. $P_z c_1$ and $P_z c_2$ are the trigram distributions under latent topic $z_i c_1$ and $z_j c_2$ respectively, with $M$ representing the mean of $P_z c_1$ and $P_z c_2$.

Fig. 3.4 shown the calculation process [218] of Flickr Distance, while Fig. 3.5 gives and example of two different concept models [218] with spatial information or without spatial information.

In addition to harnessing relatedness from these architectures to optimize search performance, we shall also experimentally develop a unified metric $\rho$ to measure dynamic relatedness of two concepts $x$ and $y$. 

43
Figure 3.4: Example of Flickr Distance Calculation.

Figure 3.5: Example of different concept models - VLM v.s. BOW.
\[ p(x, y) = \alpha_1 NGD(x, y) + \alpha_2 w(x, y) + \alpha_3 FD(x, y) \]  \hspace{1cm} (3.1.8)

where the coefficients \( \alpha_i > 0 \), and \( \sum \alpha_i = 1 \), by focusing on the top-ranked concepts for the different architectures.

Traditional features based proximity measure (text, images) are resources consuming and infeasible for large web repositories. Therefore, we need the Search/Retrieval Engine based proximity, which is using a search/retrieval engine query results as a source of probability/statistics about co-occurrence of term/objects. The proximity/similarity measures will be based on frequency/probabilities as shown in Fig. 3.6. The relationship between Frequency \( f(x) \) and Probability \( P(x) \) is:

\[ P(x) = f(x)/N \]

The Advantages of the proximity/similarity measures based on frequency/probabilities are:

- Can be applied to any retrieval engine
- Occurrence/Co-occurrence of words, terms, tags, users, objects etc.
- Reflect the user community current state of believes
- Dynamically change as results change
In the following sections, Confidence, Normalized Google Distance, PMI and PMING Distance will be introduced.

### 3.1.4 Confidence

The definition of Confidence can be shown in the following equation:

$$P(X \rightarrow Y) = P(Y|X) = P(X \wedge Y)/P(X)$$

Confidence is an unsymmetrical measurements and Values between [0,1]. The closer relationship of two concepts, the larger the Confidence value.

Fig. 3.7 illustrated an example of the confidence of a pair of concepts: $Conf(football, player)$ and $Conf(player, football)$.

### 3.1.5 Normalized Google Distance

Normalized Google distance (NGD) is proposed [60] to quantify the extent of the relationship between two concepts by their correlation in the search results from a
Figure 3.8: Example of distance calculation of three concepts using NGD.

search engine (e.g. Google) when querying both concepts [107], with

$$NGD (x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}$$ (3.1.10)

Where $f(x)$ and $f(y)$ are the numbers of the Web pages returned by Google search engine when typing $x$ and $y$ as the search term respectively, with $f(x, y)$ denotes the number of pages containing both $x$ and $y$.

An example that calculating the distance of three concepts using NGD is illustrated in Fig. 3.8.

Normalized Google Distance is symmetric measurement and can be applied to any search engine.

The concept of the NGD is derived from the Kolmogorov complexity [108], information distance and Kraft inequality, with some algebraic manipulation which involve compression-based steps that lead to normalized distances. E. g. Normalized Compression Distance (NCD), which is a family of compression functions parameterized by a given data compressor, and its limiting case Normalized Information Distance (NID), where the number of bits in the shortest code that can be decompressed by a general purpose computable decompressor [42].

The NGD assumes that a priori all the web pages are equiprobable and tries to
indirectly use the probabilistic information of the events of occurrence of a word to determine a prefix code, because the events can overlap and hence the summed probability of the Google code exceeds 1. The Google code, which represents the shortest expected prefix-code word length of an event, is then approximated in the NGD.

The parameter $M$, which represents the total number of pages indexed by the search engine, which often is not known, or is varying in a short time lapse, can be approximated using any value reasonably greater than any $f(x)$.

Even if NGD is a good proximity measure, and disregarding its name, it is neither a metric, nor a distance. In fact, the NGD does not respect the property of triangular inequality.

### 3.1.6 PMI Similarity

*Pointwise Mutual Information (PMI)* \[142, 204\] is a point-to-point measure of association, which represents how much the actual probability of a particular co-occurrence of events differs from what we would expect it to be on the basis of the probabilities of the individual events and the assumption of independence. Even though PMI may be negative or positive, its expected outcome over all joint events (i.e., PMI) is positive \[108\].

Two concepts are more likely to co-occur in a common, shared context and less likely in an unshared one. In a shared context, both have an increased probability of appearing but in an unshared one, as in Fig. 3.9 \[71\], one is more likely but the other not. The number of co-occurrences also depends on the sizes of the two concepts. PMI fits the normalized measure of co-occurrences to represent their similarity very well.

PMI is based on Shannon information entropy and used both in statistics and in information theory. It has been used both in statistics and in information theory.
Figure 3.9: Unshared contexts v.s. common contexts between concepts two sets.

PMI between two particular events $w_1$ and $w_2$, in this case the occurrence of particular words in Web-based text pages, is defined as follows:

$$PMI(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)}$$ (3.1.11)

This quantity is zero if $w_1$ and $w_2$ are independent, positive if they are positively correlated, and negative if they are negatively correlated. It's a symmetric measurement.

On particularly low frequency data, PMI does not provide reliable results. Since PMI is a ratio of the probability of $w_1$, $w_2$ together and $w_1$, $w_2$ separately, in the case of perfect dependence, PMI will be 0.

PMI is a bad measure of dependence, since the dependency score is related to the frequency of individual words. PMI could not always be suitable when the aim is to compare information on different pairs of words.

### 3.1.7 PMING Similarity

A new collaborative proximity measure was presented in 2012, named PMING Distance [66], which uses the statistical information returned by search engines as a
natural source of semantics. The general idea is to use the number of occurrence of a term or a set of terms. An approximation of the number of occurrences of a term is usually provided by most search engines which return, for example, the number of results/documents which contain the query search terms. A distance based on such number of occurrences is collaborative and dynamic, since it is based on automatic indexing of documents provided in the web by the users, and it can change dynamically over the time as new documents are indexed and old documents disappear from the web.

The PMING Distance is based on Pointwise Mutual Information (PMI) and Normalized Google Distance (NGD), which have been found to be among the measures which have a good performance in capturing the semantic information for clustering, ranking and extracting meaningful relations among concepts.

In order to understand the PMING we briefly introduce PMI, NGD and some notational conventions.

Let’s define:

- \( f(x) \) as the number of occurrence of a search term \( x \) in the query results, where \( x \) is either a single term \( t_1 \) or a group of terms;

- \( f(x,y) \) as the occurrence of two search term \( x \) and \( y \) in the query results, if \( x = t_1 \) and \( y = t_2 \) then \( f(x,y) \) will count the occurrences of \( t_1 \) AND \( t_2 \);

- \( P(x) = f(x)/N \). It summarizes the frequency based approach to probability, i.e. states that in the following formula probability \( P \) can be computed from frequency \( f \) and vice versa whenever the total \( N \) is known or can be approximated.

**PMING Distance** consists of NGD and PMI locally normalized [66], with a correction factor of weight \( \rho \), which depends on the differential of NGD and PMI. In
PMING the two component measures are locally normalized, so that their weighted combination is based on the context of evaluation, such as on the Vector Space Model (VSM).

The PMING distance of two terms $x \& y$ in a context $W$ is defined, as a function

$$PMING: W \times W \rightarrow [0, 1]$$

$$PMING(x, y) = \beta \log \frac{f(x,y)M}{\mu_1 f(x)F(y)} + (1 - \beta) \frac{\max[\log f(x), \log f(y)] - \log f(x,y)}{\mu_2 (\log M - \min[\log f(x), \log f(y)])}$$

(3.1.12)

where:

$\beta$ is a parameter to balance the weight of components ($\beta=0.3$ in our tests);

$\mu_1 \& \mu_2$ are constant values that depends on the context of evaluation, defined as:

$$\mu_1 = \max_{w_1, w_2 \in W} PMI (w_1, w_2)$$

$$\mu_2 = \max_{w_1, w_2 \in W} NGD (w_1, w_2)$$

PMING was found to incorporates the advantages of both PMI and NGD [108], outperforming state-of-the-art proximity measures in modeling contexts, modeling human perception and clustering of semantic associations, regardless of the search engine/repository. The comparison between the ranking induced by PMING and different proximity measures in contexts is shown in the chapter of experimental results.
3.2 Semantic Space and Context-based Image Similarity

Content Based Image retrieval (CBIR) enables satisfactory similarity measurements of low level features. However, the semantic similarity of deep relationship among objects could not explored by CBIR even by other state-of-the-art techniques in Concept Based Image Retrieval. An example in Fig. 3.10 illustrates the different similarity recognition of human and computer.

Hence, we propose a Context-based Group Similarity to measure deep semantic similarity of images. The proposed algorithm measures the image similarities with semantic proximity based one user provided concept clouds. Image semantic concept clouds include any semantic concept associated to or extracted from images. Typical sources for semantic concepts are tags, comments, descriptors, categories, or text surrounding the image.
Figure 3.11: Group Similarity Definition
Let’s give the definition of Semantic Group Similarity.

As shown in Fig. 3.11, Image $I_i$ and Image $I_j$ are a pair of images to be compared. $T_{i1}, T_{i2}, ... T_{im}$ are original user provided tags of image $I_i$, while $T_{j1}, T_{j2}, ... T_{jn}$ are original user provided tags of image $I_j$.

$DI_{ij}$ is the distance of image $I_i$ and image $I_j$.

We define the Semantic Group Similarity as Formula 3.2.13:

$$ DI_{ij} = AVG2 \{ AVG1[SEL(dT_{im\rightarrow jn})], AVG1[SEL(dT_{jn\rightarrow im})] \} $$

(3.2.13)

where

- the internal $SEL$ could be $Maximum – Max$, $Average – Avg$ or $Minimum – Min$;

- the function $d$ is the similarity calculating by any one of the Concept Proximity Measures ($Confidence$ or $NGD$ or $PMI$);

- the external AVG is calculating the Average value.

The reason of choosing these three Concept Proximity Measures ($Confidence$ or $NGD$ or $PMI$) is that they are dynamic changing with the Web and can be applied to any search/retrieval engine. At the same time, they have less limitations (e.g. not limited to a certain ontology).

In order to compare the similarity calculation performance, we would compare different composition and combination in the Group Similarity Measurement.

$$ dT_{im} \rightarrow dT_{jn} = \begin{pmatrix} dT_{i1\rightarrow j1}, & dT_{i1\rightarrow j2}, & dT_{i1\rightarrow j3}, & \ldots & dT_{i1\rightarrow jn} \\ dT_{i2\rightarrow j1}, & dT_{i2\rightarrow j2}, & dT_{i2\rightarrow j3}, & \ldots & dT_{i2\rightarrow jn} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ dT_{in\rightarrow j1}, & dT_{in\rightarrow j2}, & dT_{in\rightarrow j3}, & \ldots & dT_{in\rightarrow jn} \end{pmatrix} $$

(3.2.14)
\[ dT_{im} \to dT_{jn} = \begin{pmatrix} dT_{j1\to i1}, & dT_{j1\to i2}, & dT_{j1\to i3}, & \ldots & dT_{j1\to im} \\ dT_{j2\to i1}, & dT_{j2\to i2}, & dT_{j2\to i3}, & \ldots & dT_{j2\to im} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ dT_{jn\to i1}, & dT_{jn\to i2}, & dT_{jn\to i3}, & \ldots & dT_{jn\to im} \end{pmatrix} \] (3.2.15)

\[ AVG1[SEL(dT_{im\to jn})] = \text{avg}[SEL(dT_{i1\to jn}), SEL(dT_{i2\to jn}), \ldots, SEL(dT_{im\to jn})] \] (3.2.16)

\[ AVG1[SEL(dT_{jn\to im})] = \text{avg}[SEL(dT_{j1\to im}), SEL(dT_{j2\to im}), \ldots, SEL(dT_{jn\to im})] \] (3.2.17)

\[ AVG2 = AVG \cdot AVG1, AVG2 \] (3.2.18)

Fig. 3.12 and Fig. 3.13 gave the example of the semantic similarity calculation of a pair of images with annotations.

We calculate the similarity of each pair of images, while get the users’ votes of how much similar of certain pair of images. Users can score each pair of images with a number from 0 (=very different) to 5 (=very similar), based on personal opinion. Fig. 3.14 gives a set of samples that of users’ scores. The quantitative experimental results which compare the users’ scores and proposed Context-based Group Distance would be shown in the chapter of experiments and evaluations.

The core algorithm of the proposed Context-based Group Distance calculations is shown as follows:

```python
// Group Distance Calculation
    def __confidence_heuristic(self, links, end):
```

55
Figure 3.12: An example to show the Group Similarity Definition

Figure 3.13: An example to show the Group Similarity Calculation
<table>
<thead>
<tr>
<th>No.</th>
<th>Sample Image Pairs</th>
<th>Average Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1.png" alt="Image Pair 1" /> <img src="image2.png" alt="Image Pair 2" /></td>
<td>2.583</td>
</tr>
<tr>
<td>2</td>
<td><img src="image3.png" alt="Image Pair 3" /> <img src="image4.png" alt="Image Pair 4" /></td>
<td>2.333</td>
</tr>
<tr>
<td>3</td>
<td><img src="image5.png" alt="Image Pair 5" /> <img src="image6.png" alt="Image Pair 6" /></td>
<td>1.333</td>
</tr>
<tr>
<td>4</td>
<td><img src="image7.png" alt="Image Pair 7" /> <img src="image8.png" alt="Image Pair 8" /></td>
<td>1.167</td>
</tr>
<tr>
<td>5</td>
<td><img src="image9.png" alt="Image Pair 9" /> <img src="image10.png" alt="Image Pair 10" /></td>
<td>1.440</td>
</tr>
<tr>
<td>6</td>
<td><img src="image11.png" alt="Image Pair 11" /> <img src="image12.png" alt="Image Pair 12" /></td>
<td>0.917</td>
</tr>
</tbody>
</table>

Figure 3.14: Samples Image Pairs with User Voted Score
self.values_array = []

# put joint prob values into the array
for link in links:
    end_page_freq_value = int(self.__frequencyExtract([end]))
    joint_freq_value = int(self.__frequencyExtract([link[0], end]))

    # error check
    if end_page_freq_value > 0:
        candidate_link_prob_value = joint_freq_value / end_page_freq_value
        self.values_array.append(candidate_link_prob_value)
    else:
        self.values_array.append(0)

def __ngd_heuristic(self, links, end):
    self.values_array = []

    # inout vector: normalized google distance (NGD)
    for link in links:

        candidate_link_freq_value =
            int(self.__frequencyExtract([link[0]]))
        joint_freq_value = int(self.__frequencyExtract([link[0], end]))
        provider_num_indexed_pages = self.max_indexed_pages_int

        if joint_freq_value == 0:
ngd = 0  # min value

else:
    print(candidate_link_freq_value,
          self.image2_freq_value, joint_freq_value,
          provider_num_indexed_pages)
    
ngd = (max(math.log(candidate_link_freq_value),
          math.log(self.image2_freq_value)) -
          math.log(joint_freq_value)) /
    (math.log(provider_num_indexed_pages) -
     min(math.log(candidate_link_freq_value),
         math.log(self.image2_freq_value)))
ngd = 1 - ngd

    if self.debug_level > 2:
        print("NGD:",ngd)
    else:
        # if an error on the start function or 0 results, then
        # log value is input as =0
        ngd = (math.log(self.image2_freq_value) -
              math.log(joint_freq_value)) /
            (provider_num_indexed_pages - 0)
ngd = 1 - ngd

    if self.debug_level > 2:
        print("NGD:",ngd)
self.values_array.append(ngd)

def __pmi_heuristic(self, links, end):
    self.values_array = []

    # input vector: pointwise mutual information (PMI)
    for link in links:

        provider_num_indexed_pages = self.max_indexed_pages_int
        # probability
        candidate_link_freq_value =
            int(self.__frequencyExtract([link[0]]))
        joint_freq_value = int(self.__frequencyExtract([link[0], end]))
        image2_freq_value = self.image2_freq_value

        # frequency computation
        candidate_link_prob_value = candidate_link_freq_value / 
                                    provider_num_indexed_pages
        joint_prob_value = joint_freq_value / 
                           provider_num_indexed_pages
        image2_prob_value = image2_freq_value / 
                            provider_num_indexed_pages

        if joint_prob_value == 0:

            pmi = 0  # min value, independent variables
            if self.debug_level > 1:

print("PMI:","pmi, ",",",

link[0], ")")

else:

#error check on candidate link and image2
if (candidate_link_prob_value > 0) and (image2_prob_value > 0):

    pmi = math.log(joint_prob_value /

            (candidate_link_prob_value * image2_prob_value))

    if self.debug_level > 1:
        print("PMI:",pmi)

else:

    #we consider independent variables as we cannot divide
    by 0 or we have an error on the start page
    pmi = 0

    if self.debug_level > 1:
        print("PMI:",pmi)

    self.values_array.append(pmi)

def __calculatedist(self, alabeldist, blabeldist):

    #Internal calculation
    if self.formula[1] == "avg":
        value_alabeldist = sum(alabeldist) / float(len(alabeldist))
value_blabeldist = sum(blabeldist) / float(len(blabeldist))

if self.formula[1] == "max":
    value_alabeldist = max(alabeldist)
    value_blabeldist = max(blabeldist)

if self.formula[1] == "min":
    value_alabeldist = min(alabeldist)
    value_blabeldist = min(blabeldist)

#External calculation
if self.formula[0] == "avg":
    distance = (value_alabeldist + value_blabeldist) / float(2)

if self.formula[0] == "max":
    distance = float(max(value_alabeldist, value_blabeldist))

if self.formula[0] == "min":
    distance = float(min(value_alabeldist, value_blabeldist))
3.3 Summary

In this chapter, we discuss the semantic similarity measurements for image concepts. We also compared six semantic relatedness measure algorithms: Normalized Google Distance, WordNet Distance, Wikipedia Distance, Flickr Distance, PMI Similarity and PMING similarity. The performance of these relatedness algorithms (i.e. WordNet, NGD and PMING) has been evaluated. A new approach of image semantic similarity measurements - Context-based Group Distance has been proposed. Our system is evaluated quantitatively, and experimental results, which will be shown in the chapter of experiments, indicate that this approach is able to deliver highly competent performance.
Chapter 4

Ontological Query Expansion
Based on Similarity

In the prior chapter, we compared several image semantic similarity measurements. In this chapter, we discuss methods that combine the advantages of the image semantic similarity measurements and knowledge based scheme, propose a ontology-based query expansion method for effective image indexing and retrieval.

We study several semantic concept-based query expansion and re-ranking scheme and compare different ontology-based expansion methods in image search and retrieval. To improve the query expansion efficiency and accuracy, we employ the CYC knowledge base to generate the expansion candidate concepts, then filter and rank the expansion results by calculating concept similarities using the Semantic Relatedness Metrics. Using our knowledge-based query expansion in image retrieval, the efficiency and accuracy has been significantly improved.
4.1 Introduction

The relative ineffectiveness of information retrieval systems [25, 195] is largely caused by the inaccuracy with which a query formed by a few keywords models the actual user information need. One well known method to overcome this limitation is automatic query expansion (AQE), whereby the users original query is augmented by new features with a similar meaning.

The typical AQE [97] can be broken down into the four steps: preprocessing of data source, generation and ranking of candidate expansion features, selection of expansion features and query reformulation.

In image retrieval, a typical approach consists of using query examples with visual features such as colors, textures, and shapes, and iteratively refining the visual query through relevance feedback [97].

Most of the current commercial Search Engine are using user query logs analysis and word frequency (Tf-idf) [207] to assist query expansion process and relevant document ranking. Different from the current main techniques using in the commercial search engine, in this work, we are proposing a semantic similarity based query expansion system to achieve higher quality of search results.

The presence of particular objects in an image often implies the presence of other objects [63]. If term \( U \Rightarrow V \), and if only \( U \) is indexed, then searching for \( V \) will not return the image in the result, even though \( V \) is present in the image. The application of such inferences will allow the index elements \( T_i \) of an image to be automatically expanded according to some probability which will be related to the underlying ontology of the application. For example, an image of “wedding” may implies “Bride”, “rings”, “Wedding cake”, “flower”, etc. An example is shown in Fig. 4.1.

Query expansion relates to the expectation that certain semantic objects tend
Figure 4.1: An example of the semantic correlations of image objects
to occur together. The relevant weighting is expressed as a conditional probability given the presence of other objects. Given the presence of a number of objects $O_1, \ldots, O_n$ in an image, there is a probability that a related object $O_j, (O_j \notin \{O_1, \ldots, O_n\})$ has a high probability of also being present in the same image. While in general, one may set a threshold $h$ for such expansion, i.e. $\text{Prob} [O_j | O_1, \ldots, O_n] \leq h$, we shall mainly concentrate on $\text{Prob} [O_j | O_1] \leq h$. In so doing, we are effectively making a Markov assumption

$$\text{Prob}[O_j|O_1] \simeq \text{Prob}[O_j|O_1, \ldots, O_n]$$  (4.1.1)

if the subscript is regarded as time, and when the objects $O_1, \ldots, O_n$ are suitably ordered. Adopting the Markov assumption will considerably simplify the expansion mechanism and drastically reduce the computational overhead. In relation to ontology-based query expansion, we have developed a CYC-based query expansion framework [105] for effective image retrieval, in which we combine CYC with our image retrieval framework - Pixearch to expand the user’s queries and re-rank the searching results. With the CYC-based query expansion semantic image search and re-ranking model, the accuracy of web image searching has seen significant improvements. However, although CYC-based query expansion has been found to improve the performance of image searching, it also exhibits some limitations. Firstly, the relationships there often do not correspond well to those we encounter in common everyday situations. Secondly, apart from missing domain specific and contextual meaning, it also has a limited coverage of proper nouns, which often form the basis of image queries. Thirdly, CYC is static and lacks currency and unable to incorporate new concepts nor form new dynamic connections among concepts. The Web provides a rich storehouse of data and relationships, and it is advantageous to leverage this knowledge by harvesting it for concepts and semantic connections. In so doing, semantic relationships may be established and harnessed to advance semantic
image search to a new level of relevance. From preliminary experimentation using the Wikipedia Link Vector Model (WLVM), we find that such automatic ontology expansion shows great potential for semantic image retrieval. We also see that there are different ways of increasing retrieval effectiveness by exploiting the relatedness obtained from an ontology architecture, such as:

1. replacing the original query search object $O_i$ by a different yet closely related object $O_j$

2. automatically augmenting the query search object $O_i$ with another closely related object $O_j$

The proper expanded queries and rankings could significantly improve the effectiveness of image retrieval, an example is illustrated in Fig. 4.2.
Figure 4.2: An example of query expansion of image searching engine
4.2 Semantic Query Expansion Using Ontology Architectures

4.2.1 SIM-CYC Distance

The Cyc [44] Knowledge Server is a very large, multi-contextual knowledge base and inference engine developed by Cycorp. The Cyc knowledge base (KB) is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The Cyc inference engine performs general logical deduction (including modus ponens, modus tollens, and universal and existential quantification), with AI’s well-known named inference mechanisms (inheritance, automatic classification, etc.) as special cases.

Semantic similarity [166] refers to similarity between two concepts in a taxonomy such as the CYC upper ontology [125]. The semantic similarity between two classes $C$ and $C'$ is not about the classes themselves. When we say “rivers and ditches are similar”, we are not comparing the set of rivers with the set of ditches. Instead, we are comparing a generic river and a generic ditch. Therefore, we define sim($C$, $C'$) to be the similarity between $x$ and $x'$ if all we know about $x$ and $x'$ is that $x \in C$ and $x' \in C'$.

Assuming that the taxonomy is a tree, if $x_1 \in C_1$ and $x_2 \in C_2$, the commonality between $x_1$ and $x_2$ is $x_1 \in C_0 \land x_2 \in C_0$, where $C_0$ is the most specific class that subsumes both $C_1$ and $C_2$. Therefore,

$$sim(x_1, x_2) = \frac{2 \times \log P(C_0)}{\log P(C_1) + \log P(C_2)}$$

(4.2.2)

Wu and Palmer [219] proposed a measure for semantic similarity that could be regarded as a special case of sim($A$, $B$):
Figure 4.3: Knowledge-based Query Expansion System Framework

\[
sim(A, B) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \quad (4.2.3)
\]

Where \( N_1 \) and \( N_2 \) are the number of IS-A links from A and B to their most specific common superclass \( C \); \( N_3 \) is the number of IS-A links from \( C \) to the root of the taxonomy.

4.2.2 Framework of CYC Based Query Expansion Image Search Model

Using CYC, certain objects in an image may be linked to related objects \[219]. Such inferences will entail examination of the conditional probabilities \( P [J_i | J_j] \), where \( J_i, J_j \) are objects and \( J_j \) is given to be present in an image. Common sense association and ontology in CYC are used to construct an inference tree, which allows the index elements \( X_i \) of an image to be automatically expanded according to given probability linked to the underlying ontology of the domain.
As shown in Fig. 4.3, the conceptual design of our CYC-WordNet based query expansion image retrieval and re-ranking model mainly contains three modules as follows:

- **Image Searching Engine**: It contains user interface which use could pass the plain query to the image searching engine, as well as the Query Expansion System which connect to the Query Expansion Candidate Generator and Query Expansion Candidates Ranking System.

- **Query Expansion Candidate Generator**: Based on CYC knowledge base, it generates the expanded candidates for further processing.

- **Query Expansion Candidates Ranking System**: Based on the similarity measurement results, it filters and re-ranks the expanded queries.

### 4.2.3 CYC-WordNet Based Query Expansion System Work

When user input the keyword (Plain Query $PQ_n$) in Image searching Engine, the query $PQ_n$ is automatically passed to Query Expansion Candidate Generator. Query Expansion Candidate Generator, which is based on CYC knowledge base. It check if $PQ_n$ has multiple concepts and relationships with other concepts and passes the generated linked concepts Expansion Candidates Array ($EC_{n1}$, $EC_{n2}$, ..., $EC_{nm}$) to Query Expansion Candidates Ranking System. In Query Expansion Candidates Ranking System, it processes the Expansion Candidates Array ($EC_{n1}$, $EC_{n2}$EC$nm$) by calculating the concept similarity of $PQ_n$ and ($EC_{n1}$, $EC_{n2}$, ..., $EC_{nm}$) based on WordNet Similarity. It filters to keep the top T similar concepts among ($EC_{n1}$, $EC_{n2}$, ..., $EC_{nm}$), give the filtered and re-ranked results Ranked Expansion Concept Array ($RE_{n1}$, $RE_{n2}$, ..., $RE_{nm}$) by sorting the similarity measurement results. Then it passes the Ranked Expansion Concept Array ($RE_{n1}$, $RE_{n2}$, ..., $RE_{nm}$) back to the
Image Searching Engine. The Image Searching Engine gives the expanded queries back to user via the user interface.

To measure the similarity between \( PQ_n \) and each candidate in Expansion Candidates Array \( (EC_{n1}, EC_{n2}, ..., EC_{nm}) \), we propose to use WordNet Similarity Measurements. The similarity is calculated as follows:

\[
d(c_1, c_2) = \frac{2 \times \log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)}
\]  

(4.2.4)

In this calculation, we use \( PQ_n \) as \( c_1 \) and each candidate in Expansion Candidates Array \( (EC_{n1}, EC_{n2}, ..., EC_{nm}) \) as \( c_2 \).

Fig. 4.4 shows an example of searching images with knowledge-based query expansion.

### 4.2.4 User Iterative Selection and Pruning

As shown in the example After the Expanded Queries have been displayed [106], the user is able to select relevant expanded queries, or even combined expanded queries. Actually, the user also may choose the branches/expanded queries just according to the expanded queries. The reason we add example pictures here is in order to make the search page more easily visualized and user-friendly. The user may simply click the relevant EP/EPs and then the corresponding expanded queries will inform Flickr to get more detailed search results of those specified expanded queries. At this stage, the content-based image search schemes may be performed to mine similar images with the selected example image/images as well. Based on the users own participation and pruning [103], the efficiency and effectiveness of image searching will significantly increase, so that the user does not have to waste plenty of time to browse the hundreds of baseline results to find the desired ones. Of course, if the user has already found the desired results in the baseline round, the subsequent query expansion actions need not take place. Fig. 4 shows the entire workflow of all the
Figure 4.4: An Example of Searching Images with Knowledge-based Query Expansion
Figure 4.5: Flowchart of the Query Expansion system with User Interactive Selection steps in the searching and re-ranking process. A detailed example of searching with the keywords orange within our Cyc-based query expansion images search engine is shown in Fig. 4.5.
4.3 Optimization of Ontological Expansion

We shall design a benchmark set of image queries involving different semantic characteristics which are representative and repeatable. Since such queries are critical to our experimentation instead of subjectively and arbitrarily creating ad hoc queries for experimental measurements, we use a systematic procedure to ensure scientific rigor and design a set of objective benchmark queries \((180, 156)\) to gauge performance. The queries will be divided into two sets.

1. \(S\), the static set which consists of relatively permanent and static concepts and relationships

2. \(D\), the dynamic set which consists of mainly current and dynamically changing concepts and relationships

In relation to the static set \(S\), it is expected that the Visual Dictionary \([155]\) will be used, from which a set of benchmark queries are extracted using a reproducible procedure based on random numbers, with relevance judgment based on the sample images provided there. In relation to the set \(D\), this will involve mining, extracting, and ranking of dynamic queries from social networking sites. To provide a good cross-section of the query types, queries can range from simple to complex, involving single and multiple query terms as well as concrete and abstract concepts and proper nouns. These queries will be applied uniformly across the ontology architectures to calibrate and compare their performance. For the purpose of evaluation, the major Web image search engines will be used with the performance averaged across the different search engines. In addition, for the purpose of performance comparison, these will also be applied to CYC and WordNet distance \([21, 149]\) as well. To optimize performance, we shall expand queries with the highest ranking terms among the architectures. Different mechanisms for ranked expansion will be experimented.
The following are the filtering criteria that will be applied, resulting in the formation of three sets of expandable terms:

1. $S_m$: these are the set of terms with the highest computed mean rank across the ontologies

2. $S_\sigma$: these are the set of terms with a standard deviation of rank across the ontologies that fall within a given threshold

3. $S_v$: these are the set of terms which with a given proportion of ranks exceeding a certain cutoff rank; such proportion can follow a voting scheme and is usually taken to be the majority but need not always be the case.

In dealing with high precision discovery of images, the set $S_m \cap S_\sigma \cap S_v$ will be used for query expansion. For high recall image recovery, the set $S_m \cup S_\sigma \cup S_v$ will be used. Different combination of the above sets as well as different combinations of the ontology architectures will also be studied. The effectiveness of each ontology will be individually assessed and ranked. It is aimed to arrive at an optimal expansion set, based on some combination of the sets $S_m$, $S_\sigma$, $S_v$, each of which, in turn, may be formed by focusing on selected ontologies.
4.4 Extended ROC Analysis

We shall undertake a thorough evaluation of retrieval performance using ROC analysis by studying the confusion matrix \([64]\), which provides a more thorough assessment of classification performance than only using recall and precision, while subsuming these among its measures. While conventional ROC classification deals with finite sets, an extension to the basic ROC analysis will be necessary to deal with infinite sets, which is the case for Web image repositories. While hit rate and sensitivity for finite collections are normally considered to be deterministic ratios, in the present context where parameters cannot be known precisely, we shall apply probabilistic analysis to study their performance characteristics. In order to determine the sensitivity of the classification, we need to, for each element of the benchmark query set, determine the set of true positive images in order to determine the recall. We shall use two methods to do this.

**Tagged Image Method**

We shall make use of a set of tagged relevant images with cardinality \(r_{\text{tagged}}\). It can be shown that the probability of a search containing \(m_{\text{tagged}}\) out of \(N\) returned images follows the distribution \([186]\),

\[
\binom{r_{\text{tagged}}}{m_{\text{tagged}}} \left( \frac{|DB| - r_{\text{tagged}}}{N - m_{\text{tagged}}} \right) \div \left( \frac{|DB|}{N} \right)
\]  

(4.4.5)

where \(|DB|\) is the unknown size of the underlying image collection, and \(m_{\text{tagged}}\) and \(N\) are experimentally determined. From these experimental values, \(|DB|\) may be estimated by \(N^*r_{\text{tagged}}/m_{\text{tagged}}\). If it is observed that there is a total of \(r\) relevant images in the set of \(N\) returned images, the total number of relevant images for an image query \(Q\) in the database may be estimated by \(r^*r_{\text{tagged}}/m_{\text{tagged}}\).

**Regression Analysis**

It is observed in \([112]\) that the number of relevant images in a search engine may
be approximated by a linear regression: $Y = \beta_0 + \beta_1 X$, where $Y$ is the number of relevant images in a page, and $X$ is the page number, and as the number of relevant images drops to zero, we may use

$$|DB| = \sum_{k=1}^{\frac{-\beta_0/\beta_1}{Z_k}} Z_k$$

(4.4.6)

where $Z_i$ is the number of relevant images in page $i$, to estimate the number relevant images in the collection.

From the average of these estimates of true positive images, the sensitivity may be determined, and since the other entries can be obtained directly from experimental values, the full confusion matrix may be obtained to evaluate performance.
4.5 Summary

In this chapter, a knowledge-based query expansion for image retrieval has been developed. With the similarity measurement and expanded query re-ranking scheme, the effectiveness and user experience in image exploration have been improved. System performance will be illustrated in the chapter of experiment.
In this chapter, we developed an adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time [102]. In the course of normal usage, the underlying index structure and contents are gradually and dynamically re-organized. This approach realizes a form of machine learning which can be put into relationship with reinforcement learning [93] and genetic algorithms based approaches [69] where the feedback from the environment, here represented by the user community, is used to evolve the system behavior. A major difference from typical supervised learning scheme is the absence of separate training phase. Here the learning phase is a continuous interactive process since the object repository, the community of users, and user judgment of relevance of objects to search terms dynamically evolve over the time. For this reason an adaptive algorithm has been devised to reflect the continuous change in the term/objects index. The adaptive engine exploits some typical evolutionary techniques such as mutation, random tournament and elitism. Systematic simulation experiments on a large number of objects and search queries have been held.
to tune parameters and to evaluate adaptive ability, performance and scalability of the approach. In addition, experiments on real data with actual users have been performed.
5.1 Relevance Feedback

Relevance feedback [102], originally developed for information retrieval is an online learning technique used to improve the effectiveness of the information retrieval systems. Since its introduction into image retrieval around the mid 1990s, it has attracted tremendous attention in the Content-Based Image Retrieval (CBIR) community. A particularly noteworthy work is [173], which develops a relevance feedback based interactive retrieval approach focusing on content-based aspects such as color, texture and shape, and relevance feedback has since been shown to provide dramatic performance improvement [229, 38]. Recent works that apply relevance feedback techniques to image retrieval systems include [7], which makes use of kernel machines and selective sampling that adaptively modify the similarity measures, and [197], which develops kernel convex machines and exploits the idea that negative samples for relevance feedback form several sub-clusters while positive ones group only in one cluster. In [198], an orthogonal complement component analysis method was used that captures the concepts in all positive samples and demonstrates favourable comparison with those of linear and kernel principal component analysis method. Here, we apply the user-feedback media ranking in a different way. Generally, most of the user-feedback media ranking techniques are applied to the CBIR only. CBIR focuses on retrieval based on the visual feature of the image (e.g. color, texture). However, our approach supports the searching of multimedia resources such as images, videos, and audios, and it has the advantage that it is able to focus on arbitrarily higher-level human properties and perceptive details which are not extractable by machines.
5.2 Evolutionary Self-Organizing Search Engine

A search engine can be characterized as a set of structures and algorithms which indexes a database of resource objects with respect to search terms relevance [102]. Whenever a query, consisting of one or more search terms, is issued the engine is expected to return a $k$-bounded list of objects relevant to the search terms. Compared with other approaches, a particular advantage of the present approach is that it is able to incorporate highly subtle nuances, elusive attributes, and deep-level semantics that may be associated with a particular media object, and that search performance may be improved continuously and automatically without additional intervention. To ensure precision of meaning, it is useful to define some concepts accurately. Fig. 5.1 shows a general framework of the adaptive search engine.

**Definition: Indexing.** Our approach supports searching of media resources such as images, videos, and audios. As it is impossible to extract the semantics in the multimedia data automatically with the current technology, an effective indexing and retrieval of multimedia resources are necessary for a successful search system [111]. Since the concept-based (higher-level human perception) indexing of the multimedia resources are more meaningful than its low-level contents (e.g. color, texture, size), our collective indexing approach focuses on the concept-based approach. Through the iterative use of our model, knowledge from users are collected and novel indexes would be built gradually.

**Definition: Static Search Engine.** Given a set of $n$ objects $\Omega = \{o_1, \ldots, o_n\}$, a set of $m$ search terms $T = \{t_1, \ldots, t_m\}$, a query $Q \subseteq T$, a bound $k$ on the answer length, and a relevance indexing function $I : 2^T \times \Omega \rightarrow \mathbb{R}$, where $\mathbb{R} = [0, 1]$, a static search engine $E$ can be defined by a 4-tuple $E \equiv (\Omega, T, I, \alpha)$ where $\alpha$ is an answer function $\alpha : 2^T \times I \rightarrow \Omega^k$ which returns $\alpha(Q, I) = V = [o_1, \ldots, o_k]$, a vector of $k$ objects in $\Omega$ ranked according to relevance $I$, such that $o_i \preceq o_j \Leftrightarrow I(Q, o_i) \preceq I(Q, o_j)$.
Figure 5.1: The system framework of adaptive search engine
Table 5.1: Illustration of the relevance indexing function $I$

<table>
<thead>
<tr>
<th>$\Omega$</th>
<th>$2^T$</th>
<th>$\mathbb{R}$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>${t_1, t_2, t_5}$</td>
<td>0.3</td>
<td>${(t_1, t_2, t_5), o_1} \rightarrow 0.3$</td>
</tr>
<tr>
<td>$o_2$</td>
<td>${t_1, t_2, t_4}$</td>
<td>0.6</td>
<td>${(t_1, t_2, t_4), o_2} \rightarrow 0.6$</td>
</tr>
<tr>
<td>$o_3$</td>
<td>${t_4, t_6, t_7}$</td>
<td>0.2</td>
<td>${(t_4, t_6, t_7), o_2} \rightarrow 0.2$</td>
</tr>
<tr>
<td>$o_3$</td>
<td>${t_7, t_8, t_{10}}$</td>
<td>0.7</td>
<td>${(t_7, t_8, t_{10}), o_3} \rightarrow 0.7$</td>
</tr>
<tr>
<td>$o_4$</td>
<td>${t_4, t_6, t_7}$</td>
<td>0.8</td>
<td>${(t_4, t_6, t_7), o_4} \rightarrow 0.8$</td>
</tr>
<tr>
<td></td>
<td>$\cdot$</td>
<td>$\cdot$</td>
<td>$\cdot$</td>
</tr>
</tbody>
</table>

where $i, j \in [1, k]$. (Note: the notation $2^S$ denotes the power set of $S$, i.e. the set of all subsets of $S$). As an illustration, consider a specific instance of $I$ showing in Table 5.1. Assuming $\{t_4, t_6, t_7\}$ appears only in rows 4 and 6 in Table 5.1, from the query terms $\{t_4, t_6, t_7\}$, we have here $\alpha(\{t_4, t_6, t_7\}, I) = V[o_2, o_4]$. Practically, we can initially assign random values to the relative relevance values for each term of each object. Those random values can be in $[0,1]$ and follow the Gaussian distribution (normal distribution). Since the normal distribution is a well-known model of quantitative phenomena in the natural and behavioral sciences, it is appropriate to use it for the relevance indexing function.

**Definition: User Feedback.** The user feedback $F_{QV}$ in response to an answer vector $V = [o_1, o_2, ..., o_k]$ for a query $Q$ is an integer $F_{QV} \in [0, k]$ which identifies the object that the user decides to click on, where 0 encodes negative feedback, i.e. the fact that no object is clicked after a given amount of time.

As previously pointed out, we assume a single-click feedback model, and the user is expected to click on a single object because of its relevance, or not clicking at all, if the returned answer vector does not contain objects that the user considers to be
relevant.

Issues such as monitoring user multiple queries sessions or explicit scoring of the queries are beyond the scope of this paper. Most search engine interfaces allow one to easily implement the single-click feedback model, while a more complex detection of the user session behavior would require special software (such as plug-ins or client-side scripts) to monitor user activity.

**Definition: Adaptive Search Engine.** An *adaptive search engine* is a search engine which adapts its response to the user feedback, i.e. the answer to a query depends on the engine’s initial state and the query history. It can be seen as a process over the time that, given an initial engine state $E_0 \equiv (\Omega, T, I, k, \alpha)$ and a series of timed triple $(Q, V, F_{QV})$ and object insertion operations, the internal structures of the search engine is continuously being updated.

The adaptive engine can be characterized by describing the *answer function* $\alpha$ and by the methods used to update the relevance index $I$, the set of objects $\Omega$ and the set of terms $T$, which are terms given in a query to specify the properties of the target data objects.

### 5.2.1 Adaptive Search Engine Architecture

The general architecture of the Adaptive Search Engine is shown in Fig. 5.2. A user submits a query $Q$ through the user interface (*step 1*); the $\alpha$ module processes the query by analyzing the relevance index $I$ and the structure $H$ which contains additional historical statistical information on previous queries, a sorted vector $V = [o_1, ..., o_k]$ is then returned to the user (*step 2*).

The user feedback $F_{QV}$ is used by the *update* module in order to update the relevance index $I$ and the statistical data $H$ (*step 3*). The underlying intuition is that objects with a positive feedback increase their relevance, while a negative feedback produces a relevance decrement. If the query $Q$ introduces new terms, or
new objects are added to the systems, the index $I$ and the statistics $H$ are also updated accordingly (step 4).

### 5.2.2 Query Processing

In general, the aim of a typical search engine is to return the best $k$ objects which are most relevant to the query terms [2, 209, 200, 201, 131, 159, 179, 5, 185, 199], according to the current term/objects relevance index. In search engines for text-based documents [52, 57, 75, 74] a preprocessing analysis is crucial for two activities (1) identifying the potential index terms and (2) computing the relevance of objects with respect to those terms. Once the index has been built after the preprocessing phase, it can be used for retrieval [10, 157, 11]. Afterwards the index is typically updated only when either new documents are indexed, or when document relevance
indicators, such as reference links and document citations, have changed (e.g. these changes are usually detected by a spiderbot software agent [53, 192, 4, 115, 99, 31, 94, 134, 100, 15]).

In an *Adaptive Search Engine* the computation of the relevance index relies instead on the interactive process of collecting users feedback, since the purpose is to reflect the relevance rating as evaluated by a community of users at a given period of time. We shall consider a number of strategies. Fig. 5.3 shows the database level of the adaptive search engine.
Naïve Greedy Strategy

A first strategy, which can be considered as candidate for object selection, is the naïve greedy strategy consisting in returning the best $k$ relevant objects which appear in the current index for a query term $t$.

This is the typical strategy used to build hot links, such as the mentioned “top-ten list of most clicked links” which can be found in many portals home page. More precisely, if the probability of a top-ten object $o_j$ being clicked is $p$, then

$$p_i \propto r_i,$$  \hspace{1cm} (5.2.1)

where $r_i$ is the rank of object $o_i$. A question can be posed about “how reliable is a top-ten to really represent the best ten links?“ The problem is that since the “top-ten“ are more likely to be seen, because they are in the home page, then they are also more likely to be clicked. In other words, an initial bias in some of the top-ten links could randomly boost up the rate of not very relevant links.

It is easy to see that this naïve greedy strategy can easily lead to a local maximum problem when it is applied to an Adaptive Search Engine. Let us assume, for instance, that the sorted vector $V = [o_1, ..., o_k]$ contains the current best $k$ objects for query $Q = \{t\}$, i.e. $\forall o \in V, \forall o' \in \{O - \{o_1, ..., o_k\}\}$, $I(Q, o') \leq I(Q, o)$ and let us assume that the objects in $V$ are sufficiently relevant to produce a user click, i.e. a positive feedback, then any next query $Q = \{t\}$ would eventually contain the same objects, although possibly in a different order, thus hiding potentially more relevant objects which have not had the chance to be shown to the user to receive a positive feedback.

More precisely, let the probability of producing a click for object $o_i$ be $p_i$ with $p_1 \geq p_2 \geq ... \geq p_k > p_0$, where, as before, $p_0$ is the probability of producing no click from the list $V$. Consider a particular object $o_j \in V$, then for each appearance of
\( o_j \), the corresponding index relevance will increase by an average amount of

\[
\Delta I = p_j f_{\text{pos}} - p_0 f_{\text{neg}}
\]

where \( f_{\text{pos}} > 0 \) and \( f_{\text{neg}} > 0 \) respectively signify the increase in index relevance due to positive feedback and decrease in index relevance due to negative feedback. For \( f_{\text{pos}} \geq f_{\text{neg}} \), (see more detailed explanation in the next section for its justification), we have

\[
\Delta I = p_j f_{\text{pos}} - p_0 f_{\text{neg}} \geq (p_j - p_0) f_{\text{pos}} > 0.
\]

Hence, over time, what already appears in \( V \) will keep on being shown as the corresponding index relevance tend to increase. On the other hand, for a possibly highly relevant object \( o' \) not belonging to \( V \), it stands no chance of raising the corresponding index relevance. Thus the index relevance of the two groups \( V \) and \( O - V \) will tend to diverge and the objects being shown to the users may not contain the most relevant object.

It is therefore desirable to devise a mechanism where the query answer \( \alpha(Q, I) = [o_1, o_2, ..., o_k] \) reflects the current system relevance order, but at the same time, it also allows underestimated or recently inserted objects to be submitted to the user for evaluation.

**Randomized Strategy**

In order to overcome the problem of local maximum and discover the “hidde” objects in the search domain, a randomized algorithm has been designed which selects the \( k \) objects in the *answer vector* by sequential random extractions from the index. The algorithm gives proportionally higher chances to best rated objects, but it also gives a non-null probability of appearing in the answer vector to objects which have never been submitted to the evaluation of the user community.
The randomized query processing consists of a randomized tournament among the database objects, which is repeated until $k$ distinct objects are selected. The vector of selected objects, ranked by decreasing relevance, is then returned as an answer to query $Q$.

**Definition: Randomized Query Processing.** Let $A(t, o)$ be the number of times the objects has appeared in the answer for a single term $t$, and $C(t, o)$ be the number of clicks which has been received by the queries including the single term $t$. Let $A(Q, o)$, $C(Q, o)$ and $I(Q, o)$, for each object denote respectively the cumulative values for appearances, clicks and relevance over the terms of the query $Q = \{t_1, t_2, ..., t_m\}$ and $A_Q$, $C_Q$ and $I_Q$ the total values over all the objects in the current domain.

\[
A(Q, o) = \sum_{t \in Q} A(t, o) \quad A_Q = \sum_{o \in \Omega} A(Q, o) \tag{5.2.2}
\]
\[
C(Q, o) = \sum_{t \in Q} C(t, o) \quad C_Q = \sum_{o \in \Omega} C(Q, o) \tag{5.2.3}
\]
\[
I(Q, o) = \sum_{t \in Q} I(t, o) \quad I_Q = \sum_{o \in \Omega} I(Q, o) \tag{5.2.4}
\]

then an object $o$ is assigned probability $\Delta_o/\Phi$ to be selected, in the extraction tournament, as an answer for a query $Q$ where

\[
\Delta_o = (c_1 \times I(Q, o)/I_\Omega + c_2 \times C(Q, o)/A(Q, o) + c_3 \times (1/\max\{A(Q, o), \min_A\}))
\]

is a weighted combination of the object relevance and statistics, and $\min_A$ is the minimum non-zero value (but close to zero) for the number of times that the object has appeared in the answer for a query and $\Phi$ is a normalization term $\Phi = \sum_{o \in \Omega} \Delta_o$.

It is worth to point out the roles of different terms in the $\Delta_o$ expression: term $I(Q, o)/I_\Omega$ denotes the relative relevance of object $o$ in the current index; term $C(Q, o)/A(Q, o)$ denotes the success rate of an object, i.e. how many times has been successful clicked with respect to the times it has appeared in an answer vector; term $1/\max\{A(Q, o), \min_A\}$ is reciprocal of the number of answers in which it has appeared where $\max\{A(Q, o), \min_A\}$ is used to avoid a zero divide error when object
o has appeared zero times (i.e., \( A(Q, o) = 0 \)); \( c_1, c_2, c_3 \) and are weights parameters, which are fixed such that an high value for *relative relevance* prevail over the success rate, which prevails over the *inverse appearance* term.

The idea underlying the randomized tournament with weights \( \Delta_o / \Phi \) is that most successful objects are preferred among objects of similar relevance, and, among objects with the same success rate, and objects with the lesser appearance are given a greater chance to be evaluated by the user community.

### Coverage and Mutation

One of the main features of the proposed randomized query processing is that: on a large number of extraction tournaments the objects are, on average, expected to be extracted proportionally to their relative relevance, i.e. to term \( I(Q, o) / I_Q \). At the same time all the objects still have a chance to be extracted and submitted to the evaluation of the user community, which eventually can boost up their relevance thus eliminating possible bias in the current relevance index. This mechanism is similar to the technique of mutations in genetic algorithms [224, 90, 174]; the randomized extraction guarantees a coverage of the whole object domain where the term \( 1 / \max\{A(Q, o), \min_A\} \) in expression \( \Delta_o / \Phi \) is intended to take into account of such domain coverage. Genetic Algorithms (GA) [187] attempts to find a sub-optimal or optimal solution to a problem by genetically breeding the population of individuals, where each individual represents a possible solution to a given problem. A more complex technique of dynamic elitism is also used to determine the size of mutation.

### Dynamic Elitism

The randomized query processing guarantees to avoid *local maxima* and to limit the effect of initial bias in the relevance index. On the other hand it also introduces
some noise which tend to lower the overall system performance. While the set of query answers will tend, in the long run, to contain the best \( k \) objects, each single answer will actually contain some good relevant objects as well as some irrelevant ones, which downgrade the quality of the provided solution.

The risk of having irrelevant objects is intrinsic to the randomize method, but it is repaid back by the advantage of having genomic variety, which allows the discovery of new objects with a good degree of relevance not yet evaluated by the user.

In order to reduce the noise due to the randomized query processing, we introduce elitism, a technique taken from genetic algorithms [163, 62, 16] which consists of transmitting from one generation to another a certain amount \( e \) of best objects, i.e. the elite. The problem of deciding the elitism degree \( e \) is concerned with the two extreme \( e = 0 \), where no elitism guarantees fast coverage of the domain, but maximum noise, and \( e = k \), where there is no noise, but the algorithm can trap into possible suboptimal local maximum, note that elitism \( e = k \) is equivalent to apply the mentioned naïve greedy strategy.

It has been experimentally observed that a low elitism is preferable in the first stages, when the search engine is operating on new terms. Since there is no separate training phase in our adaptive indexing search engine, our model will evolve continuously through the interactive learning phase initially, which our model will only accept a low level of “noise” objects in the query results. Subsequently, the index would become convergent after this phase and the tolerance of the “noise” objects can be accepted. Therefore, the elitism degree will gradually increase with the usage of the model. i.e. when coverage is important and initial biases could mislead the convergence. On the other hand, in an advanced phase, when many queries have been eventually issued, a higher elitism degree would improve the quality of the query answer.

Our solution dynamically increases elitism \( e \) from the initial stage of 0% elitism
toward a more stable situation where elitism is about 80% of $k$, i.e. the minimal elitism is $p_{\text{min}} = 0.2$.

The dynamic progression of elitism values is computed for the $q$-th query by $e_q = \lfloor q \cdot (1 - p_{\text{min}}) \cdot k \rceil / Q_C$ if $q \leq Q_C$, and $e_q = \lfloor (1 - p_{\text{min}}) \cdot k \rceil$ if $q > Q_C$, where $q$ is the number of issued queries, $Q_C$ is the number of queries which are estimated to be necessary for the index to converge, and $p_{\text{min}}$ is the minimal elitism.

The minimal elitism $p_{\text{min}}$, i.e. the minimal percentage of solution which can vary, together with the relative relevance values of objects can be regarded as providing an analogous of probability of mutation [224, 90, 174] with respect to the best $k$ objects.

5.2.3 Feedback Processing

The general idea of a collective search engine is to prize the relevance index of objects which receive a positive feedback from users, while to “punish” negative feedback. The single-click feedback model assumes that a positive feedback can be detected by a click, since it reflects an explicit user choice. The idea of prizing a clicked object is based on the reasonable assumption that the distribution of clicks, when accumulated over the time, will tend to reflect the distribution of user relevance. On the other hand, a negative feedback can be detected only when no object is clicked in an answer list. The underlying hypothesis is that, as observed in real behaviors, users tend not to click if they receive a list of objects of little relevance, i.e. not useful for their purposes.

Note that when an answer vector $V$ receives a click on an object $o$, nothing of negative can be concluded about the rest of the objects which do not receive the click, e.g. they could have a relevance slightly lesser than $o$, but still good. The difference in relevance will eventually emerge in subsequent queries where they will collect, on average, a smaller number of clicks. On the other hand if an answer vector
V receives no click at all it can be reasonably concluded that, on the average, all the objects in the list are not very relevant and the fact can be annotated by “punishing” the associated relevance index. The feedback update mechanism reflects this criteria by appropriate increment/decrement of the index terms.

**Definition: Positive Feedback.** A *positive feedback* is a single-click on an object in the answer list. The index relevance of the clicked object is increased by a quantity $f_{pos}$; appearance and click statistics are also updated:

\[
\text{if } F_{QV} \neq 0 \text{ and } o_{\text{click}} = V[F_{QV}] \text{ then}
\]

\[
\forall t \in Q, \ I(t,o_{\text{click}}) := I(t,o_{\text{click}}) + f_{pos}
\]

\[
\forall t \in Q, \ C(t,o_{\text{click}}) := C(t,o_{\text{click}}) + 1
\]

\[
\forall i \in [1,k], \forall t \in Q, \ A(t,V[i]) := A(t,V[i]) + 1
\]

(5.2.5)

**Definition: Negative Feedback.** A negative feedback takes place when no object is clicked in in the answer list. The index relevance of all the objects in the answer vector is decreased by a quantity $f_{neg}$, appearance statistics are also updated.

\[
\text{if } F_{QV} = 0 \text{ then}
\]

\[
\forall i \in [1,k], \forall t \in Q, \ I(t,V[i]) := I(t,V[i]) - f_{neg}
\]

\[
\forall i \in [1,k], \forall t \in Q, \ A(t,V[i]) := A(t,V[i]) + 1
\]

(5.2.6)

**Increment/Decrement values.** Since the absence of clicking is a random process, i.e. answer vectors which contains relevant objects are not clicked with a low, but not zero, probability, the amount $f_{neg}$ should not “punish” too much the object relevance. It has been experimentally found that a good value for $f_{neg}$ is $f_{neg} = f_{pos}/k$, i.e. to distribute a $-f_{pos}$ decrement over the $k$ objects in the answer. The value of $f_{pos}$ is usually taken equal to 1.
5.2.4 Introducing New Terms and New Objects

The insertion of new terms and/or new objects can be seen as a moving from engine $E \equiv (\Omega, T, I, k, \alpha)$ to engine $E' \equiv (\Omega', T', I', k, \alpha)$. In the following we will characterize the modification introduced in $\Omega$, $T$ and $I$ by insertion operations.

Object Insertion

When a new object $o_{new}$ is inserted in the index, its relevance with respect to all index terms is set to the initial value $I_{init}$ and its statistics are initialized to 0:

$$\Omega' = \Omega \cup o, \quad T' = T \text{ and } I' \text{ such that}$$

$$\forall t \in T, \forall o \in \Omega, \quad I'(t, o) := I(t, o) \quad \text{and} \quad \forall t \in T, \quad I'(t, o_{new}) := I_{init}$$

$$\forall t \in T, \forall o \in \Omega, \quad A'(t, o) := A(t, o) \quad \text{and} \quad \forall t \in T, \quad A'(t, o_{new}) := 0$$

$$\forall t \in T, \forall o \in \Omega, \quad C'(t, o) := A(t, o) \quad \text{and} \quad \forall t \in T, \quad C'(t, o_{new}) := 0 \quad (5.2.7)$$

The inverse appearance component $1/\max\{A(Q, o), \min_{A}\}$ in the randomized tournament increases the chance for newly introduced objects to appear in a query answer. Initially the new object will be randomly returned to the users in relation to different terms, as more queries are issued, either the new object will eventually increase its relevance with respect to some term, or the same inverse appearance component will tend to discard the object, if selected but not clicked.

5.2.5 Term Insertion

The introduction of new terms in the index is typically triggered by a single term query or a multiple terms query which contains some new terms $t_{new1}, ..., t_{new_k}$ not present in the current domain for $T$.

When a new term is inserted into the index, the relevance index for all the objects
with respect to the new terms $t_{new}$ is set to an initial $I_{init}$ relevance value,

$$T' = T \cup \{t_{new,1}, \ldots, t_{new,k}\}, \Omega' = \Omega, \text{ and } I' \text{ such that}$$

\[
\forall o \in \Omega, \forall t \in T, I'(t,o) := I(t,o), \text{ and} \\
\forall o \in \Omega, \forall \{t_{new,1}, \ldots, t_{new,k}\}, I(t_{new,o}) := I_{init}
\] (5.2.8)

It is worth noting that the evaluation of a query $Q = \{t_{new,1}, \ldots, t_{new,k}\}$ completely based on new terms would produce an highly randomized answer, with high chances of not being clicked.

On the other hand the evaluation of a multiple terms query, like

$$Q = \{t_{new}, t_1, t_2, \ldots, t_q\}$$

which mixes one or more new term $t_{new}$ with several old terms, (like $t_1, \ldots, t_q$), would produce a more meaningful answer based on the already evaluated index. In this latter case the new term is indirectly exploiting its relationship with the other terms to increase the index relevance.

In the index growth approach, consider an object $K$ that is indexed with a term $T_1$. $K$ can be searched by a user query which contains $T_1$ and many objects may be returned in the query result since many objects are indexed with $T_1$. Among these returned objects, a user can distinguish objects by adding another index term $T_2$ to $K$. Thus, the user can search the desired object by entering both index terms in the search query. Consider the searching of the picture Starry Night, painted by Vincent Willem van Gogh, and we assume that the image object is indexed with the term Vincent Willem van Gogh initially. Users can search this image by the term Vincent Willem van Gogh, but sometimes, some user query would be more specific, with both search terms Vincent Willem van Gogh and Starry Night used. The same object would be returned in the result when searching by the term Vincent Willem van Gogh, since the term Starry Night is not indexed yet. Eventually, the user would select the object Starry Night and this suggests that a new index term,
Figure 5.4: Example of automatic image index enrichment

Original Annotation: Starry Night
Enriched Index: Vincent Willem van Gogh
De sterrennacht
Saint-Remy
Oil Painting
Nuit étoilée
The Museum of Modern Art New York
Starry Night, may be useful for indexing this data object. Thus, the new index term would be included in the lowest level of the index hierarchy for this object. For every query that specifies both terms, Vincent Willem van Gogh and Starry Night, the user on selecting this image object will cause an increase in the score of the index terms for that object. Thus, the score of the index would be gradually increased and the new index term would be properly installed. Through progressive usage, the indexing of image objects would be enriched, and such attributes as De sterrennacht, Saint-Remy, oil painting, Nuit etoilee or New York, The Museum of Modern Art may be indexed for this image object (Fig. 5.4). The index evolution and growth of the system is very efficient.
5.3 Summary

Our system [109] is able to overcome (i) by having a resilient, community validated structure which allows personal subjectivity to be filtered off through a robust scoring system, and (ii) by exploiting collective assessment and perception of multimedia objects through continuous usage by the community. By capturing, analyzing and interpreting user response and query behaviour, the patterns of searching and finding multimedia objects may be established. Within the present paradigm, the semantic index may be dynamically constructed, validated, and built-up, and the performance of the system will tend to increase as time progresses. Our system also incorporates a high degree of robustness and fault-tolerance whereby inappropriate index terms will be gradually eliminated from the index, while appropriate ones will be reinforced. By incorporating genetic variations into the design, our system will allow multimedia objects which may otherwise be hidden to be discovered.

The key contributions of this approach is the development of an adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time. A particular advantage of the present system is that the underlying index structure and contents are gradually and dynamically re-organized in the course of normal usage without the need to deliberately activate special procedures from time to time [102]. We have presented an Adaptive Indexing Search Engine whereby the indexing of images resources may be done systematically by keeping track of the users querying behavior. By analyzing the search, relevance feedback and results selection patterns of the community of users, our indexing engine allows advanced properties of images, which otherwise are not automatically extractable, to be gradually indexed and discovered. Through this engine, the retrieval and consumption of multimedia objects (besides images, also includes sounds and videos) becomes possible and effective. Given that the auto-
matic capturing of the properties of most media resources, and hence their automatic indexing, is not possible, such an evolutionary approach will allow user intelligence and judgment to be progressively captured and transferred from the community to the index and will bring substantial benefits to the quality of query answers. In particular, this will obviate the need to perform time consuming, intensive, dedicated manual cataloging and indexing which has shown to be costly and, if done by a small unrepresentative group, can also produce a biased and subjective indexing structure not shared by the social community. Although such indexing is not one-off or immediate, we have shown that a competent level of retrieval performance may be achieved over a reasonable time period. Our engine also incorporates genetic algorithms to enable the mining ad discovery of otherwise obscured or hidden media, and is able to respond dynamically to changing usage patterns caused by evolving community interests and social trends.
Chapter 6

Image Retrieval Based on
MPEG-7 Description Structure

To meet the challenge of the exploration of image big data, we present a comprehensive fully automated approach based on the analysis of image metadata in conjunction with image analysis techniques. In this chapter, we propose an automatic image semantic enrichment approach, which could inject deeper semantics to MPEG-7 Description for effective image search. The semantic concepts enriched from the proposed methods in Chapter 3, 4 and 5 can also bee injected ti the MPEG-7 Annotation structure introduced in this chapter.
6.1 Analyze and Match MPEG-7 Descriptors

MPEG-7 Multimedia Description Schemes (DSs) are metadata structures for describing and annotating multimedia content \[43\]. The DSs provide a standardized way of describing in XML the important concepts related to multimedia content description and content management in order to facilitate searching, indexing, filtering, and access. The DSs are defined using the MPEG-7 Description Definition Language (DDL), which is based on the XML Schema Language, and are instantiated as documents or streams. The resulting descriptions can be expressed in a textual form (i.e., human readable XML for editing, searching, filtering) or compressed binary form (i.e., for storage or transmission). In this paper, we provide an overview of the MPEG-7 Multimedia DSs and describe their targeted functionality and use in multimedia applications \[113\].

The goal of the MPEG-7 standard is to allow interoperable searching, indexing, filtering and access of multimedia content by enabling interoperability among devices and applications that deal with multimedia content description. MPEG-7 describes specific features of multimedia content as well as information related to multimedia content management. MPEG-7 descriptions take two possible forms:

1. a textual XML form suitable for editing, searching, and filtering, and
2. a binary form suitable for storage, transmission, and streaming delivery. Overall, the standard specifies four types of normative elements: Descriptors, Description Schemes (DSs), a Description Definition Language (DDL), and coding schemes.

The MPEG-7 Descriptors are designed primarily to describe low-level audio or visual features such as color, texture, motion, audio energy, and so forth, as well as attributes of multimedia content such as location, time, quality, and so forth. It is
expected that most Descriptors for low-level features shall be extracted automatically in applications.

On the other hand, the MPEG-7 DSs are designed primarily to describe higher-level multimedia features such as regions, segments, objects, events; and other immutable metadata related to creation and production, usage, and so forth. The DSs produce more complex descriptions by integrating together multiple Descriptors and DSs, and by declaring relationships among the description components. In MPEG-7, the DSs are categorized as pertaining to the multimedia, audio, or visual domain. Typically, the multimedia DSs describe content consisting of a combination of audio, visual data, and possibly textual data, whereas, the audio or visual DSs refer specifically to features unique to the audio or visual domain, respectively. In some cases, automatic tools can be used for instantiating the DSs, but in many cases instantiating DSs requires human assisted extraction or authoring tools.

The objective of this section is to provide an overview of the MPEG-7 Multimedia Description Schemes (DSs) being developed as part of the MPEG-7 standard, also briefly reviews the organization of the MPEG-7 DSs and highlights the most relevant aspects of the different classes of DSs.

Figure 6.1 provides an overall structure of the organization of MPEG-7 Multimedia DSs into the following areas: Basic Elements, Content Description, Content Management, Content Description, Content Organization, Navigation and Access, and User Interaction.

MPEG-7 gives us the ability to combine and store the semantic information with the visual features extracted from the image in one XML file. Based on the structure of MPEG-7 Descriptors, we are going to enrich the image information via the following aspects:

- Color Descriptors
Figure 6.1: Overall structure of the organization of MPEG-7 Multimedia DSs

- Texture Descriptors
- Shape Descriptors
- Localization
- Metadata
- Semantic Concept
6.2 Exploiting the Relationship between Image Metadata and Semantic Content

In the photographic world, many images may be broken down to several basic scenes and categories\[217, 114\], such as nature, wildlife, portrait, landscape and sports. A landscape scene comprises the visible features of an area of land, including physical elements such as landforms, living elements of flora and fauna, abstract elements such as lighting and weather conditions. Landscape photography is a normal approach to ensure as many objects are in focus as possible, which commonly adopts a small aperture setting. Sports photography corresponds to the genre of photography that covers all types of sports. The equipment used by a professional photographer usually includes a fast telephoto lens and a camera that has an extremely fast exposure time that can rapidly take photos \[113\].

Therefore, there are definite relationships exist between the category of image and image acquisition parameters \[122\]. Fig. 6.2 lists out some typical image categories. The clustering behavior of images with respect to the image acquisition parameters is illustrated in Fig. 6.3.

The primary idea of our ASA approach is to group images based on embedded image-capture metadata and camera acquisition properties. As there are relationships between the type of scenes and image acquisition parameters, from parameters embedded in images, we may extract the intended scenes of images or semantic concepts from images.

We develop our automated annotation system by using an image database which consists of a collection of images obtained from a photograph album over the Web at random. All images in the database are metadata-embedded and stored in JPEG format. Since those images have no tags associated with them, we manually label all images with semantic concept (i.e. the scene of images) before arbitrarily dividing
Figure 6.2: Image Category according to Image Metadata. Each column shows the different categories of Micro, Night Scenes, Indoor Activities, Day Scenes, Portraits and Outdoor Activities.
Figure 6.3: An example of Image distribution in 3D space of some Flickr images
Figure 6.4: Sample rules of Automatic Semantic Annotation

\[
\forall i \in I, ((f_i \leq 5.6) \land (5 < d_i \leq 8)) \land ((t_i \leq 0.00625) \land (L_i \leq 30)) \lor ((30 < L_i \leq 182) \land (ISO_i \leq 250)) \land (L_i > 182) \lor (t_i \leq 0.003125)) \Rightarrow i \in S_{op}
\]

\[
\forall i \in I, (f_i > 5.6) \land (L_i \leq 25) \land (5 < d_i \leq 8) \land (t_i > 0.003125) \Rightarrow i \in S_{opt}
\]

\[
\forall i \in I, (f_i > 5.6) \land (0.003125 < t_i \leq 0.011111) \land (5 < d_i \leq 8) \land (L_i > 25) \Rightarrow i \in S_{tp}
\]

\[
\forall i \in I, (5 < d_i \leq 8) \land ((f_i \leq 5.6) \land ((L_i \leq 30) \land (t_i > 0.00625)) \lor ((ISO_i > 250) \land (30 < L_i \leq 182))) \lor ((b_i = 1) \land (f_i > 5.6) \land (L_i > 25) \land (t_i < 0.011111)) \Rightarrow i \in S_{ce}
\]

\[
\forall i \in I, (d_i > 10) \land (150 < L_i \leq 400) \land (t_i \leq 0.005) \Rightarrow i \in S_{e}
\]

this image database into two groups, a training set and a test set. We use the training set to build our annotation model while the test set to measure its performance in terms of its error rate.

We construct our algorithm first from the training set by using decision tree technique, then measure its performance by the test set. After structured learning procedures with hundreds of testings, the best rules are obtained [217]. Some sample rules are given in Fig. 6.4 and a comprehensive listing of the rules are given in [217].
6.3 Architecture for image Semantic Information Extraction

As in the case of the Segment DS [113], the conceptual aspect of description can be organized in a tree or in a graph, shown in Figure 6.5. The graph structure is defined by a set of nodes, representing semantic notions, and a set of edges specifying the relationship between the nodes. Edges are described by the Semantic Relation DSs.

The work architecture and flow of our system are shown in Figure 6.6.
Figure 6.6: Overall system workflow
6.4 Automatically Structural Casting to MPEG-7 Representation and Index Building

6.4.1 Scheme of Information Extraction and Standardization


Based on our previous approach of MPEG-7 Structured Annotation Description, in the case of images without any captions or tags, some of these fields, particularly the first four, are automatically filled in a meaningful manner. In addition, semantic concepts have been enriched and expanded through semantic manipulation, which works in conjunction with specialization and generalization hierarchies [10]. Thus, the act of searching for an image with a particular object type has been specialized to a narrower type. For example, in searching for people, the portrait category is used, and in searching for animals, the wildlife category will fit the requirement. The more extensive and complete such hierarchies, the greater the scope for rich semantic manipulation. Besides annotating images with predefined semantic concepts in conjunction with methods and techniques of image processing, visual-feature extraction and semantic concept manipulation, with the help of ontology-based query expansion and object association, we could further enrich semantic concept of Web images, so that more fields in MPEG-7 Structured Annotation Description will be filled automatically. So we are going to develop architecture to combine the pre-described ontology knowledge base with MPEG-7 Structured Annotation Description (SAD), to achieve higher precision of image annotation and retrieval.
Due to the extensive model and complexity of MPEG-7 the standard has not been adopted in industry very well [136]. Only a limited number of scientific prototypes and very few commercial applications for creation, visualization, retrieval and interchange of MPEG-7 metadata exist. The availability of the applications also poses a problem: for example, the MPEG-7 experimental model – the standardized reference implementation – is no longer freely available on the web. What we intend to do is developing a user interface to generate MPEG-7 XML files and to search and retrieve them. We are going to use the MPEG-7 descriptors to provide an efficient scheme to support the following information extraction and standardization

- Free text metadata
- Descriptors for subjective quality assessment
- Structured textual metadata
- Creation information
- Media information
- Shapes
- Color
- Location
- Time
- Image semantics

The prototype of our architecture in Fig. 6.7 is as follows:

After extracting the features and concepts from images, our system will automatically annotate the images by updating the MPEG-7 XML table. So that when
Figure 6.7: The prototype of the ontology-based and MPEG-7 SAD image retrieval architecture
users search for the target images, not only the text or title is indexed. The results will be refined by the affection of the new added MPEG-7 descriptors. The image results will be much more accurate.

### 6.4.2 Real user participative interface

We develop an interface to simulate real searching engine and adopt our methods in to the intelligent searching engine, which involves online user’s interaction. The users are able to do the pruning step for ontology-based expansion for the image search. Because the ideas of a concept in an image are quite different from one another, get user themselves involved could further improve the searching efficiency. Our system could also display the searching results in user-defined order according to the different user’s favor about different interesting point in image features reflected by MPEG-7 descriptors.

By assigning different weight to different image Semantic concepts derived from different descriptors of MPEG-7, our system may provide users the ability of defining significance and affection degree to the indexing results. Our system will automatically adjust weight parameters (FTM, QA, STM, CI, MI, SH, CL, L, T, S...) via the learning process and indexing with the dynamic Image Indexing Score (IIS), which is defined by the following formula:

\[
IIS = \alpha \cdot FTM + \beta \cdot QA + \mu \cdot STM + \gamma \cdot CI + \delta \cdot MI + \varepsilon \cdot SH + \xi \cdot CL + \eta \cdot L + \theta \cdot T + \zeta \cdot S = \int_{k=0}^{n} x^n D^n
\]

where \( x \) is the weight of different descriptors while \( D \) stands for the certain descriptors related to \( x \).

Here is an example of how to assign the weights to each descriptor in MPEG-7.

- Free text metadata – FTM (30%)
- Descriptors for subjective quality assessment – QA (10%)

- Structured textual metadata – STM (10%)

- Creation information – CI (15%)

- Media information – MI (5%)

- Shapes – SH (10%)

- Color – CL (10%)

- Location – L (5%)

- Time – T (5%)

- Image semantics – S (10%)

Through thousand times process with user’s interaction, our system will learn and grow to annotate the images precisely by updating the content of XML and the value of each weight of the descriptors.
6.5 Automatic Tags Generation for MPEG-7 Descriptions

In MPEG-7, the Multimedia Description Scheme (MDS), the MPEG specified Descriptors and Descriptor Schemes dealing with generic features and multimedia descriptions are defined. Objects derived from semantic objects and an extendable set of predefined relations between these objects can be used for constructing a semantic graph, describing the multimedia content [137]. MPEG-7 facilitates TextAnnotation Datatypes to support high-level concept enrichment. An MPEG-7 TextAnnotation Datatype supports annotations in different forms and allows multiple annotations. In this paper, we focus on the StructuredAnnotation Datatype [113].

The Structured Annotation Datatype is one that gives a textual description of events, people, animals, objects, places, actions, purposes, times, attributes and behavior. It provides a structured format that is simple yet an expressive and powerful annotation tool. Syntactic relations such as subject, object and verb modifiers between actions and objects can be described.

By using our method, some of these fields, particularly the first four, may be automatically filled in a meaningful manner. In addition, semantic concepts may be enriched and expanded through semantic manipulation, which may work in conjunction with specialization and generalization hierarchies. Thus, searching for an image with a particular object type may be specialized to a narrower type; for example, in searching for people, the portrait category may be used, and in searching for animals, the wildlife category will fit the requirement. The more extensive and complete such hierarchies, the greater the scope for rich semantic manipulation.

Fig. 6.8 gives some examples of automatically generated MPEG-7 StructuredAnnotation descriptions. Consider the third photograph, the application of our rules indicates that this is a portrait image category and scenes of an outdoor event.
Figure 6.8: Automatic Generation of Semantic MPEG-7 Descriptions for Image from Metadata
Figure 6.9: An example of automatic generation of semantic MPEG-7 descriptions for image from metadata and low level features

Thus, “Landscape” and “The Bund” are indicated under WhatObject. As this is an night scene, “Night Scene” is indicated under WhatAction [113]. The location is obtained from the GPS coordinates which map to The Bund, Shanghai in China; thus in the Where field, the “The Bund, Shanghai, China” is indicated which may be used as a search argument, and in traversing the generalization hierarchy, China is also indicated. Timestamp information allows the season “Winter”, time of day “Night”, and year to be variously indicated “2010”. Further descriptions are possible, depending on how far up or deep down the relevant hierarchy is being traversed, all of which may be used as search arguments.

Besides the descriptions generated from image metadata, some of the semantic description fields can be further filled with other image features. Fig. 6.9 gives an example of further enrichment of semantic concepts with face recognition. So in the description field Agent Object, it indicate the labels of people.
6.6 Summary

The model we proposed is able to automatically fill up the Structured Annotation fields in the MPEG-7 Description Standard which previously could only be performed manually. We envisage that such standard semantic fields may be routinely incorporated into image files similar to current metadata fields such as timestamps to take image retrieval to a deeper semantic level. Our system is evaluated quantitatively, which will be shown in the chapter of experiment, and our approach is able to yield highly promising results.
Chapter 7

Experiments and Evaluation

In the previous chapters, the theories of four aspects in the research has been introduced: semantic-based concept similarity of image, ontological query expansion based on similarity, an adaptive image search engine for deep knowledge, and image retrieval based on MPEG-7. In this chapter, the experiment design, results and evaluation will be illustrated.
7.1 Experiments of Semantic-based Image similarity

7.1.1 Semantic Image Similarity Measurements

We compute the relatedness of the searching keywords and their related concept using WordNet Similarity and Wikipedia Similarity [104], and then search images by the plain keywords as well as the related words expanded from the WordNet Similarity computation and Wikipedia Similarity computation. As shown in Fig. 7.1 and 7.2, for example, the word “downtown” can be inferred to “Brooklyn”, “Maryland”, and so on according to WordNet Ontology Similarity; while “downtown” can be inferred to “driving”, “tower”, and so on according to Wikipedia Ontology Similarity. The blue lines (Relatedness line) indicate the performance of relatedness rates between concept “downtown” and other 327 concepts computed by the WordNet distance/Wikipedia distance. The red lines (Precision 1 line) show the image search precision with only the expanded keyword such as search by “Brooklyn”. The green lines (Precision 2 line) represent the image search precision with the expanded keyword combined with the original plain keyword such as search by “downtown + Brooklyn”. We use Flickr as the Web Image Database the test the precision.

The precision is calculated as follows:

\[
Precision = \frac{|\{relevant\_images\} \cap \{retrieved\_images\}|}{|\{retrieved\_images\}|} \quad (7.1.1)
\]

We also do the experiment on other 100 concepts, 10,000+ web images. And the results are showing in table 7.1 and 7.2:

In these two tables, tagged term is the original keyword, such as “downtown”. Word 1 to word 12 are the expended words/concepts base on WD (WordNet Distance) and WLVM (Wikipedia Link Vector Model). Precision 1 column shows the
Figure 7.1: Performance of relatedness rates between concept downtown and other concepts computed by the WordNet distance and image searching precision

![WordNet Graph]

Figure 7.2: Performance of relatedness rates between concept downtown and other concepts computed by the Wikipedia distance and image searching precision

![WLVM Graph]
Table 7.1: Performance of semantic relatedness measures of WD algorithms with their standard deviations.

<table>
<thead>
<tr>
<th>WD</th>
<th>Relateness</th>
<th>Precision 1</th>
<th>Precision 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagged Term</td>
<td>1</td>
<td>0.630</td>
<td>0.630</td>
</tr>
<tr>
<td>word 1</td>
<td>0.687</td>
<td>0.786</td>
<td>0.905</td>
</tr>
<tr>
<td>word 2</td>
<td>0.582</td>
<td>0.730</td>
<td>0.827</td>
</tr>
<tr>
<td>word 3</td>
<td>0.498</td>
<td>0.629</td>
<td>0.752</td>
</tr>
<tr>
<td>word 4</td>
<td>0.467</td>
<td>0.560</td>
<td>0.710</td>
</tr>
<tr>
<td>word 5</td>
<td>0.401</td>
<td>0.541</td>
<td>0.633</td>
</tr>
<tr>
<td>word 6</td>
<td>0.393</td>
<td>0.429</td>
<td>0.575</td>
</tr>
<tr>
<td>word 7</td>
<td>0.352</td>
<td>0.425</td>
<td>0.514</td>
</tr>
<tr>
<td>word 8</td>
<td>0.325</td>
<td>0.325</td>
<td>0.405</td>
</tr>
<tr>
<td>word 9</td>
<td>0.308</td>
<td>0.270</td>
<td>0.388</td>
</tr>
<tr>
<td>word 10</td>
<td>0.301</td>
<td>0.125</td>
<td>0.345</td>
</tr>
<tr>
<td>word 11</td>
<td>0.265</td>
<td>0.124</td>
<td>0.235</td>
</tr>
<tr>
<td>word 12</td>
<td>0.239</td>
<td>0.05</td>
<td>0.210</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>uninterestingness</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 7.2: Performance of semantic relatedness measures of WLVM algorithms with their standard deviations.

<table>
<thead>
<tr>
<th>Tagged Term</th>
<th>WD</th>
<th>Relateness</th>
<th>Precision 1</th>
<th>Precision 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>word 1</td>
<td>0.995</td>
<td>0.790</td>
<td>0.956</td>
<td></td>
</tr>
<tr>
<td>word 2</td>
<td>0.980</td>
<td>0.764</td>
<td>0.920</td>
<td></td>
</tr>
<tr>
<td>word 3</td>
<td>0.932</td>
<td>0.733</td>
<td>0.885</td>
<td></td>
</tr>
<tr>
<td>word 4</td>
<td>0.887</td>
<td>0.634</td>
<td>0.820</td>
<td></td>
</tr>
<tr>
<td>word 5</td>
<td>0.813</td>
<td>0.625</td>
<td>0.752</td>
<td></td>
</tr>
<tr>
<td>word 6</td>
<td>0.789</td>
<td>0.598</td>
<td>0.630</td>
<td></td>
</tr>
<tr>
<td>word 7</td>
<td>0.781</td>
<td>0.521</td>
<td>0.615</td>
<td></td>
</tr>
<tr>
<td>word 8</td>
<td>0.774</td>
<td>0.450</td>
<td>0.568</td>
<td></td>
</tr>
<tr>
<td>word 9</td>
<td>0.770</td>
<td>0.350</td>
<td>0.485</td>
<td></td>
</tr>
<tr>
<td>word 10</td>
<td>0.762</td>
<td>0.315</td>
<td>0.418</td>
<td></td>
</tr>
<tr>
<td>word 11</td>
<td>0.750</td>
<td>0.308</td>
<td>0.366</td>
<td></td>
</tr>
<tr>
<td>word 12</td>
<td>0.736</td>
<td>0.300</td>
<td>0.354</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>uninterestingness</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
image search average precision with only the expanded keywords of all tested tags. Precision 2 column represents the image search precision with the expanded keyword combined with the original plain keywords of all tested tags.

As we can see in Table 7.1 and 7.2, average precision are all satisfactorily higher compared with the only searching by the original terms. The value of precision increases with the relatedness increases. The performance of semantic relatedness of WD and WLVM could affect the image search performance positively. Meanwhile, the image search precision line goes with the WLVM semantic similarity line a little better than WD semantic similarity line. Furthermore, the precision of searching with original concept combined with the expanded concepts are higher than searching with only expanded keywords. These results indicate that significant improvement in performance may be attained from using the keywords expansion approach base on WordNet or Wikipedia ontology.

7.1.2 Comparison of Different Concept Distance Measurements

The quality of the perception model defined by the $\eta$-VSM was finally measured as the rate of couples correctly classified with respect to the ground truth of human perception. The result of this comparison is shown in the diagram of Fig. 7.3 (As published in one of our work work describing a new collaborative proximity measure was presented in 2012, named PMING Distance [66]), where the 100% level corresponds to a perception identical to ground truth. Regarding the PMING Distance, notice that it has in this model the best performance, even better than its components NGD and PMI, with a 69% of correct assignments on the three classes of human perception (high, medium, low) compared to the ground truth of dataset2 published in [218].
7.1.3 Context-based Group Similarity

We designed and implemented a test platform to collect the ground truth (GT) produced by user by getting the users’ votes of how much similar of certain pair of images. The experiment is collaborated with Valentina Franzoni and Alfredo Milani, in the research group at University of Perugia, to get the similarity ranking score from different user group (different country, different culture background). 40 users are involved in this experiment. Users can score each pair of images with a number from 0 (=very different) to 5 (=very similar), based on personal opinion. The user interface of image similarity test experiment is shown in Fig. 7.5.

The system is evaluated quantitatively using more than 500 pairs of Web images on a subset of the standard images dataset Flickr ImageCLEF. For each pair of
Figure 7.4: Concept Distance Comparison

Figure 7.5: The user interface of image similarity test experiment.
images, we compare the users’ scores and the similarity level calculated by the proposed Context-based Group Distance. The tags/annotation which are original ones provides by the authors of a certain images are used. In order to make the experimental results comparable, the User Total Score and Image Group Similarity have been normalized to $[0,1]$.

Two methods are used to evaluate the precision of calculated semantic group similarity. One is to compare the similarity between same image and other images; the other is to compare the similarity ranking of user score and similarity level calculated by the proposed Semantic Group Similarity for the entire target image sets. The sample set of comparison results is shown in the following.

**Comparison 1: Comparing the similarity between same image and other images**

Fig. 7.6 and 7.7 give two randomly chosen sample set of image pairs similarity experiment results - comparing user score and then calculation results by the proposed Group Similarity. In the two Figures, we compare the similarity between one image and another three different images.

In Fig. 7.6, we are using “Bing” as searching engine to get the concept frequency. “Confidence” is used as similarity type and “Average” is used as SEL option in our group similarity edition. The similarity rank results in column “User Voted Average Score” is 1, 2, 3, while rank results in “Similarity Calculated by Group Similarity” is also 1, 2, 3. The same ranking results indicate the image similarity rank results calculated by our proposed methods are exactly the same as the rank of ground truth.

In Fig. 7.7, we are also using “Bing” as searching engine to get the concept frequency. “Normalized Google Distance” is used as similarity type and “Max” is used as SEL option in our group similarity edition. The ranking of “User Voted Average Score” is 2, 1, 3, while the “Similarity Calculated by Group Similarity”
<table>
<thead>
<tr>
<th>Image Pair Number</th>
<th>Sample Image Pairs</th>
<th>User Voted Average Score</th>
<th>Similarity Calculated by Group Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-24</td>
<td><img src="image1.png" alt="Sample Image" /></td>
<td><img src="image2.png" alt="Sample Image" /></td>
<td>0.47826087, 0.398011</td>
</tr>
<tr>
<td>19-20</td>
<td><img src="image1.png" alt="Sample Image" /></td>
<td><img src="image2.png" alt="Sample Image" /></td>
<td>0.434732609, 0.249373</td>
</tr>
<tr>
<td>19-47</td>
<td><img src="image1.png" alt="Sample Image" /></td>
<td><img src="image2.png" alt="Sample Image" /></td>
<td>0.07826087, 0.168459</td>
</tr>
</tbody>
</table>

**Original Tag ID and Tags**

- **19936 beagle**
- **19936 dog**
- **19936 black**
- **19936 white**
- **19936 nose**

- **16054 dog**
- **16054 people**
- **16054 kiss**

- **17405 cat**
- **17405 eye**
- **17405 nose**
- **17405 fur**

- **18239 airplane**
- **18239 aviation**
- **18239 blue**
- **18239 california**
- **18239 commercial**

**Searching Engine:** Bing  
**Similarity Type:** Confidence  
**SEL Option Used:** Average (AVG)

Figure 7.6: Sample set 1 of image pairs similarity experiment results - comparing user score and proposed Group Similarity.
<table>
<thead>
<tr>
<th>Image Pair Number</th>
<th>Sample Image Pairs</th>
<th>User Voted Average Score</th>
<th>Similarity Calculated by Group Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>28-24</td>
<td><img src="image1.jpg" alt="Image Pair" /> <img src="image2.jpg" alt="Image Pair" /></td>
<td>0.32173913</td>
<td>0.837824</td>
</tr>
<tr>
<td></td>
<td>Original Tag ID and Tags</td>
<td>1233 wedding 1233 beach 1233 bw 1233 ocean 1233 romance</td>
<td>16054 dog 16054 people 16054 kiss</td>
</tr>
<tr>
<td>28-29</td>
<td><img src="image1.jpg" alt="Image Pair" /> <img src="image2.jpg" alt="Image Pair" /></td>
<td>0.61739104</td>
<td>0.751992</td>
</tr>
<tr>
<td></td>
<td>Original Tag ID and Tags</td>
<td>1233 wedding 1233 beach 1233 bw 1233 ocean 1233 romance</td>
<td>15436 rings 15436 wedding 15436 bride 15436 groom</td>
</tr>
<tr>
<td>28-46</td>
<td><img src="image1.jpg" alt="Image Pair" /> <img src="image2.jpg" alt="Image Pair" /></td>
<td>0.12173913</td>
<td>0.510378</td>
</tr>
<tr>
<td></td>
<td>Original Tag ID and Tags</td>
<td>1233 wedding 1233 beach 1233 bw 1233 ocean 1233 romance</td>
<td>18168 bell 18168 bicycle 18168 bike 18168 cargobike</td>
</tr>
</tbody>
</table>

Searching Engine: Bing  
Similarity Type: Normalized Google Distance (NGD)  
SEL Option Used: Maximum (Max)

Figure 7.7: Sample set 2 of image pairs similarity experiment results - compares user score and proposed Group Similarity.
is 1, 2, 3. For these three pairs, comparing with the ground truth - “User Voted Average Score”, the similarity ranking of pair 28-24 swaps to the top in proposed group similarity measurement. The user votes image pair 28-29 is the most similar as they are both related to “Wedding and couple” visually. However, our Semantic Group Similarity results treat image pair 28-24 is the most similar as they are both in the context of “kiss and love”. It is interesting to notice that in image number 28, although it does not contain the tag “kiss”, our Semantic Group similarity still can detect it is in the context of “kiss”. That is why it is considered to be very similar to image number 24 as they are both in the same contest.

The results reflect that our Semantic Group Similarity is able to intelligently detect the deep meaning and relationship of images and measure the context similarity in an accurate and effective manner. Comparing with the traditional low level feature extraction techniques, our Group Similarity Measures have much better performance in deep semantic concept similarity measurement and computational efficiency.

Fig.7.8 and 7.9 illustrate the similarity ranking comparison of above two sets of image pairs.
Comparison 2: Comparing the similarity ranking of user score and similarity level calculated by the proposed Semantic Group Similarity for the entire target image sets

For this comparison method, we compare the similarity ranking of an entire image dataset. According to the ranking difference of the ground truth provided
by user (in the column “user total score”) and the similarity level calculated by
our proposed Semantic Group Similarity (in the column “group similarity”), the
Pearson’s r/Spearman’s rho has been calculated to evaluate the similarity ranking
performance of our proposed Semantic Group Similarity.

Two sample sets of experimental results are shown as below.

Using “Bing” as search engine, Table 7.3 shows the ranking results of this situa-
tion: distance type: Confidence, SEL function option: Max. Here, the Semantic
Group Similarity of the images in this dataset are calculated by the following equa-
tion 7.1.2.

\[
D_{ij} = \text{AVG} \{ \text{AVG}[\text{Max}(\text{Confidence}_{(im \rightarrow jn)})], \text{AVG}[\text{Max}(\text{Confidence}_{(jn \rightarrow im)})] \} 
\]

Using “Bing” as search engine, Table 7.4 shows the ranking results of this situa-
tion: distance type: Normalized Google Distance (NGD), SEL function option:
Max. Here, the Semantic Group Similarity of the images in this dataset are calcu-
lated by the following equation 7.1.3.

\[
D_{ij} = \text{AVG} \{ \text{AVG}[\text{Max}(\text{NGD}_{(im \rightarrow jn)})], \text{AVG}[\text{Max}(\text{NGD}_{(jn \rightarrow im)})] \} 
\]

Using “Bing” as search engine, Table 7.5 shows the ranking results of this situa-
tion: distance type: PMI, SEL function option: Max. Here, the Semantic Group
Similarity of the images in this dataset are calculated by the following equation 7.1.4.

\[
D_{ij} = \text{AVG} \{ \text{AVG}[\text{Max}(\text{PMI}_{(im \rightarrow jn)})], \text{AVG}[\text{Max}(\text{PMI}_{(jn \rightarrow im)})] \} 
\]

As shown in the two sample ranking experimental results, the Pearson’s r of
“Confidence-Max”, “NGD-Max” and “PMI-Max” are 0.726427, 0.586892 and 0.446794
Table 7.3: A Subset of Experimental Results *Confidence* – *Max*

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.478261</td>
<td>0.845161</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.756522</td>
<td>0.640428</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.530435</td>
<td>0.612043</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.295652</td>
<td>0.586424</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.373913</td>
<td>0.542236</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.321739</td>
<td>0.540610</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.417391</td>
<td>0.533177</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.434783</td>
<td>0.508839</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.653308</td>
<td>0.498288</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.452174</td>
<td>0.497827</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.313043</td>
<td>0.497988</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.436433</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.278261</td>
<td>0.421583</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.617391</td>
<td>0.392688</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.165217</td>
<td>0.381848</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.400000</td>
<td>0.381199</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.356522</td>
<td>0.370677</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.156522</td>
<td>0.345735</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.345387</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.591304</td>
<td>0.339310</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.486957</td>
<td>0.335856</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.217391</td>
<td>0.303753</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.139130</td>
<td>0.266903</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.191304</td>
<td>0.257082</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.253511</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.139130</td>
<td>0.252647</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.198533</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.194258</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.193243</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.165217</td>
<td>0.191374</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.180035</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.169842</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.278261</td>
<td>0.159529</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.365217</td>
<td>0.155267</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.121739</td>
<td>0.154616</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.148241</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.156522</td>
<td>0.116778</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.121739</td>
<td>0.116349</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.115734</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.115469</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.113890</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.304348</td>
<td>0.111874</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.090853</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>$GD - Confidence$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.067723</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.726427
### Table 7.4: A Subset of Experimental Results $NGD - Max$

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.478261</td>
<td>0.937540</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.452174</td>
<td>0.844856</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.321739</td>
<td>0.837824</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.356522</td>
<td>0.813447</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.417391</td>
<td>0.808851</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.530435</td>
<td>0.808583</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.313043</td>
<td>0.799329</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.165217</td>
<td>0.795358</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.373913</td>
<td>0.792009</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.434783</td>
<td>0.790260</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.156522</td>
<td>0.790180</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.778979</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.756522</td>
<td>0.777028</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.774399</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.754147</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.617391</td>
<td>0.751992</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.278261</td>
<td>0.748208</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.400000</td>
<td>0.745988</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.295652</td>
<td>0.741294</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.734090</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.278261</td>
<td>0.733770</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.591304</td>
<td>0.731008</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.486957</td>
<td>0.729136</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.156522</td>
<td>0.722298</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.191304</td>
<td>0.716306</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.653308</td>
<td>0.716065</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.139130</td>
<td>0.715446</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.699186</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.692404</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.139130</td>
<td>0.690761</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.689678</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.165217</td>
<td>0.682254</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.130435</td>
<td>0.668724</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.304348</td>
<td>0.661953</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.630371</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.078261</td>
<td>0.630352</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.627757</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.624879</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.365217</td>
<td>0.609507</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.217391</td>
<td>0.600549</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.113043</td>
<td>0.588393</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.121739</td>
<td>0.574517</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.121739</td>
<td>0.510378</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>$GD - NGD$</td>
<td>$avg - avg - max$</td>
<td>0.095652</td>
<td>0.486959</td>
</tr>
</tbody>
</table>

Pearson’s $r$: 0.586892
Table 7.5: A Subset of Experimental Results $PMI – Max$

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.356522</td>
<td>7.827740</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.452174</td>
<td>7.429060</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.478261</td>
<td>7.402140</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.756522</td>
<td>7.040440</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.165217</td>
<td>6.755550</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.156522</td>
<td>6.751890</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.313043</td>
<td>6.736570</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.530435</td>
<td>6.703310</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.321739</td>
<td>6.668340</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.095652</td>
<td>6.666440</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.278261</td>
<td>6.650950</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.378913</td>
<td>6.597050</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.417391</td>
<td>6.582890</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.653308</td>
<td>6.299490</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.617391</td>
<td>6.294870</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.191304</td>
<td>6.113070</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.434783</td>
<td>6.084300</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.130435</td>
<td>6.012710</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.304348</td>
<td>6.004770</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.130435</td>
<td>5.930390</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.113043</td>
<td>5.894140</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.078261</td>
<td>5.882570</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.078261</td>
<td>5.857930</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.165217</td>
<td>5.794870</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.139130</td>
<td>5.792870</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.113043</td>
<td>5.725310</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.156522</td>
<td>5.723450</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.095652</td>
<td>5.715680</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.130435</td>
<td>5.646220</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.139130</td>
<td>5.641770</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.400000</td>
<td>5.493680</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.278261</td>
<td>5.487880</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.486957</td>
<td>5.436550</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.078261</td>
<td>5.350320</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.113043</td>
<td>5.349150</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.217391</td>
<td>5.342880</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.591304</td>
<td>5.079930</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.365217</td>
<td>5.005800</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.113043</td>
<td>4.903700</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.113043</td>
<td>4.793480</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.295652</td>
<td>4.671690</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.121739</td>
<td>4.521760</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.095652</td>
<td>3.926160</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>$GD – PMI$</td>
<td>$avg – avg – max$</td>
<td>0.121739</td>
<td>3.891840</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.446794

138
respectively. Summarizing more experimental results, we found that for image similarity measurement, the performance of similarity ranking is \( \text{Confidence} > \text{NGD} > \text{PMI} \). \( \text{Confidence} \) have the best similarity measurement performance is because that the relationship of visual object/semantic concepts and co-occurrence can be better represent by using Confidences. We also found that for SEL function, the performance of similarity measurements is \( \text{Max} > \text{Ave} > \text{Min} \) which means using \( \text{Max} \) can give the best performance of similarity measurement results.

More experimental results are presented in the appendices.

### 7.1.4 Summary

The system is evaluated quantitatively using more than 500 pairs of Web images outside the training database. Experimental results indicate that the Semantic Group Similarity can give more accurate and effective similarity measurements than calculating based on image low level features, in terms of the deep semantic concept and relations similarities, as well as computational cost.
7.2 Experiments of Knowledge-based Query Expansion

7.2.1 Experimental Results

Measures of performance are taken between the unaided approach similar to that in searching engines and the proposed approach for each individual query as well as collectively for their union. A set of representative semantic queries, which usually contain different confusing concepts, is designed for the experiments. The following are used to measure system performance of both approaches on the same image collection:

\[ \text{precision} \]

\[ \text{Precision} = \frac{|\text{RelevantImages} \cap \text{RetrievedImages}|}{|\text{RetrievedImages}|} \quad (7.2.5) \]

A subset of the tested queries (here only shows the plain queries) is contained in Table 7.6. For each query, top five expanded concepts are tested to get the precision \( P_1, P_2, ..., P_5 \). Then we take the mean value of them \((P_1 + P_2 + P_5)/5\) as the precision of proposed approach for each plain query.

\[ \text{average precision} \]

The average precision is the mean value of all individual queries search precisions.

For the unaided approach, we pass the original plain queries the searching engine. As we can see, the column of increased precision obviously indicates the significant advantage of our proposed approach over the unaided approach. As we can see in Fig. 7.10, The average precision of the unaided approach is around 51% while it rises up to approximately 77% for the proposed approach, which is about 27% higher than the unaided approach. These results indicate that significant improvement in performance may be attained from using the proposed approach.
Table 7.6: Sample Plain Queries and Precision Comparison of Image Search with Plain Query and Plain Query + Expanded Query.

<table>
<thead>
<tr>
<th>Plain Query</th>
<th>Unaided Approach</th>
<th>Proposed Approach</th>
<th>Increased</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>47%</td>
<td>62%</td>
<td>15%</td>
</tr>
<tr>
<td>ball</td>
<td>52%</td>
<td>74%</td>
<td>22%</td>
</tr>
<tr>
<td>case</td>
<td>66%</td>
<td>78%</td>
<td>12%</td>
</tr>
<tr>
<td>drive</td>
<td>41%</td>
<td>69%</td>
<td>28%</td>
</tr>
<tr>
<td>ear</td>
<td>73%</td>
<td>91%</td>
<td>18%</td>
</tr>
<tr>
<td>firm</td>
<td>69%</td>
<td>89%</td>
<td>20%</td>
</tr>
<tr>
<td>game</td>
<td>63%</td>
<td>83%</td>
<td>20%</td>
</tr>
<tr>
<td>head</td>
<td>50%</td>
<td>77%</td>
<td>27%</td>
</tr>
<tr>
<td>iron</td>
<td>67%</td>
<td>81%</td>
<td>14%</td>
</tr>
<tr>
<td>jam</td>
<td>42%</td>
<td>90%</td>
<td>48%</td>
</tr>
<tr>
<td>land</td>
<td>38%</td>
<td>57%</td>
<td>19%</td>
</tr>
<tr>
<td>miss</td>
<td>63%</td>
<td>90%</td>
<td>27%</td>
</tr>
<tr>
<td>net</td>
<td>21%</td>
<td>69%</td>
<td>48%</td>
</tr>
<tr>
<td>orange</td>
<td>30%</td>
<td>93%</td>
<td>63%</td>
</tr>
<tr>
<td>park</td>
<td>56%</td>
<td>82%</td>
<td>26%</td>
</tr>
<tr>
<td>right</td>
<td>17%</td>
<td>54%</td>
<td>37%</td>
</tr>
<tr>
<td>space</td>
<td>23%</td>
<td>69%</td>
<td>46%</td>
</tr>
<tr>
<td>table</td>
<td>59%</td>
<td>75%</td>
<td>16%</td>
</tr>
<tr>
<td>wave</td>
<td>69%</td>
<td>85%</td>
<td>16%</td>
</tr>
<tr>
<td>yard</td>
<td>64%</td>
<td>77%</td>
<td>13%</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>AVE Precision</td>
<td>51%</td>
<td>77%</td>
<td>27%</td>
</tr>
</tbody>
</table>
Figure 7.10: Precision Comparison of Image Search with Plain Query and Plain Query + Expanded Query
7.2.2 Summary

We perform several of similar experiments to demonstrate the ability of our ontology based expansion approach. As shown from the results of our experiments, with the proposed approach of the knowledge based query expansion system, the accuracy of web image searching has seen significant improvements. With the Query Expansion Candidate Generator and Query Expansion Candidates Ranking System, the expanded queries has been selected and refined by the combination system of CYC knowledge base and WordNet similarity. The semantic meanings and concepts of web images are significantly enriched. These experimental results clearly indicate the feasibility of the proposed framework.
7.3 Experiments of Evolutionary Self-Organizing Search Engine

One of the main problems in testing an adaptive search engine is to use a methodology which meets the scientific requirements of being systematic, scalable and repeatable. Moreover tests are also needed in order to refine and tune system parameters (such as $c_1, c_2, c_3, c_4$ weights in the randomized query evaluations, $I_{init}$ initial values, minimal elitism $p_{min}$ and other parameters). Last but not least there is the problem of establishing reference points for system performance evaluation, i.e. being able to assess the relevance indexes the system is producing [102].

Although testing the effectiveness of the adaptive search engine with real users on real data is a final and primary objective, some major problems exist for tests with real users: they are not repeatable due to the interactive nature of the tests, they cannot be used for a systematic tuning of the adaptive search engine parameters, moreover they cannot be done on a large scale to show asymptotic behaviors.

We have then decided to design a two-phase set of experiments consisting of: simulated user tests and real user tests. In the first stage, in order to better evaluate the effectiveness of the adaptive engine on a large scale, a simulation approach based on hidden relevance values has been adopted. Simulated user tests have been preliminarily done to validate the theoretical expected behavior of the adaptive engine and to fine tune the engine parameters. In the second phase real user tests have been held to verify the theoretical expectations and the simulated results, on a set of benchmark images from Flickr with a set of predesigned queries covering different semantic situations. The real user tests have been run on a prototype version of the adaptive engine where volunteer student users of Hong Kong Baptist University have been using the system for a total of about 1300 user sessions and 21000 feedback evaluations.
7.3.1 Simulated User Test Model

The main idea behind the simulation approach based on hidden relevance is that each term/object pair is assigned a hidden relevance value, representing an artificially generated user relevance; the “hidden” values are used to generate positive/negative user feedback. The goal of the adaptive search engine in the simulated user tests is then to approach as close as possible these hidden values which are unknown to the adaptive algorithm. This simulated experiment is a collaborative work with Alice Chan, who was in the same research group.

The architecture of such test model is shown in Fig. 7.11. A blackbox user model provides a sequence of random queries to the search engine. The search engine, which has no access to the hidden relevance values, computes the query answers using its internal structures as described in the previous paragraphs. Answers are then sent
to user module which simulates user feedback on the basis of user’s hidden values.

The main advantage of such an approach is that the tests are repeatable and can be held on different problem sizes, i.e. terms and objects, as well as different distributions of hidden relevance. Moreover, objective performance indexes can be easily established, like measures of distance between the system computed relevance and the user hidden relevance.

In the following, we describe the main features of the user model used in the experiments.

Relevance Distribution Models

Let us assume that $U : T \times \Omega \rightarrow [0, 1]$ is a function which assigns a hidden relevance value to every term/object pair, where 0 means not relevant at all, and 1 totally relevant, then a relevance distribution model establishes how the hidden relevance values are distributed among the objects. In our tests we have experimented normal distributions, with different parameters, and $r$-reduced normal distribution. The latter consists in setting to 0 the relevance of $r$ percent objects in $\Omega$ and assigning a normal distribution to the others. The $r$-reduced normal distribution models the more realistic situation in which, each term has a certain percentage of completely uncorrelated objects in the database. The assignment of a distribution is realized by a pseudo random generator.

Feedback Model

The feedback model is the main part of the user model since it simulates a user evaluating the query answer and decides whether to click an object or not click any object in the answer list, and, in the first case, which object to click.

The user feedback model is based on the assumption that evaluation of a user community is a randomized process with a bias toward user hidden relevance values.
Since each user in the community which builds the index has his/her own mental model of relevance between terms and objects, we expect that, on average, the user tends to click more on objects with greater relevance, i.e. greatest hidden values with respect to objects with lower relevance.

We also assume that the attitude of the user to not clicking at all is influenced by the most relevant objects on the answer list, instead of being determined by the global relevance of the answer list. In other words, the attitude to click a list which has very good objects, is much higher than to click in a list with a greater global relevance but not a good maximum. For example, assume list \(a_1\) of 20 objects, which sum up a total relevance of 1.490 where two objects have 0.7 relevance while the other 18 objects have a very low 0.005 relevance; let \(a_2\) be another list of 20 objects having a total relevance of 3.0 distributed uniformly on the 20 object for a relevance of 0.15 each, it is apparent the \(a_2\) would result in being less appealing than \(a_1\).

These criteria have been implemented by a model where the feedback \(F_{QV}\) to the answer vector \(V\) of length \(k\) for a query \(Q\) is computed by making a randomized tournament among \(k+1\) elements, where choosing the \((k+1)th\) element represents the choice of not clicking the answer list at all.

Let us assume that the user model preliminary sorts the vector \(V\) in ascending order with respect to the hidden values, then the weight for not clicking in the random tournament is taken as

\[
w_{\text{noclick}} = (1 - \text{max}_{VQ}) \times c_4, \tag{7.3.6}\]

where \(\text{max}_{VQ}\) is the maximum hidden value and \(c_4\) a tuning parameter, while the weights of the candidate elements to be clicked are computed by:

\[
w_{V[i]} = U(Q, V[i]) \times U(Q, V[i]) \quad i = 1, \ldots, k, \tag{7.3.7}\]

where \(U(Q, V[i])\) is the sum of hidden evaluations over \(t \in Q\) for the \(i\)-th object, and
the square component amplifies the difference among elements. The effect is that
the weight of elements with high relevance is greatly amplified when, for instance
they coexist with many low relevance elements. On the other hand elements which
differ a little still tend to have uniform chances.

The probability values used in the tournament are respectively:

\[ w_{\text{noclick}}/\Phi_{\text{user}}, \text{ probability that the user does not click on any object,} \]
\[ w_{V[i]}/\Phi_{\text{user}}, \text{ probability that the user clicks on object of answers list,} \]

where \( \Phi_{\text{user}} \) is a normalization factor

\[ \Phi_{\text{user}} = w_{\text{noclick}} + \sum_{i \in [1,k]} w_{V[i]} \]. \quad (7.3.8) \]

The modeled behavior is then (i) to have a higher probability of not clicking
when the list does not contain good elements, (ii) to have similar probability of
clicking for elements with similar hidden relevance, and (iii) to amplify the gradient
between good and not good elements.

Let us suppose for instance a vector \( V \) of 20 elements, where \( V[1] \) and \( V[2] \) has
relevance 0.9 and \( V[3]...V[20] \) has relevance 0.1. A random click tournament based
only on total hidden relevance would click 50% of the time on the first two elements
and 50% of the times on the other 18 ones. On the other hand the quadratic
amplification of gradient would give to \( V[1] \) and \( V[2] \) a probability of about 90%
and a probability of about 10% of choosing one the others 18 elements.

**Query Model**

The query model is the component of the simulated user model responsible for
generating a sequence of random queries \( Q_1, ..., Q_{n_q} \) to be submitted to the *adaptive
eengine*. In the case of single term queries the only relevant parameter is the single
term \( t \) to test, and the number of queries \( n_q \).

For multiple term queries the relevant parameters are the terms domain \( T \), and
the bounds \( \min_q \) and \( \max_q \) i.e. the minimum and maximum number of terms respectively allowed in a query, and for each query the following must hold: \( \min_q \leq Q_i \leq \max_q \). Another relevant element is the distribution of the occurrence of the terms in each query when \( \min_q < |T| \), in the experiments they have been tested single term queries, fixed size queries where \( |T| = \min_q = \max_q \), fixed size queries where \( \min_q = \max_q < |T| \), and variable length queries where \( \max_q \) is uniformly ascending/descending over time.

### 7.3.2 Real User Test Design

Real user tests have been designed in order to verify the results obtained from the theoretical tests both in terms of convergency properties and behavior determined by the parameters. The benchmark is represented by a set of 1200 images downloaded from Flickr, a popular image and video storing and sharing platform (www.flickr.com) where the ground truth (GT) is represented by the relevance tags assigned by Flickr which have been successively filtered and checked manually.

Since real user feedbacks are a very rare and precious resource for the experiment, the queries and the experimental settings have been carefully designed in order to point out different aspects of the engine and to asses different properties of domains and system settings.

Five different classes of queries with specific keywords covering different semantic situations have been proposed to the users. We have tested three noun keywords: apple, bicycle, flower, and two adjective/noun keywords: green and orange. The idea is that the visual information connected with a noun describing an object, e.g. a flower, is intrinsically less ambiguous than more smooth concepts which are usually linked to adjective. Let consider, for example, the notion of relevant to green the green keyword, depending on the user intended meaning/context, can either refer to a vegetable or to the dominant color in the picture, or to an environmental friendly
Table 7.7: Real user tests: experimental setting

<table>
<thead>
<tr>
<th>Test</th>
<th>Query Key</th>
<th>GTImages</th>
<th>TotImages</th>
<th>AnswerSize</th>
<th>Elitism</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>apple</td>
<td>80</td>
<td>354</td>
<td>20</td>
<td>dynamic</td>
</tr>
<tr>
<td>2</td>
<td>orange</td>
<td>80</td>
<td>1200</td>
<td>20</td>
<td>25%</td>
</tr>
<tr>
<td>3</td>
<td>bicycle</td>
<td>80</td>
<td>1200</td>
<td>10</td>
<td>40%</td>
</tr>
<tr>
<td>4</td>
<td>green</td>
<td>80</td>
<td>500</td>
<td>20</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>flower</td>
<td>80</td>
<td>354</td>
<td>10</td>
<td>–</td>
</tr>
</tbody>
</table>

object in the picture etc. Moreover visual ambiguity can also be associated to certain objects, for instance we can be quite sure about the presence/absence of a bicycle in a picture, while an apple can be more easily confused with other round and red objects or fruits like a red ball, a strawberry or a peach. The presence of a fuzzy semantic and the visual ambiguity can lead real users to make feedback “errors” with respect to the GT thus introducing a form of noise into the system.

In order to assess the engine with the real user a range of different settings: the maximum answer list size, i.e. the number of images returned by a query, has been fixed either to 20 (for tests 1, 2 and 4) or to 10 (for test 3 and test 5), the number of ground truth relevant images being 80 for each keyword, while a different number of total images ranging from 354 to 1200 have been used. The other parameter varying in the tests was the elitism, where dynamic elitism has been tested in test 1, static elitism in test 2 (20%) and 4 (40%) and no elitism in test 4 and 5.

Table 7.7 shows the detailed settings used in the real user tests.

Volunteer students of Hong Kong Baptist University have used on the system for a total of about 1300 user sessions and 21000 feedback evaluations.

It is interesting to note that in all the experiments, the engine started with a completely empty index, where all the terms have the same initial relevance value.
with the same default score, and the adaptive engine has no clue of image relevance. As the user feedbacks are collected, the relevance indexes are dynamically built by the randomized algorithm in order to converge toward the ground truth.

### 7.3.3 Performance Evaluation

Both in the hidden relevance tests and in the real user tests the following quality measures for the system has been considered:

- **Relative answer relevance**, $R_{\text{tot}}$ the ratio between the total (hidden/ground truth) relevance of a query answer and the best possible answer of length $k$; the aim of this measure is to express how optimal is the current query answer with respect to the object currently in the database; if relevance is interpreted as a binary classification (relevant/not relevant) this metric can be seen as a measure of the *precision* of the query answer with respect to the class of relevant objects;

- **Global answer relevance**, is the sum over time of answer relevance normalized with respect to the maximum possible relevance in a given period of time. It is a measure of the global performance of the system over a fixed period of time, with values normalized for comparison purposes;

- **Relevance coverage**, it measures the proportion of objects which have been assessed as relevant over the total number of relevant objects, it is similar to the concept of *recall* in classification problems.

Another important parameter, *convergence speed*, is easily readable from diagrams which show the time evolution of the quality measures. *Relative answer relevance* is a measure of the single query optimality, and *global relative answer relevance* is a measure of the quality over the time of the system. The *relevance coverage*,
on the other hand, tries to assess if the adaptive system focuses only on the top relevant elements and how much is the randomized algorithm reliable in measuring the relevance of the whole set of objects.

### 7.3.4 Test Results and Discussions

The experiments focus on testing the convergence of the simulation model with respect to the total number of objects in the index, the total number of queries, the answer size (i.e. the number of objects returned by a query) and with respect to different configuration settings of the weight parameters \(c_1, c_2, c_3, c_4\) are respectively weighting accumulated relevance, success rate, inverse appearance, not clicking) and different configuration of the static/dynamic elitism ratio.

#### Parameters Tuning

The first part of the experiments focused on system parameters tuning of weights \((c_1, c_2, c_3, c_4)\). In the first series of runs shown in Fig. 7.12, 1,000 objects in 5,000 queries of answer size 10 were performed with different sets of the weight parameters value \((c_1, c_2, c_3)\). In these figures, the “Number of Queries” refers to the number of queries issued against the collection of data objects. By comparing these runs (Fig. 7.12(f)), the performance of the 3\(^{rd}\) run \((c_1 = 100, c_2 = 0.1, c_3 = 0.01)\) results is the best one and very similar to the 4\(^{th}\) run \((c_1 = 1, c_2 = 0, c_3 = 0)\) while the performance of the 5\(^{th}\) run \((c_1 = 1, c_2 = 10, c_3 = 100)\) is the worst. As already noted in paragraph B of section II, the experiments confirm that the contribution of term \(c_1\), i.e. accumulated relevance, should prevail on \(c_2\) and \(c_3\), i.e. on success rate and inverse appearance.

It is worth noting that the convergence is quite fast with the best configuration settings of Fig. 7.12(c). After 500 queries, we already obtain results with over 90% relative relevance, with an average of 70%; eventually after about 2,500 queries, the
average relative relevance is over 90% and constantly converging to about 98%. The noise which can be noted in the graphs is due to the randomized component in the query answer and it is essential to guarantee the exploration of the available object space. The poor result in Fig. 7.12(e) is due to the fact that giving more weight to inverse appearance the algorithm tend to distribute uniformly the appearance of objects, this tendency is apparent and is present also in Fig. 7.12(a) and Fig. 7.12(b).

In comparing the performance of Fig. 7.12(c) and Fig. 7.12(d), it may be concluded that the performance of the former is slightly superior. Therefore, we investigate these two cases further. We re-run these two cases five times, and in these two sets of runs, we keep all the variables unchanged except the values of the weight parameters $c_1$, $c_2$, and $c_3$. In addition, the runs within the same set are performed with the same variables except the initial random relative relevance values. These initial random relative relevance values follow the same distribution with the same mean and standard deviation. In these runs, we have found that the first set of runs consistently outperforms those in the second set. Although the differences are sometimes marginal, in one case the two exhibits noticeable difference in performance. The results are given in Table 7.8, where the difference can be as much as over 10%.

The tests used to tune the weight parameter $c_4$, i.e. weight for not clicking decrement, are shown in Fig. 7.13. The tests has been held varying $c_4$ in the set 1, 0.1, 10, 0, 0.01 while keeping other weight parameters constant as determined in the best configuration ($c_1 = 100$, $c_2 = 0.1$, $c_3 = 0.01$). In this tests the best performance results when $c_4 = 10$, while it gives the worst performance when $c_4 = 0$. This result can be explained intuitively with the idea that not clicking should not drastically decrease the relative relevance of an object with respect to a term.
Table 7.8: Results Comparison between Two Sets of Runs

<table>
<thead>
<tr>
<th>Queries</th>
<th>( c_1=100, c_2=0.1, c_3=0.01 )</th>
<th>( c_1=1, c_2=0, c_3=0 )</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.694454</td>
<td>0.648661</td>
<td>7.06</td>
</tr>
<tr>
<td>1000</td>
<td>0.819258</td>
<td>0.740686</td>
<td>10.61</td>
</tr>
<tr>
<td>1500</td>
<td>0.864326</td>
<td>0.780576</td>
<td>10.73</td>
</tr>
<tr>
<td>2000</td>
<td>0.890976</td>
<td>0.806454</td>
<td>10.48</td>
</tr>
<tr>
<td>2500</td>
<td>0.904069</td>
<td>0.822060</td>
<td>9.98</td>
</tr>
<tr>
<td>3000</td>
<td>0.916235</td>
<td>0.832256</td>
<td>10.09</td>
</tr>
<tr>
<td>3500</td>
<td>0.922070</td>
<td>0.841829</td>
<td>9.53</td>
</tr>
<tr>
<td>4000</td>
<td>0.927092</td>
<td>0.846728</td>
<td>9.49</td>
</tr>
<tr>
<td>4500</td>
<td>0.932765</td>
<td>0.853442</td>
<td>9.29</td>
</tr>
<tr>
<td>5000</td>
<td>0.937448</td>
<td>0.855796</td>
<td>9.54</td>
</tr>
</tbody>
</table>

Scalability and Queries / Objects ratio

In the series of tests shown in Fig. 7.14, the scalability of the adaptive search engine is evaluated with respect to an increasing number of objects (i.e., number of objects = 1000, 2000, 3000, 4000, 5000) while keeping the number of queries and answer size constant, respectively to 5000 and 10. The results shown in Fig. 7.14(f) are as expected, that while the number of objects increases the general performance decreases. The intuitive reason is that the same amount of queries, and answer size, cannot guarantee an adequate coverage to an increasing amount of objects. Nevertheless after 5000 queries the index converges on 5000 objects up to a 90% of relative relevance with an average of 85% (Fig. 7.14(e) and Fig. 7.14(f) ). A more remarkable result shown in Fig. 7.14f is that the time of index convergece seems to be linearly proportional to the number of queries over number of objects ratio \#queries/\#objects.
Figure 7.12: Parameters Tuning on Weight Parameters $c_1$, $c_2$, and $c_3$. 

(a) 1\textsuperscript{st} run : $c_1 = 1$, $c_2 = 1$, $c_3 = 1$  
(b) 2\textsuperscript{nd} run : $c_1 = 100$, $c_2 = 10$, $c_3 = 1$  
(c) 3\textsuperscript{rd} run : $c_1 = 100$, $c_2 = 0.1$, $c_3 = 0.01$  
(d) 4\textsuperscript{th} run : $c_1 = 1$, $c_2 = 0$, $c_3 = 0$  
(e) 5\textsuperscript{th} run : $c_1 = 1$, $c_2 = 10$, $c_3 = 100$  
(f) Mean Value
Figure 7.13: Parameters Tuning on Weight Parameters $c_4$. 

(a) 1st run: $c_4 = 1$

(b) 2nd run: $c_4 = 0.1$

(c) 3rd run: $c_4 = 10$

(d) 4th run: $c_4 = 0$

(e) 5th run: $c_4 = 0.01$

(f) Mean Value
Figure 7.14: Increasing the Total Number of Objects.
The purpose of this series of tests was to investigate the influence of elitism degree on the convergence of the adaptive engine. The tests has been held with no elitism, static elitism degree, and dynamic elitism, the other parameters settings remains unchanged. Fig. 7.15 shows the results for static elitism degree of 10%(b), 30%(c), 50%(d), 70%(e) and 90%(f). Fig. 7.15(a to f) shows that an increasing elitism degree does not produce any improvement with respect to no elitism. The static 90% elitism improve quickly in the early stage, but afterwards it does not produce any improvement in the long term. It is worth noting that the static 90% elitism corresponds to the greedy strategy of keeping always the best elements, only 10% is the evolution allowed. On the other hand the best approach results dynamic elitism which performs significantly better, also with respect to no elitism. In dynamic elitism, the elitism degree is low in the early stage when adaptation and coverage of objects are important and is gradually increased when relevant objects have been focused on.

A further confirmation of this fact can be found by evaluating the global relative relevance of the experiments, shown in Table 7.9, where the relevance is accumulated and normalized over the time. For each experiment the first number is the global relevance for the all 5000 queries, while the number in bold face is the same measure limited to the last 1000 queries. It can be observed the general increment of the global relative relevance in the second case, which is due to the impact on the global performance of the preliminary convergence phase and, once more, it is worth to point out the results of Dynamic Elitism run (f) which is very sensitive to the performance increment.
Figure 7.15: Comparison of Elitism Strategies.
### Table 7.9: Global Relative Relevance

<table>
<thead>
<tr>
<th>Test</th>
<th>Run</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
</tr>
<tr>
<td>Tuning Weights</td>
<td>0.68</td>
</tr>
<tr>
<td>c₁, c₂, c₃</td>
<td>0.63</td>
</tr>
<tr>
<td>Tuning Weight</td>
<td>0.91</td>
</tr>
<tr>
<td>c₄</td>
<td>0.85</td>
</tr>
<tr>
<td>Increasing Objects</td>
<td>0.92</td>
</tr>
<tr>
<td>Dynamic Elitism</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>0.78</td>
</tr>
</tbody>
</table>

### Real Users Tests: Convergence

The tests with real users has been conducted with the c₁-c₄ best parameters combination determined in the first phase of simulated user tests. The convergence results for the experiments are shown in figures 7.16.

The results with real users tests are quite encouraging and they generally confirms the properties of convergence, scalability of the algorithm as well as the benefits of the dynamic elitism observed in the simulated user tests.

It is also useful to point out that in the experiments with answer size 10, the number of relevance feedbacks was about half of that obtained for answer size 20. The behavior with real users confirms that performance linearly depends on the total number of images and on the answer size, since they constraint the discovery of relevant image and the coverage capability of the algorithm. The test also confirms that static elitism negatively affects the convergence performance despite quickly reaching a relevance level corresponding exactly to the static elitism quota.
Figure 7.16: Relative Relevance for Real User Tests.
(respectively 25% in test 2, and 40% in experiment test 3); the static elitism acts against the explorative capability of the algorithm especially in the early stage of the research, avoiding that other relevant objects can be assessed.

The greater variance which can be observed in the real user tests with respect to the simulated user tests has a twofold explanation: first, the ground truth as provided by Flickr is a crispy boolean value, while the ground truth artificially generated in the simulated tests was a real value distributed in [0,1]. In other words, even when not fully relevant images are extracted in the answer list, they can have intermediate relevance values which can make the simulated test graph smoother; the second reason is that this phenomenon is greatly amplified when there is no dynamic elitism control like in test 4, see fig.7.16-e, since occasionally, despite the average good performance, very bad answers are still possibly generated by a completely free random tournament.

On the other hand, as it has been observed in the first paragraphs, having a certain amount of irrelevant objects is an intrinsic property of the proposed randomized method which guarantee its adaptive behavior, and due to the binary GT of Flickr and the small size of the generated query answer (i.e. a list of 10 or 20 images); variations of few elements can result in great variations of the performance indicators.

**Real Users Tests: Coverage and Noise**

Another important result which is worth to point out is the good *relevance coverage* i.e. the ability of the engine to assess the relevance of the objects in the repository. The *precision* of such relevance evaluation according to the ground truth is also remarkable. Table 7.10 shows that *relevance coverage* and *precisions* are generally quite satisfactory. Another positive result is the ability of the algorithm to be noise tolerant; in table 7.10 we indicated as *WNF* and *WPF*, respectively the *wrong
Table 7.10: Coverage and noise in real user tests

<table>
<thead>
<tr>
<th>Test</th>
<th>Query</th>
<th>Answer Size</th>
<th>Coverage</th>
<th>Precision</th>
<th>WNF</th>
<th>WPF</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>apple</td>
<td>20</td>
<td>1.00</td>
<td>0.96</td>
<td>760</td>
<td>39</td>
<td>0.12</td>
</tr>
<tr>
<td>2</td>
<td>orange</td>
<td>20</td>
<td>0.75</td>
<td>0.76</td>
<td>381</td>
<td>23</td>
<td>0.07</td>
</tr>
<tr>
<td>3</td>
<td>bicycle</td>
<td>10</td>
<td>0.77</td>
<td>0.92</td>
<td>123</td>
<td>2</td>
<td>0.05%</td>
</tr>
<tr>
<td>4</td>
<td>green</td>
<td>20</td>
<td>0.75</td>
<td>0.97</td>
<td>447</td>
<td>96</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>flower</td>
<td>10</td>
<td>0.98</td>
<td>0.97</td>
<td>116</td>
<td>23</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Negative feedbacks and the wrong positive feedbacks, i.e. when the user wrongly gave a feedback rating as non relevant an image whose ground truth defines relevant and vice versa. The wrong user feedback is a form of data noise that can be due to the phenomenon of visual or semantic ambiguity, to an actual user error, to a user disagreeing with the community knowledge, or possibly to a bias in the ground truth. In table 7.10 answer size, WNF and WPF are given as number of units, while coverage, precision and noise are given as percentage values. It must be noted that even a high level of noise (12%) is not affecting the precision of classification, with the exception of orange where the ambiguity can be a suitable explanation. Finally note that, as expected, the level of observed noise for wrong negatives accounts for most of the total noise, from a minimum of 82% for test 4 (green) to 95% for test 1 (apple). In other words, the cases of relevance not recognized by the user are largely greater than noise generated from wrong positive, i.e. images which the users consider relevant while they are not (only two users apparently recognized a bicycle where there was not one!)
7.3.5 Summary

The self-organizing and the exploration capabilities of the algorithm which is able to do the indexing continuously covering most the objects, while maintaining a good performance in terms of total relevance to the query answers, together with the noise tolerance behavior are some of the remarkable benefits which result from the randomized approach. On the other hand strategies based on straightforward 'promotion of the best' focus on some very relevant objects, which prevent them from assessing the others which are basically ignored and have no chance of receiving feedback by the users. Although they can obtain good performance in the short term, this lack of flexibility is a major drawback in domains where the user relevance evaluation dynamically evolve over the time (e.g. social trends) or when new more relevant objects enter the repository.
7.4 Experiments of Image Retrieval Accuracy Based on MPEG-7 Descriptors

7.4.1 Experimental Results

To measure the effectiveness of the approach, controlled experiments are performed. 1,000+ Images are collected using an unbiased randomized mechanism from Flickr.com to form the basis of the experimentation [122]. These represent a cross-section of the different types of Web images, and they consist of three main subsets:

- images where text information is completely absent – subset 1
- images with basic caption – subset 2
- images annotated with keywords and tags – subset 3

Measures of performance are taken between the unaided approach similar to that in search engines and the present approach for each individual subset as well as collectively for their union. A set of representative semantic queries is designed which is used for all the experiments. The following are used to measure system performance of both approaches on the same image collection:

- precision

\[
\text{Precision} = \frac{|\{\text{relevant_images}\} \cap \{\text{retrieved_images}\}|}{|\{\text{retrieved_images}\}|} \quad (7.4.9)
\]

\[
\text{Recall} = \frac{|\{\text{relevant_images}\} \cap \{\text{retrieved_images}\}|}{|\{\text{relevant_images}\}|} \quad (7.4.10)
\]

- average precision
\begin{itemize}
  \item fallout
  \[
  \text{Fallout} = \frac{|\{\text{non-relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{non-relevant images}\}|}
  \tag{7.4.11}
  \]
  \item Fa-score for $\alpha << 1$
  \[
  F_\alpha = \frac{(1 + \alpha^2)(\text{Precision} \times \text{Recall})}{(\alpha^2 \times \text{Precision} + \text{Recall})}
  \tag{7.4.12}
  \]
\end{itemize}

Recall is not included as a direct measure, since for potentially infinite collections, the total number of relevant images cannot be directly determined. However, the Fa-score gives some indication of recall, which may be ascertained for the finite collections in these experiments. Substantially higher weight is assigned to precision ($\alpha=0.01$), which is much more important for Internet search.

Fig. 7.17 shows the accuracy of searching by original human tags and searching by MPEG-7 Descriptors. We found that experimental results of traditional human-tagging are reliable with over 72% accuracy across all scenes [217] and categories. Our model deliver good results and sometimes have better accuracy than human tagging models.

The experimental results are shown in Table 7.11. We see that the precision, F-score and average precision are all satisfactorily higher compared with the unaided approach, while the fallout or false alarm (which is the proportion of non-relevant images retrieved) is kept reasonably low. These results indicate that significant improvement in performance may be attained from using the proposed approach.

\subsection*{7.4.2 Summary}

The MPEG-7 Structured Annotation Datatype is often regarded as a powerful semantic information bearing scheme for images and multimedia objects. Through
Figure 7.17: Image search precision rate comparison between searching by original tags and searching by MPEG-7 Descriptors
Table 7.11: Image search precision rate comparison between unaided approach and proposed approach

<table>
<thead>
<tr>
<th>-</th>
<th>-</th>
<th>Unaided Approach</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subset 1</td>
<td>Precision</td>
<td>-</td>
<td>66.20%</td>
</tr>
<tr>
<td></td>
<td>Fallout</td>
<td>-</td>
<td>35.00%</td>
</tr>
<tr>
<td></td>
<td>$F_a$</td>
<td>-</td>
<td>66.20%</td>
</tr>
<tr>
<td>Subset 2</td>
<td>Precision</td>
<td>57.09%</td>
<td>70.26%</td>
</tr>
<tr>
<td></td>
<td>Fallout</td>
<td>38.60%</td>
<td>21.13%</td>
</tr>
<tr>
<td></td>
<td>$F_a$</td>
<td>58.20%</td>
<td>72.68%</td>
</tr>
<tr>
<td>Subset 1</td>
<td>Precision</td>
<td>82.29%</td>
<td>89.01%</td>
</tr>
<tr>
<td></td>
<td>Fallout</td>
<td>16.74%</td>
<td>12.10%</td>
</tr>
<tr>
<td></td>
<td>$F_a$</td>
<td>79.92%</td>
<td>90.91%</td>
</tr>
<tr>
<td>All Subsets</td>
<td>Average Precision</td>
<td>51.02%</td>
<td>74.76%</td>
</tr>
</tbody>
</table>
the use of the Structured Annotation Datatype, semantic identification and search of images using arguments meaningful to human beings may be used. In the past, Structured Annotation descriptions were generally handcrafted by humans manually. This has become increasingly impractical due to the rapid rate with which images are captured, created, and uploaded.

By the use of image metadata, augmented with image processing and ontological mechanisms, a methodology has been presented which allows the automatic generation of MPEG-7 Structured Annotation descriptions. Admittedly, in its utmost generality, the full Structured Annotation description is comprehensive and challenging even for humans. However, the present automatic approach is able to go a long way towards providing humanly useful and meaningful descriptions by filling out automatically some of the key fields within the Structured Annotation Datatype. In the same way as other forms of metadata such as timestamps are currently already stored together with the images, such automatically generated standard semantic fields may also be routinely attached to images to take their search to a deeper and richer content-oriented semantic level. Additional refinement is no doubt possible and desirable in the future to further fine tune and enrich such descriptions, and the present method is able to provide an important first step towards this.
7.5 Summary

Summing up the experimental results, the semantic-based concept similarity measurement approach provide a new measurements of semantic similarity among images, which outperform than content based image retrieval in terms of deep semantic comparison; ontological query expansion based on similarity promise a better user experience and efficiency in image retrieval; the adaptive image search engine for deep knowledge provides self-organizing and the exploration capabilities of the algorithm which is able to do the indexing continuously covering most the objects, while maintaining a good performance in terms of total relevance to the query answers; the approach of image retrieval based on MPEG-7 enables automatic injection of deeper semantics for effective image search, as well as standardized multimedia management and exploration scheme.
Chapter 8

Summary and Future Research Directions

In the chapter, we conclude the overall results and contributions from the thesis presented in previous chapters. The chapter ends with some pointers to the future work regarding chapters 3 to 7 and an overview of the future of semantic image similarity based on deep knowledge for effective image retrieval.
8.1 Summary and Contributions

Facing the challenges of image big data exploration, Researchers have already devoted a great deal of effort to advancing this technology. In chapter 1, we identify the semantic gap is the main obstacle which limits the effectiveness of image indexing and retrieval. The accuracy of image retrieval depends on meaningful indexing; the key problem of image retrieval is to organize them based on semantics. The word ‘semantic’, which frequently appears in the content of this chapter, is the linguistic interpretation of multimedia objects, such as images and video clips, and is closely associated with the nature and meaning of the underlying objects.

The aim of the literature review, in chapter 2, is to explore the key theoretical and empirical contributions in the current decade related to image indexing and retrieval, image similarity measurements, as well as user involved relevance feedback models. We review a number of models related to content-based image retrieval and concept-based image retrieval for incorporating semantic relationships and the state-of-the-art bridging of the semantic gap.

One of the most important aims of automatic image similarity comparison is to bridge the semantic gap, which is considered as vital problem for image retrieval systems. In chapter 3, we present our approach to measure the context-based group similarity among concepts and images. Comparing with Content Based Image Retrieval (CBIR), which measure the image content similarities by low level features, the proposed Context-based Image Similarity our performed CBIR in effectiveness and accuracy of comparing the deep concept among images.

A knowledge-based query expansion for image retrieval has been developed in chapter 4. With the similarity measurement and expanded query re-ranking scheme, the effectiveness and user experience in image exploration have been improved. We demonstrate the applicability of ontology to the image indexing. The knowledge
based could be fully used to provide query expansion candidates while the concept similarity measurements refine the ranking of expanded queries. The effectiveness and user experience image image explorations have been improved.

An adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time has been proposed in chapter 5. Making full use of high-level human knowledge, perception, incorporating subtle nuances and emotional impression on the multimedia resources, the key contributions of this approach is the development of an adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time. A particular advantage of the present system is that the underlying index structure and contents are gradually and dynamically re-organized in the course of normal usage without the need to deliberately activate special procedures from time to time. Through successive usage of our indexing model, novel image content indexing can be built from deep user knowledge incrementally and collectively by accumulating users judgment and intelligence.

MPEG-7 allows a systematic description of the entities, actions, purposes and times that are represented in audiovisual materials. In chapter 6, the matching between image content and MPEG-7 descriptors have been analysed. An automatic approach which enrich image semantic meanings by automatically filling out some of the key fields within the Structured Annotation Datatype of MPEG-7 standard.

The performances of proposed approaches have been quantitatively evaluated. The experimental results of the proposed approaches have been summarized in chapter 7.
8.2 Future Research Directions

Semantic image similarity measurements for effective image retrieval is an open-ended research topics in need of incremental improvement. Currently commercial multimedia search engines have made significant achievements with very promising results in image searching on text-based search engines.

Research on image classification based on low level feature extraction has become manual while the techniques of deep semantic meaning exploration of multimedia is relatively behindhand. Although Web image retrieval by similar features is developing at a rapid rate, is still not widely deployed for public use due to the computational expense associated with rapid index querying and updating on such an enormous scale. Research on effective, accurate and low computational cost image retrieval from multimedia big data resources still has a long way to go before it can be utilised in our multipurpose image exploration and applications. Obviously, to fill the semantic gap, content-based image retrieval cannot solely be used without assistance from large knowledge base, ontology, dynamic object relationship, and this is one of the directions research is taking in this domain.

According to the statistics data on social media [206], there are more than 3125 photos uploaded on Flickr every 60 seconds while 1 million photos delivered per second on Facebook. As social network users are the center of multimedia big data generator and explorer, the approach for deep concept-based multimedia information retrieval, which make full use of users’ high-level human knowledge, perception, incorporating subtle nuances and emotional impression on the multimedia resources, is in great demand.

Last but not least, application-oriented or domain-specific applications, such as medical image indexing, is also a great trend in image retrieval systems. This kind of systems requires the linkage of professional knowledge. Therefore, image retrieval
is an interdisciplinary research topic.
8.3 Conclusions

In this thesis, we present our approach to measure the context-based group similarity among concepts and images. Comparing with Content Based Image Retrieval (CBIR), which measure the image content similarities by low level features, the proposed Context-based Image Similarity our performed CBIR in effectiveness and accuracy of comparing the deep concept among images. Based one the semantic concept similarity measurements, a knowledge-based query expansion for image retrieval has been developed. With the similarity measurement and expanded query re-ranking scheme, the effectiveness and user experience in image exploration have been improved. We demonstrate the applicability of ontology to the image indexing.

We also develop an adaptive search engine architecture and a robust adaptive index update strategy which enable the system to improve its performance over time. A particular advantage of the present system is that the underlying index structure and contents are gradually and dynamically re-organized in the course of normal usage without the need to deliberately activate special procedures from time to time. Through successive usage of our indexing model, novel image content indexing can be built from deep user knowledge incrementally and collectively by accumulating users judgment and intelligence.

Besides, we propose an automatic approach which enrich image semantic meanings by automatically filling out some of the key fields within the Structured Annotation Datatype of MPEG-7 standard.

The systems are evaluated quantitatively involving the user judgements. Experimental results indicate that our approaches could not only enable advanced degree of semantic similarity measurements and deep meaning enrichment of images, but also deliver highly competent performance, attaining excellent precision and efficiency.
Bibliography


2005. bu entry chapter dan mult -1 olup buraya girildi GG OK 16-01-06 gecen yýl degerlendirildi.


theory of symbolic projections. *Journal of Visual Languages and Computing*,

[31] M. Chau, Z. Huang, and H. Chen. Teaching key topics in computer science and
information systems through a web search engine project. *J. Educ. Resour.

computing approach to creating engineering concept spaces for semantic re-
trieval: The illinois digital library initiative project. *IEEE Transactions on

[33] H.-l. Chen. An analysis of image queries in the field of art history. *Journal of
the American Society for Information Science and Technology*, 52(3):260–273,

[34] J.-L. Chen and G. C. Stockman. Indexing to 3d model aspects using 2d contour

content-based image retrieval. *Pattern Analysis and Machine Intelligence,

[36] Z. Chen and B. Zhu. Some formal analysis of Rocchio’s similarity-based rele-


graphics, virtual Reality, visualisation and interaction in Africa, AFRIGRAPH '03, pages 133–139, New York, NY, USA, 2003. ACM.


[107] C. H. C. Leung and Y. Li. The correlation between semantic visual similarity and ontology-based concept similarity in effective web image search. In


[144] M. Mencacci. Randomized trails in collaborative networks. technical report, Department of Mathematics and Computer Science, University of Perugia, Italy.


Appendices

A.1 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: Conference) - Table 1

A.2 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: NGD) - Table 2

A.3 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{avg} \), Similarity type: PMI) - Table 3

A.4 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: Confidence) - Table 4

A.5 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: NGD) - Table 5

A.6 A Subset of Experimental Results (Formula Combination: \( \text{avg} - \text{avg} - \text{min} \), Similarity type: PMI) - Table 6
<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.478261</td>
<td>0.398011</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.530435</td>
<td>0.337218</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.373913</td>
<td>0.295688</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.417391</td>
<td>0.288467</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.321739</td>
<td>0.277883</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.295652</td>
<td>0.259349</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.434783</td>
<td>0.249373</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.756522</td>
<td>0.237417</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.313043</td>
<td>0.228815</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.226604</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>0.210476</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.452174</td>
<td>0.205114</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>0.202045</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.356522</td>
<td>0.193325</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.653308</td>
<td>0.190817</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>0.187551</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.183659</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.486957</td>
<td>0.178529</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.400000</td>
<td>0.178277</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.591304</td>
<td>0.172530</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.191304</td>
<td>0.131065</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.617391</td>
<td>0.127032</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.124071</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>0.104139</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>0.103194</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.102144</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.097456</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.096355</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.217391</td>
<td>0.094303</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>0.087205</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>0.081220</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>0.073665</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.073544</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.069695</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.365217</td>
<td>0.066863</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.066533</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.064182</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.060757</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>0.054387</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>0.054032</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.304348</td>
<td>0.053899</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.042670</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.041611</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.036204</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.668358
Table 2: A Subset of Experimental Results $NGD – Avg$

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.356522</td>
<td>0.772710</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.478261</td>
<td>0.744305</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>0.734764</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.321739</td>
<td>0.732283</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.530435</td>
<td>0.721759</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.425174</td>
<td>0.711036</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.313043</td>
<td>0.711011</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>0.707290</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.373913</td>
<td>0.705348</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.701538</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.689924</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.687308</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>0.686067</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.417391</td>
<td>0.686995</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.674191</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>0.661906</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.400000</td>
<td>0.659521</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.434783</td>
<td>0.658751</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.650672</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.191304</td>
<td>0.620341</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.486597</td>
<td>0.609717</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.617391</td>
<td>0.602648</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>0.602396</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.591304</td>
<td>0.599022</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.295652</td>
<td>0.595794</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.304348</td>
<td>0.591330</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.590947</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>0.589674</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.653398</td>
<td>0.588247</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.756522</td>
<td>0.576982</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>0.573094</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.568837</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>0.562548</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.547520</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.543670</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>0.539536</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>0.534464</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.365217</td>
<td>0.526088</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.501968</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>0.468874</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>0.426906</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.095652</td>
<td>0.403154</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.217391</td>
<td>0.401079</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>$GD – NGD$</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>0.349354</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.420578
<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.356522</td>
<td>7.431410</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.495652</td>
<td>6.371030</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.452174</td>
<td>6.252340</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>6.245500</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.30435</td>
<td>6.242460</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>6.240880</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>6.232440</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.378913</td>
<td>6.083800</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.313043</td>
<td>6.059640</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.478261</td>
<td>5.997730</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.321739</td>
<td>5.942550</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.417391</td>
<td>5.780230</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>5.544320</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>5.525230</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.304348</td>
<td>5.448700</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.191304</td>
<td>5.411919</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.756522</td>
<td>5.409749</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>5.390660</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.95652</td>
<td>5.371190</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.653308</td>
<td>5.301130</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.156522</td>
<td>5.281900</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.617391</td>
<td>5.201300</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>5.168020</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>5.144510</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.434783</td>
<td>5.131270</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.400000</td>
<td>4.994840</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>4.973130</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.165217</td>
<td>4.922020</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.139130</td>
<td>4.911800</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>4.709960</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.480957</td>
<td>4.698140</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.078261</td>
<td>4.693080</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.278261</td>
<td>4.664850</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.130435</td>
<td>4.650990</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>4.593540</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.591304</td>
<td>4.579720</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.365217</td>
<td>4.364230</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>4.295850</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.295652</td>
<td>4.234110</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.113043</td>
<td>3.870710</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.217391</td>
<td>3.802080</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>3.528360</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.096562</td>
<td>3.317720</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>GD – PMI</td>
<td>avg – avg – avg</td>
<td>0.121739</td>
<td>2.944600</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.330514
Table 4: A Subset of Experimental Results *Confidence* – *Min*

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.530435</td>
<td>0.994696</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.313043</td>
<td>0.980390</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.356522</td>
<td>0.979480</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.478261</td>
<td>0.976879</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.373913</td>
<td>0.975553</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>0.975135</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.321390</td>
<td>0.974743</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.156522</td>
<td>0.974233</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.417391</td>
<td>0.967899</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.591304</td>
<td>0.967892</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>0.954900</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>0.952929</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.165217</td>
<td>0.951760</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.295652</td>
<td>0.945908</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>0.944254</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>0.942764</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.278261</td>
<td>0.941525</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.452174</td>
<td>0.937911</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.486957</td>
<td>0.934741</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.434783</td>
<td>0.932087</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.156522</td>
<td>0.929838</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>0.927634</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.278261</td>
<td>0.927178</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>0.926191</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.400000</td>
<td>0.924704</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.139130</td>
<td>0.924148</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.617391</td>
<td>0.913098</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.191304</td>
<td>0.912618</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>0.911391</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.165217</td>
<td>0.910679</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>0.910548</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.304348</td>
<td>0.910531</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.365217</td>
<td>0.908470</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>0.907128</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>0.905775</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>0.905263</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.653308</td>
<td>0.905261</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>0.905216</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.756522</td>
<td>0.905153</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.121739</td>
<td>0.905095</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>0.904988</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.139130</td>
<td>0.904307</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.217391</td>
<td>0.902014</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>GD – Confidence</td>
<td>avg – avg – min</td>
<td>0.121739</td>
<td>0.901875</td>
</tr>
</tbody>
</table>

Pearson’s r = 0.312896
Table 5: A Subset of Experimental Results \( NGD – Min \)

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.356522</td>
<td>0.724058</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.165217</td>
<td>0.672964</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.995652</td>
<td>0.667212</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.278261</td>
<td>0.646671</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.313043</td>
<td>0.640722</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.130435</td>
<td>0.638605</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.530435</td>
<td>0.627773</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.452174</td>
<td>0.609185</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.156522</td>
<td>0.600171</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.113043</td>
<td>0.596879</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.373913</td>
<td>0.594539</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.156522</td>
<td>0.586523</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.321739</td>
<td>0.580106</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.095652</td>
<td>0.574098</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.478261</td>
<td>0.566926</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.400000</td>
<td>0.555947</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.130435</td>
<td>0.547727</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.191304</td>
<td>0.523077</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.417391</td>
<td>0.518250</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.434783</td>
<td>0.513588</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.113043</td>
<td>0.507636</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.304348</td>
<td>0.500906</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.591304</td>
<td>0.495184</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.139130</td>
<td>0.493884</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.113043</td>
<td>0.488276</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.078261</td>
<td>0.486324</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.295652</td>
<td>0.483010</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.165217</td>
<td>0.478378</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.480957</td>
<td>0.475399</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.617391</td>
<td>0.438434</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.113043</td>
<td>0.432809</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.078261</td>
<td>0.426835</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.653308</td>
<td>0.424209</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.078261</td>
<td>0.422785</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.356217</td>
<td>0.422209</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.278261</td>
<td>0.420194</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.130435</td>
<td>0.419518</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.139130</td>
<td>0.418565</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.756522</td>
<td>0.388041</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.113043</td>
<td>0.303408</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.095652</td>
<td>0.282412</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.121739</td>
<td>0.281134</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.217391</td>
<td>0.243040</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>( GD – NGD )</td>
<td>\text{avg} – \text{avg} – \text{min}</td>
<td>0.121739</td>
<td>0.129265</td>
</tr>
</tbody>
</table>

Pearson’s \( r \) = 0.186751
Table 6: A Subset of Experimental Results $PMI - Min$

<table>
<thead>
<tr>
<th>image 1</th>
<th>image 2</th>
<th>search engine</th>
<th>distance type</th>
<th>formula setting</th>
<th>user total score</th>
<th>group similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.356522</td>
<td>6.960960</td>
</tr>
<tr>
<td>13</td>
<td>40</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>6.059290</td>
</tr>
<tr>
<td>41</td>
<td>48</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.278261</td>
<td>5.889220</td>
</tr>
<tr>
<td>41</td>
<td>51</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.165217</td>
<td>5.715970</td>
</tr>
<tr>
<td>25</td>
<td>41</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.530435</td>
<td>5.569090</td>
</tr>
<tr>
<td>18</td>
<td>25</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.156522</td>
<td>5.540450</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.313043</td>
<td>5.386730</td>
</tr>
<tr>
<td>17</td>
<td>41</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.373913</td>
<td>5.379880</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.452174</td>
<td>5.356730</td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>5.158290</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.321739</td>
<td>5.093330</td>
</tr>
<tr>
<td>41</td>
<td>45</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>5.060530</td>
</tr>
<tr>
<td>36</td>
<td>49</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>4.990280</td>
</tr>
<tr>
<td>35</td>
<td>41</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.417391</td>
<td>4.799150</td>
</tr>
<tr>
<td>36</td>
<td>43</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.156522</td>
<td>4.792620</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>4.739080</td>
</tr>
<tr>
<td>45</td>
<td>48</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.303438</td>
<td>4.695749</td>
</tr>
<tr>
<td>19</td>
<td>24</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.478261</td>
<td>4.669240</td>
</tr>
<tr>
<td>38</td>
<td>48</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>4.613750</td>
</tr>
<tr>
<td>45</td>
<td>51</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.191304</td>
<td>4.587810</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.400000</td>
<td>4.584100</td>
</tr>
<tr>
<td>52</td>
<td>53</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.653308</td>
<td>4.453370</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.434783</td>
<td>4.405380</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>4.350560</td>
</tr>
<tr>
<td>38</td>
<td>41</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.139130</td>
<td>4.217110</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.756522</td>
<td>4.152800</td>
</tr>
<tr>
<td>38</td>
<td>51</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.165217</td>
<td>4.130750</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.591304</td>
<td>4.117950</td>
</tr>
<tr>
<td>27</td>
<td>29</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>4.041230</td>
</tr>
<tr>
<td>47</td>
<td>48</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>4.027200</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.486957</td>
<td>3.994080</td>
</tr>
<tr>
<td>45</td>
<td>47</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.139130</td>
<td>3.933120</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.617391</td>
<td>3.921000</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>3.854560</td>
</tr>
<tr>
<td>38</td>
<td>47</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.130435</td>
<td>3.821140</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.365217</td>
<td>3.767310</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.278261</td>
<td>3.765810</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.295652</td>
<td>3.737140</td>
</tr>
<tr>
<td>47</td>
<td>51</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.078261</td>
<td>3.688460</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.113043</td>
<td>2.800630</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.095652</td>
<td>2.640530</td>
</tr>
<tr>
<td>38</td>
<td>45</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.121739</td>
<td>2.546820</td>
</tr>
<tr>
<td>38</td>
<td>44</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.217391</td>
<td>2.341350</td>
</tr>
<tr>
<td>28</td>
<td>46</td>
<td>bing</td>
<td>$GD - PMI$</td>
<td>avg – avg – min</td>
<td>0.121739</td>
<td>1.610920</td>
</tr>
</tbody>
</table>

Pearson’s $r$ = 0.236505


**Publications**

1. C. H. C. Leung, R.C.F. Wong and Yuanxi Li, Automatic Injection of Deeper Semantics for Effective Image Search, ACM Trans on Intelligent Systems and Technology (Submitted)

2. C. H. C. Leung, Xiaoling Wang and Yuanxi Li, A Regression Model for Predicting Image Search Engines Behaviors, ACM Trans on the Web (Submitted and Revising)

3. Valentina Franzoni, Alfredo Milani, Clement H. C. Leung and Yuanxi Li, Set Similarity Measures for Images Based on Collective Knowledge, IEEE International Conference on Data Mining, 2014 (Accepted)


Curriculum Vitae

Academic qualifications of the thesis author, Miss LI Yuanxi:

- Received the degree of Bachelor of Engineering from Qingdao University, China, July 2009.
- Received the degree of Master of Science from Hong Kong Baptist University, November 2010.

August 2014