SEMI-AUTOMATIC KNOWLEDGE REPRESENTATION FOR SEMANTIC-BASED IMAGE RETRIEVAL EMPLOYING TOPIC MAPS TECHNOLOGY

Siriporn Chongpratheep

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2015

Thesis Title	Semi-Automatic Knowledge Representation for Semantic-Based Image Retrieval Employing Topic Maps Technology
By	Siriporn Chongpratheep
Field of Study	Information Technology
Advisor	Dr. Saprangsit Mruetusatorn

The Graduate School of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirements for the Master's Degree

……………………….………………….. Dean of Graduate School
(Assoc. Prof. Dr. Pichit Sukchareonpong)
Month…..…...… Date…….., Year……..…..

Thesis Committees

……………………….………………….. Chairperson
(Dr. Supakorn Siddhichai)

……………………….………………….. Committee
(Dr. Kanticha Kittipeerachon)

……………………….………………….. Committee
(Dr. Thitiporn Lertrusdachakul)

……………………….………………….. Advisor
(Dr. Saprangsit Mruetusatorn)

SIRIPORN CHONGPRATHEEP : SEMI-AUTOMATIC KNOWLEDGE REPRESENTATION FOR SEMANTIC-BASED IMAGE RETRIEVAL EMPLOYING TOPIC MAPS TECHNOLOGY. ADVISOR : DR. SAPRANGSIT MRUETUSATORN, 108 PP.

Many pieces of research work on image retrieval have been extensively conducted in past decades employing various methods including simple keyword tagging and content-based image retrieval. However, some gaps are still presented as those mentioned methods are missing semantic approach. Basically image search is quite a complex area as image elements require sophisticated algorithm to extract, analyze and find ones with a closest match. The semantic-based Topic Maps are used to identify the image elements to further understand the image content and relationship between each image composition. Thus, this research has been done to prove the concept by implementing Topic Maps related processes on top of a conventional content-based image retrieval engine. Experimental results show that the proposed semantic-based image retrieval system is able to improve the image interpretation. With this way, the proposed image retrieval system is able to process image low-level elements while maintain high-level understanding to produce better result of precision and recall.

Graduate School Student's Signature................................

Field of Study Information Technology Advisor's Signature................................

Academic Year 2015

Acknowledgement

This study has been significantly driven by Dr. Saprangsit Mruetusatorn who reviews on each single part of this study and gives suggestion for improvement.

On the other hand, this research is supported by Takahashi Industrial and Economic Research Foundation (http://takahashi-f.or.jp) regarding monetary fund for research related equipment and activities.

Above all, the author would like to express appreciation to family for all inspiration leading to achievement of this study.

Siriporn Chongpratheep

Table of Contents

Page

Abstract iii

Acknowledgements iv

Table of Contents v

List of Tables viii

List of Figures ix

Chapter

1 Introduction.. 1

1.1 Background..1

1.2 Problem Statement 2

1.3 Scope of Work 2

1.4 Objective...2

1.5 Expected Result 2

1.6 Research Methodology..2

1.7 Chapter Summary 3

2 Literature Review 4

2.1 Related Works and Theories..4

2.1.1 Computer Vision 4

2.1.2 Digital Image Processing and Feature Extraction 5

2.1.3 Distance Metrics.. 7

2.1.4 Content-Based Image Retrieval 8

2.1.5 Semantic Technologies 9

2.1.6 Ontologies..10

2.1.7 Knowledge Representation and Annotation..................11

2.1.8 Topic Maps...11

2.2 Related Programs and Tools...13

2.3 Chapter Summary...13

Table of Contents (Continued)

Chapter Page

3 Research Methodology 14

3.1 System Design..14

3.1.1 Use Case Diagram 14

3.1.2 System Elements 16

3.1.3 System Diagram 17

3.1.4 System Functional Flows 18

3.2 Comparison Approach...20

3.3 Experimental Design 21

3.3.1 Infrastructure Preparation Phase 21

3.3.2 Pre-Processing Phase 22

3.3.3 Training Phase 22

3.3.4 Testing Phase 58

3.3.5 Evaluation Phase 59

3.4 Chapter Summary 60

4 Result, Assessment and Remarkable Features 61

4.1 Experiment Result..61

Test Scenario #1:... 65

Test Scenario #2:... 69

Test Scenario #3:... 73

Test Scenario #4:... 77

4.2 Assessment of Precision and Recall...81

4.3 Remarkable Features of Topic Maps 82

4.3.1 Annotation.. 82

4.3.2 Image Understanding Using TMQL.............................. 90

4.5 Chapter Summary...92

Table of Contents (Continued)

Chapter Page

5 Conclusion 94

5.1 Overall Summary 94

5.2 Additional Analysis 95

5.3 Future Works 95

References 97

Appendix 1 105

Biography 108

List of Tables

Table		Page
2.1	Existing Researches and Related Low-Level Features Used	8
3.1	Use Case Description	15
3.2	Conventional Content-Based Image Retrieval Operations	19
3.3	Topic Maps-Based Image Retrieval Operations	19
3.4	Process Description of CBIR Database Acquisition	24
3.5	Screen Definition for Conventional CBIR Operations	25
3.6	Core LTM Structure	33
3.7	Summary of Experimental Design	59
4.1	Cartoon Characters Used for Testing	61
4.2	Test Scenario #1: Similarity Distance of CBIR for Top 20 Images	66
4.3	Test Scenario #2: Similarity Distance of CBIR for Top 20 Images	70
4.4	Test Scenario #3: Similarity Distance of CBIR for Top 20 Images	74
4.5	Test Scenario #4: Similarity Distance of CBIR for Top 20 Images	78
4.6	Comparison of Precision Rate	82
4.7	Test Scenario #3: Similarity Distance after Conducting Annotation	84
4.8	Test Scenario #4: Similarity Distance after Conducting Annotation	88
4.9	Query for all related instances of a topic	90
4.10	Query for character names wearing necktie	91
4.11	Query for character names wearing necktie and being human	91
4.12	Query for character names wearing red clothes	92

List of Figures

Figure Page

2.1 Example of Image Histogram [10] ... 5
2.2 Example of Color Auto-Correlogram ... 6
2.3 Calculation of Manhattan Distance ... 7
2.4 Composition of Topic Maps TAO ... 12
3.1 Use Case Diagram ... 15
3.2 Overall system elements ... 16
3.3 System Diagram ... 17
3.4 Overall System Process ... 18
3.5 Concept Overview of Comparison Approach ... 20
3.6 Comparison Approach at Specific Point ... 21
3.7 System Infrastructure ... 22
3.8 Conventional CBIR Database Creation Process ... 23
3.9 GUI for CBIR ... 24
3.10 Ontopoly - Main Class for "Character Name" ... 26
3.11 Ontopoly - Main Class for "Image Name" ... 26
3.12 Ontopoly - Instances of "Character Name" ... 27
3.13 Ontopoly - Instances of "Image Name" ... 28
3.14 Ontopoly - Association of "Presents" ... 29
3.15 Ontopoly - Association of "presents" from Character Point of View ... 30
3.16 Ontopoly - Association of "presents" from Image Point of View ... 31
3.17 Topic Maps Network - Semantic-Based Dataset ... 32
4.1 Query Image for Test Scenario #1 ... 65
4.2 Test Scenario #1: Top 20 Images with Closest Distance ... 65
4.3 Test Scenario #1: Topic Maps Network and Related Images ... 67
4.4 Test Scenario #1: Result Images for Topic Maps-Based Method ... 68
4.5 Query Image for Test Scenario #2 ... 69
4.6 Test Scenario #2: Top 20 Images with Closest Distance ... 69
4.7 Test Scenario #2: Topic Maps Network and Related Images ... 71
4.8 Test Scenario #2: Result Images for Topic Maps-Based Method ... 72

List of Figures (Continued)

Figure Page

4.9 Query Image for Test Scenario #3 73
4.10 Test Scenario #3: Top 20 Images with Closest Distance 73
4.11 Test Scenario #3: Topic Maps Network and Related Images 75
4.12 Test Scenario #3: Result Images for Topic Maps-Based Method 76
4.13 Query Image for Test Scenario #4 77
4.14 Test Scenario #4: Top 20 Images with Closest Distance 77
4.15 Test Scenario #4: Topic Maps Network and Related Images 79
4.16 Test Scenario #4: Result Images for Topic Maps-Based Method 80
4.17 Calculation of Precision and Recall 81
4.18 Test Scenario #3: Image Annotation in Ontopoly 83
4.19 Test Scenario #3: New Results after Conducting Annotation 84
4.20 Jerry: Topic Maps Network and Related Images 85
4.21 Jerry Related Images 86
4.22 Test Scenario #4: Image Annotation in Ontopoly 87
4.23 Test Scenario #4: New Results after Conducting Annotation 87
4.24 Sylvester: Topic Maps Network and Related Images 89
4.25 Sylvester Related Images 89

Chapter 1
Introduction

1.1 Background

Digital multimedia is being expanded throughout global storage. People feel that it is not easy to find songs from their favorite movies, some images for their business presentation or even some videos for dinner recipe. This is because the traditional keyword search which is can be used well at the beginning of the World Wide Web becomes lack of effectiveness due to the growing volume of data, which means the keywords as well. Besides, some indexing terms is so informal that they cannot be used for machine computation. These factors obstruct the search engine to perform information analysis and retrieval. Many researches proposed different solutions to fix these issues. The Semantic Technology is also another piece of the cake. With the semantic structure, the computer will be able to comprehend input data and process it in the optimized approach while human-like natural language and expression are still remained. Mostly proposed semantic structure is integrated ontology and information annotation. Moreover, some researchers have implemented the ability of reasoning and inferring to add more robustness to the system.

Cartoon character is another domain facing with a bulk of database resulting in low precision retrieval and non-related excessive recalls. According to [1], cartoon character images are important for cartoon animators, designers and enthusiasts to effectively create new animations or characters by reusing and synthesizing the old resources. Besides, creating new cartoon components including key-frames drawing, painting and in-betweening require a lot of human labor and time which can be reduced if cartoonists do not need to start from scratch [2]. The mentioned issues lead to several solutions in cartoon character retrieval. Regarding this matter, this research then proposes the cartoon character image retrieval system driven by content-based image retrieval in associated with the integrated Topic Maps-based semantic module. The objective is to enhance performance of image retrieval by employing integrated semantic module allowing computation of low-level visual feature similarity matching and human high-level interpretation.

This chapter includes some background of this research and the reason why it should be further conducted to find more solutions which is going to be determined in problem statement. The scope is clarified in order to narrow the experiment. The brief research methodology is presented at the end of this chapter.

1.2 Problem Statement

The continually increased volume of digital image on the global storage requires a unique approach for analyzing, indexing, annotating, visualizing and retrieving to bridge the semantic gap between image low-level features and human high-level interpretation.

1.3 Scope of Work

(1) Images used in the experiment are cartoon characters from fifty cartoon stories based on the Top 50 Cartoon Characters published on the website http://animatedtv.about.com/od/showsaz/tp/top50chrctrs.04.htm [3].

(2) Image used in the experiment is based on traditional 2D image type.

1.4 Objective

To develop image retrieval system which is able to optimize the retrieved results by using Topic Maps engine to increase precision rate.

1.5 Expected Result

The expectation of this research associates with the development of image retrieval system that can produce better results under integrated semantic environment.

1.6 Research Methodology

This study is conducted along with experiment which is divided into five phases; infrastructure preparation, pre-processing, training, testing and evaluation phase.

1.7 Chapter Summary

The objective of this study is to develop semantic-based image retrieval system which can enhance accuracy of retrieved results acquired by the conventional CBIR method. Regarding this purpose, some experiments are required based on five different phases starting from the development of infrastructure element to the assessment of results.

Chapter 2
Literature Review

This chapter is divided into two sections; related works and tools. The first part describes several theories and related works associated with this research including computer vision, digital image processing, distance metrics, content-based image retrieval, semantic technologies, ontologies, knowledge representation and annotation and the Topic Maps. The other part introduces some software and tools used in this study.

2.1 Related Works and Theories

This research is relevant to several principles which are very crucial to create Topic Maps-based image retrieval system. The section covers the conventional Content-based Image Retrieval (CBIR), Semantic Technologies and related concepts which are ontologies, knowledge representation and annotation. The introduction of Topic Maps background and mechanism are also provided as it involves with the proposed method of this study.

2.1.1 Computer Vision

According to [4], computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. The scope of computer vision is classified in [5] as following.

Access:	Retrieval of image data
Transfer:	Communication of image data
Convert:	Conversion into required format
Modify:	Applying filters, crop, transforms
Analyze:	Using vision to understand a scene

Computer vision can be applied to several domains such as object recognition for machine inspection and image retrieval, medical imaging, automotive safety, match move for movie shooting technique, surveillance and so forth [6]. In this regards, semantic technology is implemented as described in the next concept.

2.1.2 Digital Image Processing and Feature Extraction

Digital image processing comprises of many theories including image feature extraction which are foundation of image retrieval system. This study employs several feature extraction methods in order to create low-level feature vector of each image including image histogram, color auto-correlogram, color moments, Gabor wavelet and discrete wavelet transform.

(1) Image Histogram

Basically, histogram is a graph showing the number of pixels in an image at each different intensity value [7]. This technique is simple to use and easy to be calculated by software [8]. Image histogram can be applied to various color space including RGB, HSV, LAB and etc. However, HSV color space is selected for this research to represent distribution of each color composition. This is because the HSV color system is one of the most frequently used when people selects color as well as it is considerably closer to human perception [9].

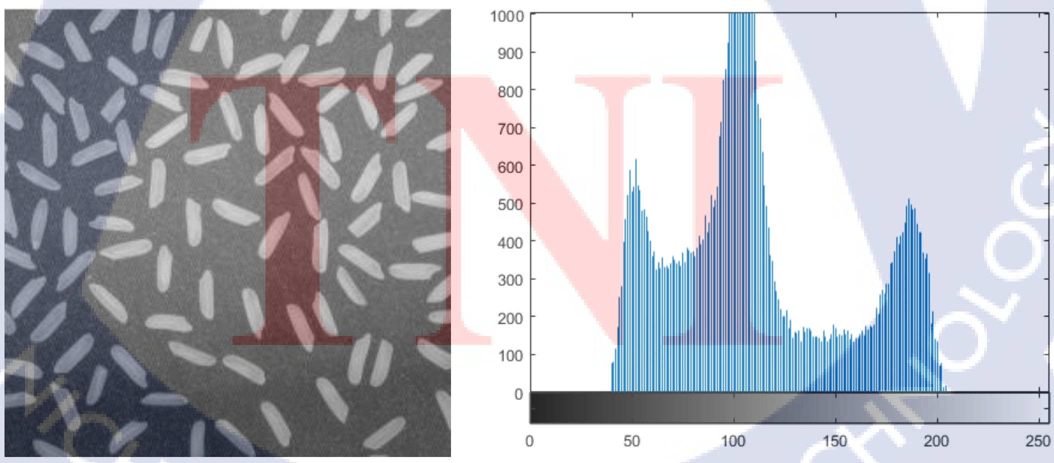

Figure 1.1 Example of Image Histogram [10]

(2) Color Auto-Correlogram

Color auto-correlogram is implemented as it allows to describe distribution and spatial correlation of color as shown in Figure 1.2 thus, the correlogram robustly tolerates large changes in appearance and shape caused by changes in viewing positions, camera zooms, etc. [11] [12]. This method is then calculated to strengthen the extracted low-level feature vector used in this research.

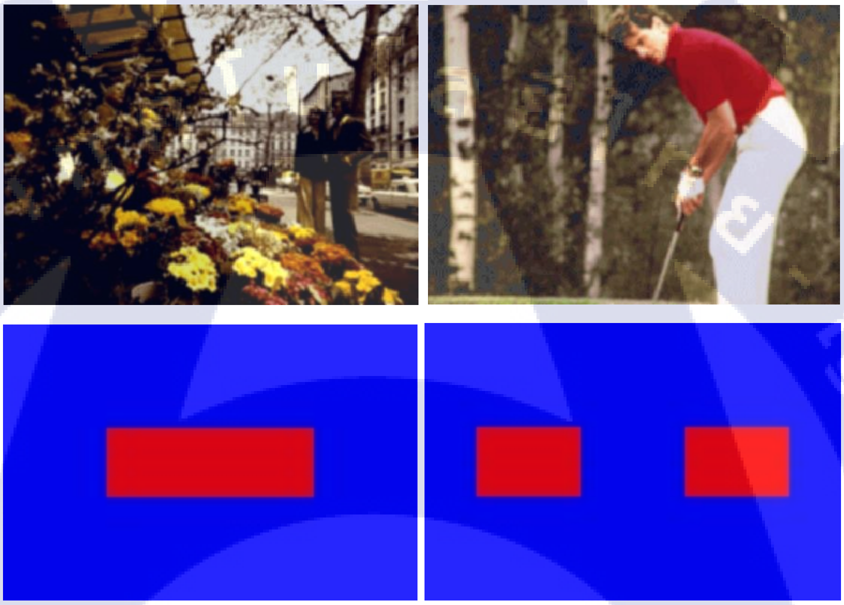

Figure 1.2 Example of Color Auto-Correlogram

(3) Color Moments

Although color features are acquired using histogram, moments are applied to extract color feature. Referring to [13], color moments are proved for its performance over normal histogram method. Two first color moments including mean and standard deviation are calculated for RGB color space to construct image low-level feature vector.

(4) Gabor Wavelet / Gabor Filters

Gabor wavelet is employed as it is suitable for image low-level feature extraction in terms of edge detection and object recognition [14] [15].

According to [13], the scale and orientation tunable property of Gabor filter makes it especially useful for texture analysis as it captures energy at a specific frequency and specific orientation. Thus, mean square energy and mean amplitude are calculated as another parameters to construct image low-level feature vector.

(5) Discrete Wavelet Transform (DWT)

According to [8], discrete wavelet transform is considered as multi-resolution analysis in time and frequency domain. This method has been proved for efficiency in several areas of image enhancement, classification, recognition and retrieval including [16] [17] [18] [19]. Thus, discrete wavelet transform is calculated in this study to extract wavelet coefficients and calculate the first two moments to construct image low-level feature vector.

2.1.3 Distance Metrics

The distance metrics are used to measure how close two elements are, where elements do not have to be numbers but can also be vectors or matrices. Thus, this distance function is used for data classification, clustering including calculation of similarity level between two objects. Regarding this matter, there are many distance functions available in common such as Euclidean distance, Mankowski, Mahalanobis and Manhattan distance [20]. The Manhattan distance is selected as distance method in this study as it produces the best result when measuring similarity between a query image and each image in database. Hence, this can enhance possibility to get correct relevant retrieved image which is a key dependency of the proposed Topic Maps-based method.

The Manhattan Distance focuses on sum of absolute difference (SAD) which can be represented as below [20].

$$d_{\mathbf{SAD}} : (x, y) \mapsto \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i|$$

Figure 1.3 Calculation of Manhattan Distance

This calculation is applied to calculate degree of similarity in this study so as to find images with closest distance.

2.1.4 Content-Based Image Retrieval

Content-based image retrieval or CBIR has been used as an alternative to text-based image retrieval with the ability to categorize images based on low-level features extracted from the image itself which recently are color, texture, shape and spatial relations [21]. Refer to [22], CBIR is employed in many areas including computer vision, pattern matching, cognitive psychology and many more in order to extract intrinsic image features suitable for automatic indexing and retrieval. The common elements of CBIR are arranged for four parts as referred to [23]; data collection, build up feature database, search database and index the results after researching. The low-level extracted feature of CBIR is exploited for diverse computing on many successful researches with different approaches as summarized in Table 1.1 below.

Table 1.1 Existing Researches and Related Low-Level Features Used

Low-level features	Researches	Approaches
Color	W. Niblack et al	Histogram and color moments
	Chad Carson et al	Region Histogram
	Wang	Dominant Color
	J. Sawhney & Hafner	Color Histogram
	Stricker & Orengo	Color Moment
	Kankanhalli et al	Color Cluster
	Wang	Big Signature
Shape	MichaelOrtega et a	Fourier Transform
	F. Mokhtarian et al	Curvature Scale Space
	Sougata Mukherjea	Template Matching
	Fumikazu Kanehara	Convex Parts
Texture	J. R. Smith	Wavelet Transform
	S. Michel	Edge Statistic
	B. S. Manjunath	Gabor Filters

Table 2.1 Existing Researches and Related Low-Level Features Used (Cont.)

Low-level features	Researches	Approaches
Texture	George Tzagkarakis & Panagiotis Tsakalides	Statistical

There are many researches done in the domain of cartoon character image retrieval by employing CBIR techniques. However, the approach is different. The training scheme is proposed in [24] to emphasize the significant regions or aspects based on the user's preferences and actions, such as selecting a desired image or an area for more retrieval accuracy. In [2], cartoon character image retrieval system is built with two applications; namely content-based cartoon image retrieval and cartoon clip synthesis exploiting Unsupervised Bi-Distance Metric Learning (UBDML). This research considers two main visual features which are Edge Features (EF) and Motion Direction Features (MDF).

Although CBIR can perform better result than traditional keyword search, other solutions for more accurate retrieval is examined. The reason is neither a single feature nor a combination of multiple visual features could fully capture high level concept of images which extremely associates with human perception [25]. In other words, CBIR considers only low-level visual features but ignores contextual expression and meaning which are the important factors for human perception. This semantic gap prevents the machine from retrieving comprehensive result while non-related excessive recalls are increased.

2.1.5 Semantic Technologies

Semantic Technologies are Web technologies that provide sophisticated, knowledge-driven platform for storing and manipulating information. They allow Websites and other Web resources to be understood and used by computer, as well as by humans [26]. According to [27] [28], semantic technologies are proved for their achievement in diverse business domains including public institution, e-Government, health care, IT industry, telecommunication, life sciences, energy, broadcasting, library and automotive. The area covers data integration, semantic search, semantic content discovery, semantic annotation, social networks, natural language interfaces, service integration as well as oil and gas exploration and production.

At first, Semantic Technologies are driven through Resource Description Framework or RDF which is the formal format of semantic-based data modelling. The basic RDF model is separated into three parts which are a resource (the subject) linked through an arc of association (the predicate) to a value (the object) [29] comprising a RDF triple. Many triples are linked across each other generating a data universe called “Triple Store”. However, this study does not directly employ the RDF, but using some other next generations of technologies developed based on the RDF concept which is called “Topic Maps” as described later in this chapter.

Another key enablers of Semantic Technologies which is also described later in the next part is ontologies, the embedded elements within the software systems, providing information and structure to other parts of the system such as databases, inference engines, web services, information agents and user interface tools [26].

2.1.6 Ontologies

Ontologies are widely used in knowledge engineering, artificial intelligence as well as applications related to knowledge management, information retrieval and the semantic web [30]. There are various definitions for the term “Ontology”. The most accepted one is referred in [31] which is given by Tom Gruber in 1993. He said that ontology is a specification of a representational vocabulary for shared domain of discourse definitions of classes, relations, functions and other objects resulting in logical conceptual network. To develop ontology, some basic practices are needed to be considered as described in [32]; properties of the domain, range, association rule and field as a whole. Using ontology can be varied in terms of ontological representation purpose, tools and techniques such as description logics, commonsense reasoning, Topic Maps technology and so forth. The third one is selected as the semantic technology applied in this research. Therefore, using ontology and ontological structured data allow semantic mechanism of knowledge representation enabling human to be able to manipulate machine data while maintaining well-organized data structure for machine processing.

2.1.7 Knowledge Representation and Annotation

Knowledge representation (KR) is an area of artificial intelligence research aimed at representing knowledge in symbols to facilitate inference from those knowledge elements, creating new elements of knowledge. The KR can be made to be independent of the underlying knowledge model or knowledge base system (KBS) such as a semantic network [33]. In the context of digital image retrieval, knowledge representation can be referred to image annotation. Annotation of multimedia data provides an opportunity for making the semantics explicit and facilitating multimedia retrieval [34]. For the best result, the annotation should be able to perform learning in order to create scalable network as all representations are imperfect, and any imperfection can be a source of error [35]. Thus, the annotation feature is introduced as a part of semantic-based image retrieval mechanism in order to allow human to manipulate retrieval process in more semantic manner and eventually improve retrieval performance in terms of accuracy.

2.1.8 Topic Maps

Topic Maps is an international industry standard or ISO13250 for information management and interchange [36]. Most applications of Topic Maps fall into four broad categories: enterprise information integration, knowledge management, e-learning and web publishing [37]. This is because the Topic Maps possesses some prominent functions including i). Indexing, ii). Classifying and organizing, iii). Navigating, filtering and visualizing, iv). Merging and v). Querying [38]. These properties allow the Topic Maps to be able to semantically structure and provide self-describing linked network of knowledge representation [39]. Besides, it can be utilized for navigating through information resources better than normal browsing through hyperlinks that are generally unstructured and often misleading [40]. Thus, there are plenty of researches employing Topic Maps approach to achieve their objective in several domains including medical analysis [41] [42] [43] [44], information and digital multimedia retrieval [36] [45] [46] [47] [48] [49] [50], e-learning and teaching aided system [36] [51], database management [52] and knowledge management [53] [54].

Topic Maps composes of three main elements which are topic, association and occurrence [55]. Topic is a very basic element which can be any subject or terms. Association is a relationship between each topic. Occurrence is some kinds of information resources which is relevant to some specific topics. Below is how these three elements look like when they are combined together [56].

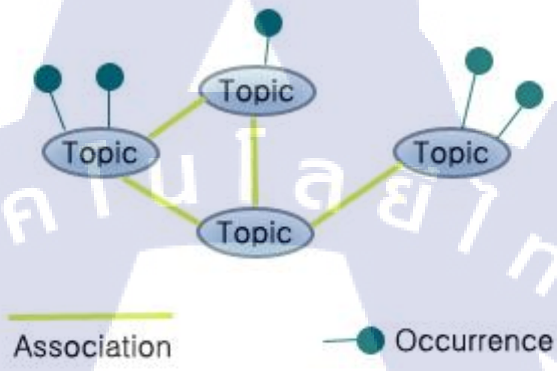

Figure 1.4 Composition of Topic Maps TAO

These elements can be represented by using ontological terms which will then become the Topic Maps resources. Each resource is considered as a topic no matter what type of word it is. This property of the Topic Maps is different from another ontological structured data i.e. commonsense reasoning because it combines several resources into one topic resulting in more compact information processing. Another remarkable function of the Topic Maps is that it allows to handle information in object-oriented way encouraging systematic information resources management and manipulation [55].

In short, the Topic Maps is a knowledge model representing all related ontological concepts. This study employs this model to retrieve relevant images by setting up a key ontological term on the topic Maps network then traversing along related association to get related results as many as possible. Regarding this approach, the Topic Maps is able to enhance retrieval performance in terms of precision.

2.2 Related Programs and Tools

The tools can be separated into 2 topics which are tools for the conventional CBIR method and tools for the proposed Topic Maps-based method.

For the conventional CBIR, the study uses a Matlab script developed and shared on [57] as this script can perform common task done by most CBIR systems.

For the Topic Maps-based method, Ontopia Knowledge Suite (OKS) [58] is selected as main tool to support the proposed concept and experiment. This tool is a java based open-source software suite comprising of three modules including:

- Ontopoly - ontology editor specifically for Topic Maps ontological terms,
- Omnigator - Topic Maps network navigator
- Vizigator - Topic Maps visualization engine

More detail and usage of tools are described later in the next related chapters.

2.3 Chapter Summary

Regarding the objective, this research involves with many theories. In terms of the conventional CBIR development, this requires some knowledge on digital image processing and distance metrics in order to create image retrieval system. On the other hand, for the proposed Topic Maps-based method, the principle domain includes Semantic Technologies comprising of ontologies, knowledge representation and annotation, and another most important knowledge as it directly relates to the proposed method of this study, the Topic Maps.

Chapter 3
Research Methodology

This chapter elaborates research methodology divided into three parts; system design, comparison approach and experimental design. The system design illustrates system components, flows and functionalities including some screenshots of related development software aiming to provide overall concept and functional detail of the proposed Topic Maps-based image retrieval system. Comparison approach presents how this study is going to conduct an experiment to compare between conventional CBIR and the proposed Topic Maps-based system. The other part is experimental design which is separated into five consecutive phases; infrastructure preparation, pre-processing, training, testing and evaluation phase aiming to provide steps of action to proceed through experiment and prove that the proposed method does optimize retrieved results.

3.1 System Design

In order to deal with the objective of this study, system analysis has been done so as to develop fundamental design of overall concept and functions of the proposed image retrieval system. Thus, this part illustrates several types of system model including use case, system elements, system diagram and functional flows as well as some screenshots of related development software.

3.1.1 Use Case Diagram

The proposed semantic-based image retrieval system allows user to do as following.

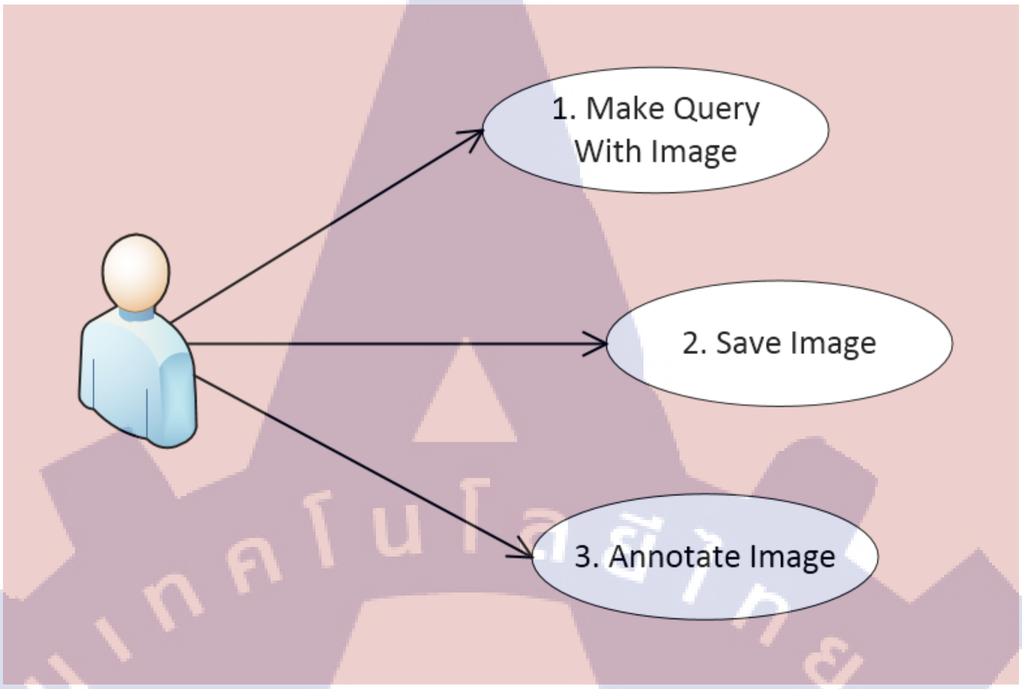

Figure 1.1 Use Case Diagram

Table 1.1 Use Case Description

No.	Case	Description
1	Make query with image	User is able to query for similar images using a query image.
2	Save image	User is able to save a new query image in case that it has not yet existed in database.
3	Annotate image	User is able to annotate query image by assigning Topic Maps ontological terms to that image in order to extend Topic-Maps network.

3.1.2 System Elements

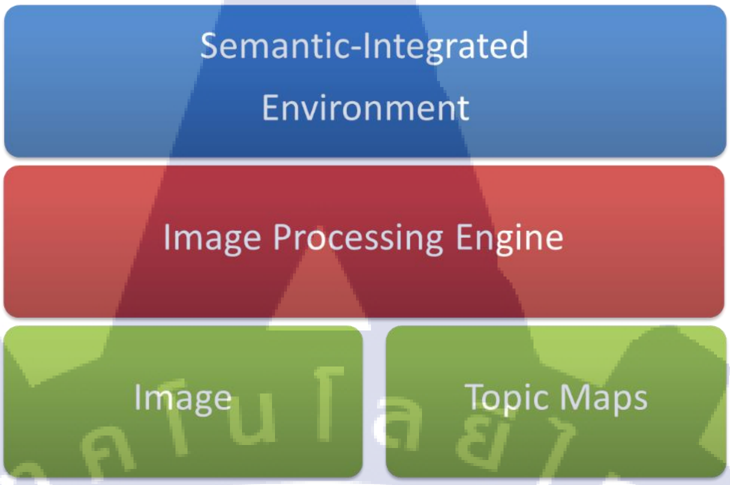

Figure 1.2 Overall system elements

The above diagram can be described in three layers using additional analogy to a car with the objective to provide clear concept of this architecture.

(1) Database Layer (Colored with green)

This layer is fuel. Without the fuel, the car is useless. Database provides matter used for being processed in the higher level including original image, image extracted feature, image feature annotation and Topic Maps ontology.

(2) Image Processing Layer (Colored with red)

The image processing engine is a kind of engine room generating power to the system. Any image input into the system will be processed in this layer for initial image analysis and segmentation for further image feature extraction. The main responsibility of this layer is to perform image analysis, segmentation, feature vector extraction and similarity matching.

(3) Semantic Layer (Colored with blue)

The car may not be safely and precisely driven to its destination without good steering control. Semantic technology is implemented to fulfill this demand. Using semantic technology, the system will be able to classify and retrieve image by its contextual expression not just visual similarity. Thus, semantic technology can increase level of retrieval precision while reducing non-related excessive recall.

3.1.3 System Diagram

System diagram is illustrated to provide overall concept of system functional flows as well as interaction between user and the system.

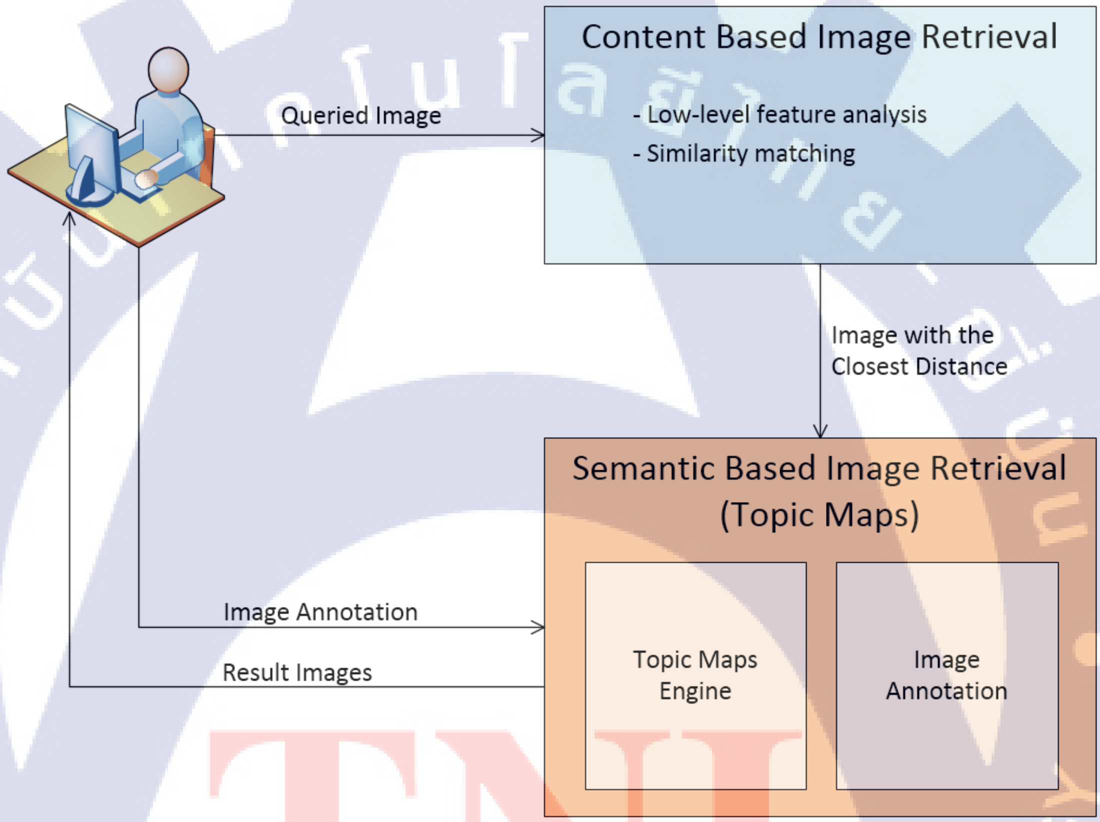

Figure 1.3 System Diagram

According to Figure 1.3, the system composes of two main modules which are conventional low-level feature analysis (CBIR) and the proposed Topic Maps-based semantic engine. The system can be manipulated using a query image. The query image is processed by CBIR engine to create low-level feature vector and calculate similarity distance before passing an image with the closest distance to semantic engine. Semantic engine then gets ontological term of the passed image and traverses along related association in Topic Maps network in order to find result images. Apart from normal system operation, user can provide image annotation to query image if required as well as save the image for next retrieval. More detail regarding system functions is described later in next part.

3.1.4 System Functional Flows

Detailed functional flows are provided to describe process and system activities.

Figure 3.4 illustrates system activities and flow across two modules inside the system which composes of content-based image retrieval engine and the proposed Topic Maps-based system. Thus, the process description is separated into two parts following these two modules.

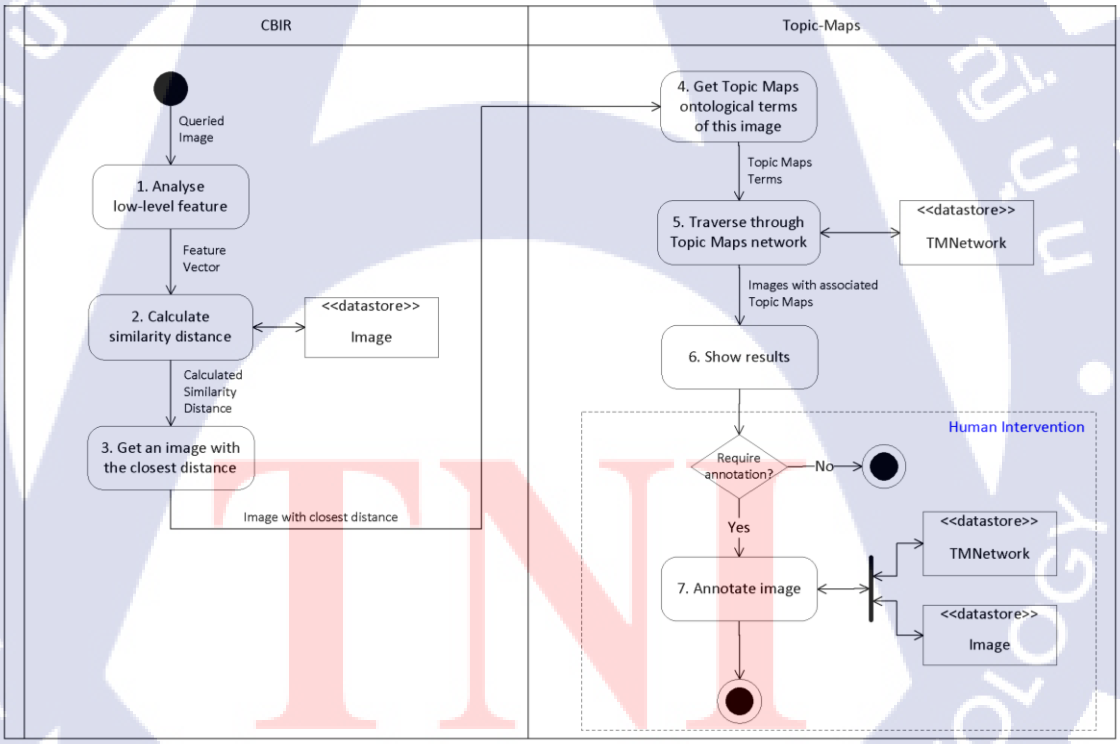

Figure 1.4 Overall System Process

Process Description of System Functional Flows

Table 1.2 Conventional Content-Based Image Retrieval Operations

No.	Process	Description
1	Analyze low-level feature	After image is input, CBIR engine will extract image low-level features and construct a 1x191 feature vector (See Appendix1) composing of HSV histogram, spatial color auto-correlogram, color moments, Gabor wavelet and wavelet transform
2	Calculate similarity distance	CBIR calculates the feature vector of query image and feature vector of image in database using Manhattan distance. For calculation detail, please see Formula X in chapter 2.
3	Get an image with the closest distance	The calculated results of all images are sorted in ascending order. The first 20 images with the closest distance are displayed but only the first result is selected to be processed further in the Topic Maps-based engine.

Table 1.3 Topic Maps-Based Image Retrieval Operations

No.	Process	Description
4	Get Topic Maps ontological terms of this image	The first retrieved result of CBIR is selected as a key image to get an ontological which is hereby an image name. The image name is associated with other ontological instances including character name.
5	Traverse through Topic Maps network	The key ontological term acquired from step 4 is used as a beginning point to traverse through Topic Maps association and related instances in order to find relevant images.
6	Show results	Semantic engine displays relevant results. This is the end of automatic mechanism of the system.

Table 3.3 Topic Maps-Based Image Retrieval Operations (Cont.)

No.	Process	Description
7	Annotate image	This process requires human intervention by considering if the results are correct or not. If no, user can annotate query image and save both image and annotation terms back to database for next time retrieval. If the results are correct then, no need to do annotation.

3.2 Comparison Approach

Comparison approach is provided to present how this study is going to conduct an experiment to compare between conventional CBIR and the proposed Topic Maps-based system.

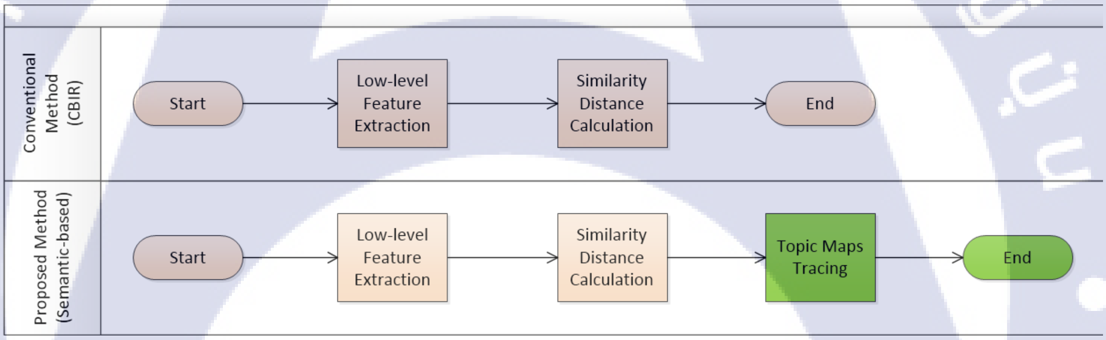

Figure 1.5 Concept Overview of Comparison Approach

According to above figure, conventional method only proceeds through low-level based analysis while the proposed Topic Maps-based method employs one more process which is traversing along Topic Maps network. The results of each method are observed at different point as shown in Figure 1.6.

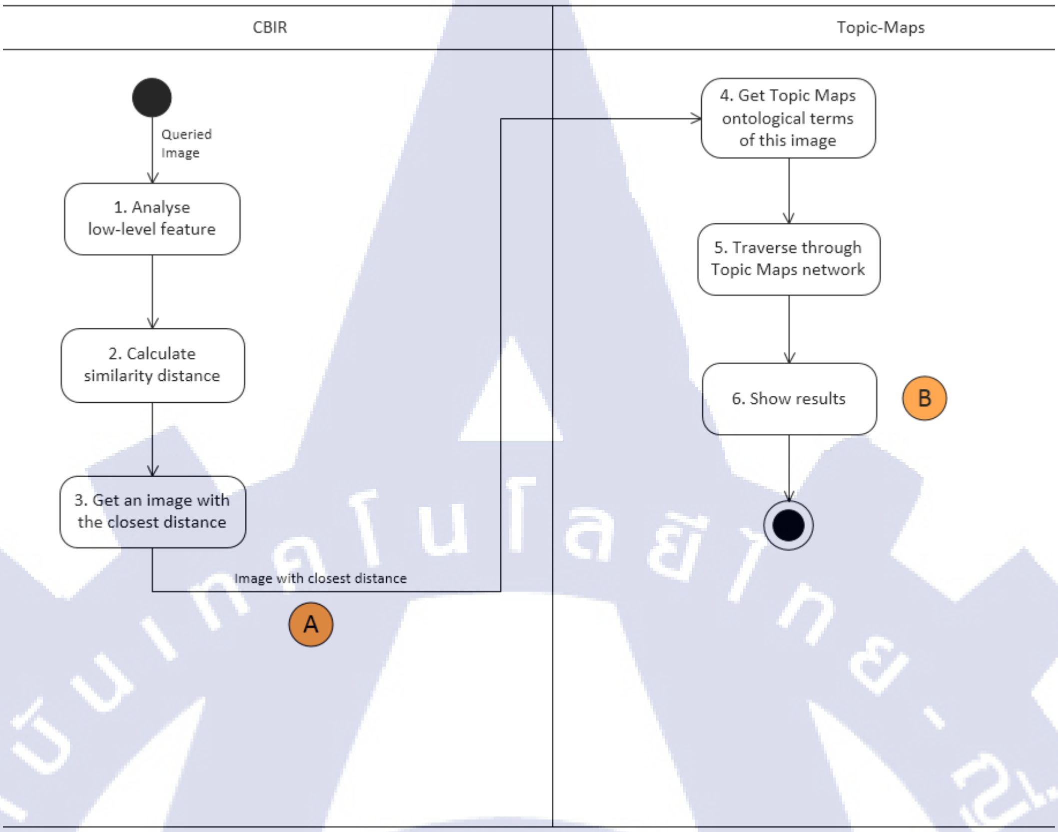

Figure 1.6 Comparison Approach at Specific Point

According to above figure, the result is observed once at point A to get the result of conventional CBIR method. Next, the result is observed again at point B to get the result of the proposed Topic Maps-based method. These results are used for precision and recall calculation for further assessment which is described later in Chapter 4.

3.3 Experimental Design

The experiment has been done with the objective to provide steps of action to proceed through retrieval mechanism and prove that the proposed method does optimize retrieved results. Regarding this matter, the experiment is designed as five phases; infrastructure preparation, pre-processing, training, testing and evaluation which detail as followings.

3.3.1 Infrastructure Preparation Phase

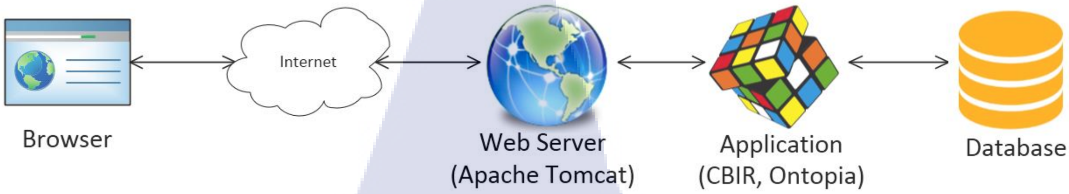

Figure 1.7 System Infrastructure

This infrastructure is divided into two parts regarding backend and frontend. The backend includes database and application service hosting main applications used which are CBIR engine and Ontopia. The frontend includes web server for java servlet calling the application which is displayed on normal web browser.

3.3.2 Pre-Processing Phase

Some pre-processing activities are required for the conventional CBIR method including:

- Rename image to numerical numbers
- Resize image to 384 x256
- Apply solid white background

3.3.3 Training Phase

Training involves some activities regarding database creation for both the conventional CBIR and the proposed Topic Maps engine.

For the conventional CBIR, database is constructed with image low-level feature vectors using an open-source Matlab-based program which is originally developed and distributed by Chez on Matlab File Exchange Community [51] as mentioned in chapter 2. The logical process of this operation can be illustrated as shown in Figure 1.8.

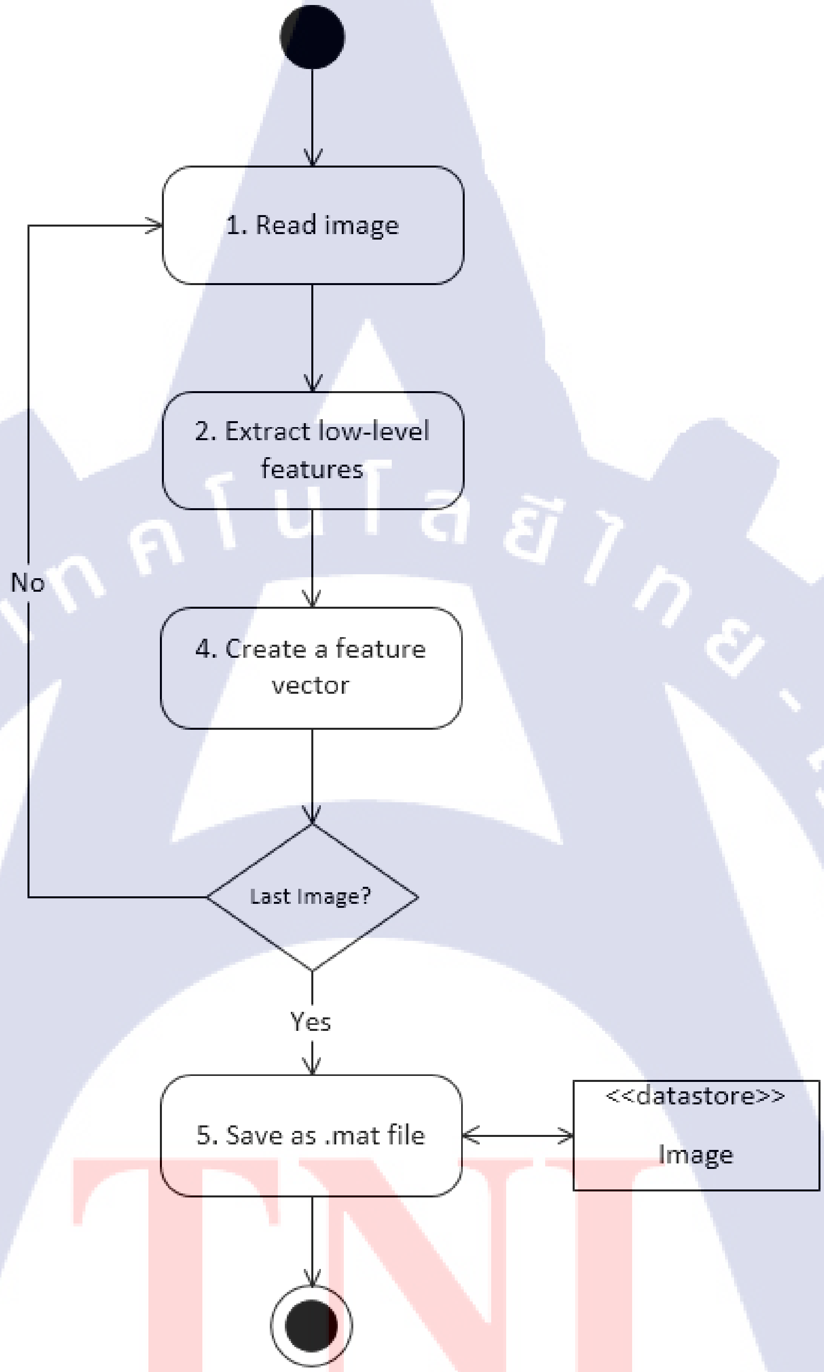

Figure 1.8 Conventional CBIR Database Creation Process

Table 1.4 Process Description of CBIR Database Acquisition

No.	Process	Description
1	Read image	Read image into the program
2	Extract low-level features	Extract image low-level features including: - HSV Histogram - Color Moments - Color Auto Correlogram - Gabor Wavelet - Wavelet Transformation See **Error! Reference source not found.** for more detail.
3	Create a feature vector	Combine extracted features into a vector
4	Save as .mat file	After feature extraction has been done to all images, the program will ask to save data into .mat file which can be used as dataset.

The above operations are conducted using simple GUI [51] with functions as below.

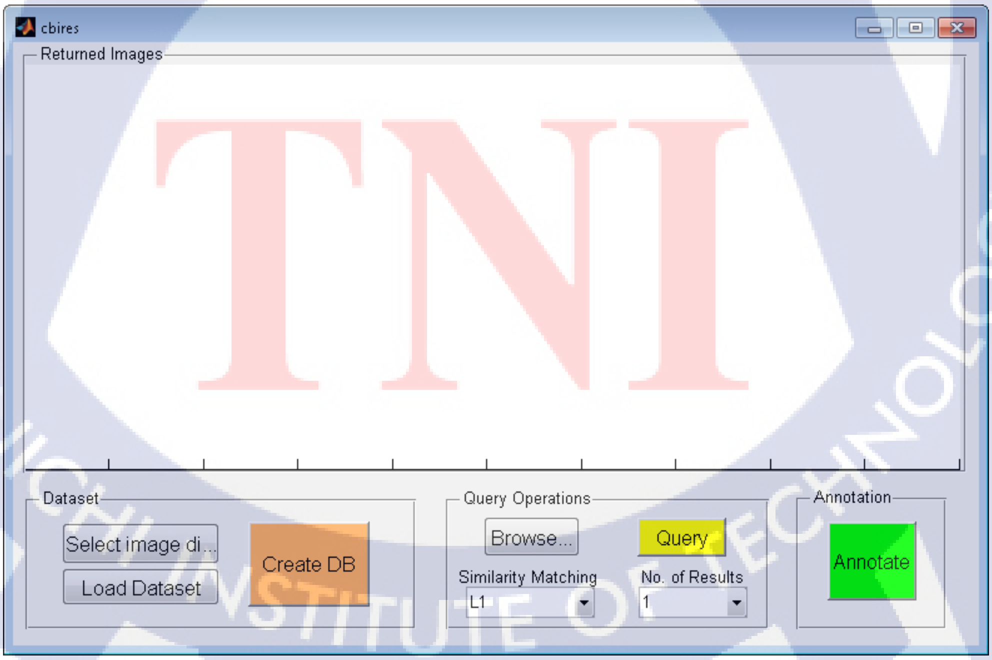

Figure 1.9 GUI for CBIR

Table 1.5 Screen Definition for Conventional CBIR Operations

No.	Type	Name	Description
1	Data Element	Returned Images	Space for displaying result images
2	Action	Select Image Directory	Select image directory to extract features.
3	Action	Create Database File	Create .mat file from extracted low-level feature vector.
4	Action	Load Dataset	Load .mat file to be used as dataset.
5	Action	Browse Image	Browse for query image.
6	Action	Query	Make query.
7	Action	Similarity Matching	Select distance method (Manhattan Distance)
8	Action	No. of Results	No. of Results to be retrieved (max.20)
9	Action	Annotate Image	Make annotation to query image which is currently used.

On the other hand, the proposed Topic Maps-based experiment requires ontological Topic Maps network as a dataset. This dataset is constructed using Ontopoly, one of Topic Maps manipulation tools available in Ontopia Knowledge Suite. There are 4 steps to create ontological terms and associations including; create main class, create instances of the class, create association term and make relationship, assign correct instance to the association.

(1) Create main topic

According to this research scope, the ontological terms which are possibly used as main classes are for example "character name" and "image name". Thus, these main classes are created via Ontopoly UI as below. Note that "Subject Identifier" is mandatory as a unique reference.

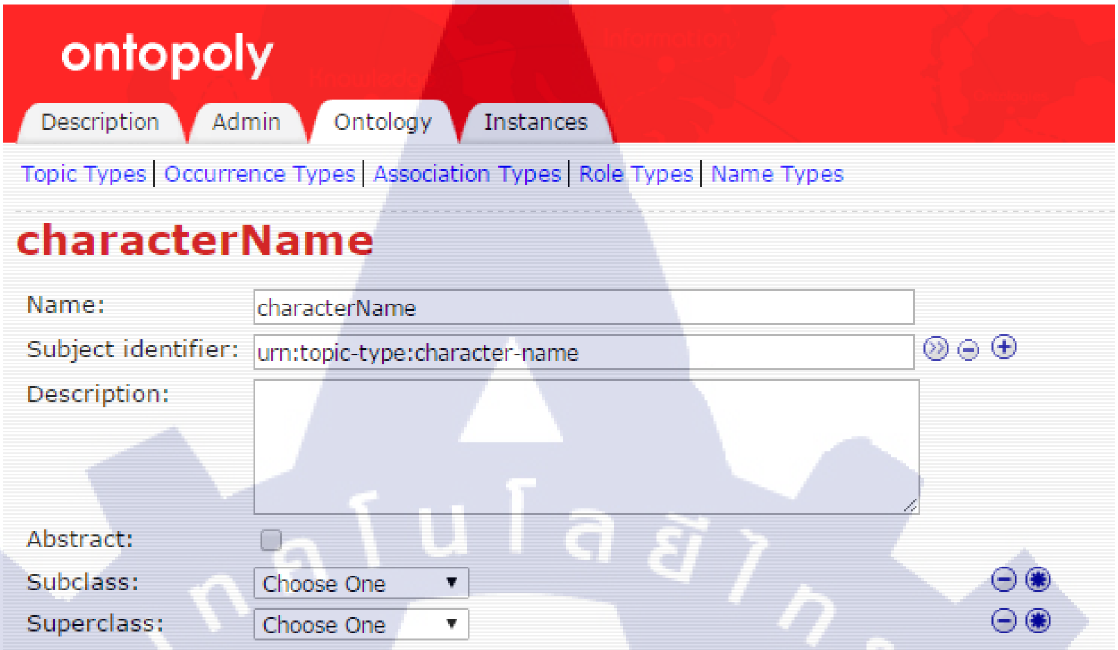

Figure 1.10 Ontopoly - Main Class for "Character Name"

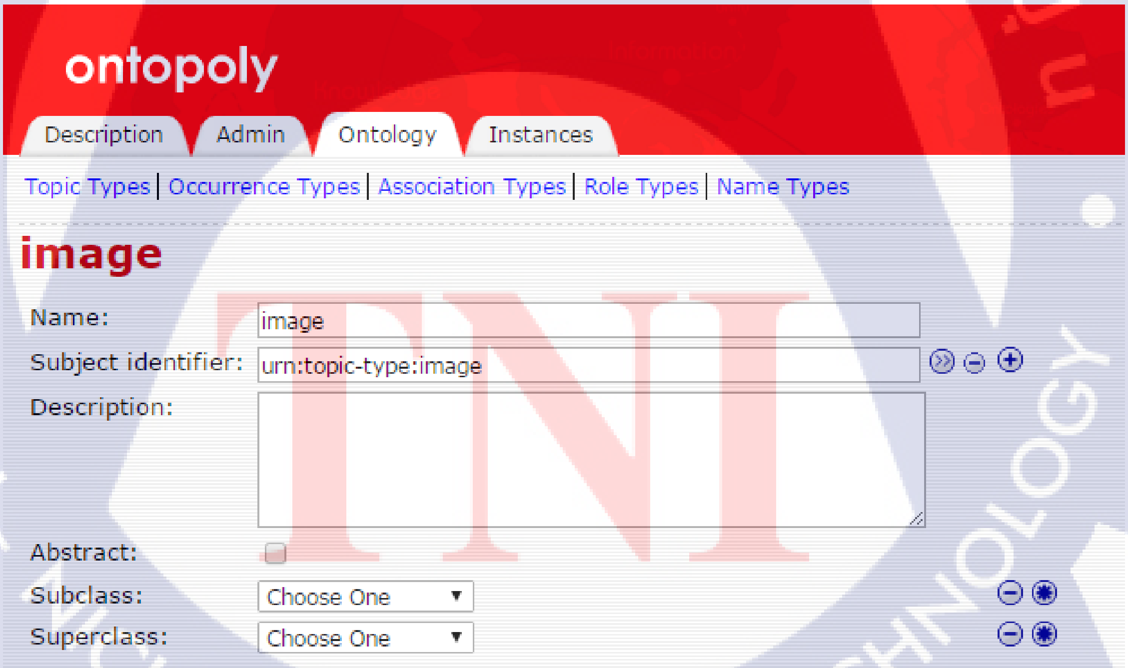

Figure 1.11 Ontopoly - Main Class for "Image Name"

(2) Create instance of the topic

Name of characters and images are created as an instance of the main class as shown in Figure 1.12 and Figure 1.13.

Figure 1.12 Ontopoly - Instances of "Character Name"

Figure 1.13 Ontopoly - Instances of "Image Name"

(3) Create association term and make relationship

Association is a key of semantic-based image retrieval mechanism as it allows information to be manipulated in relational manner. Association term and its relationship is created as below.

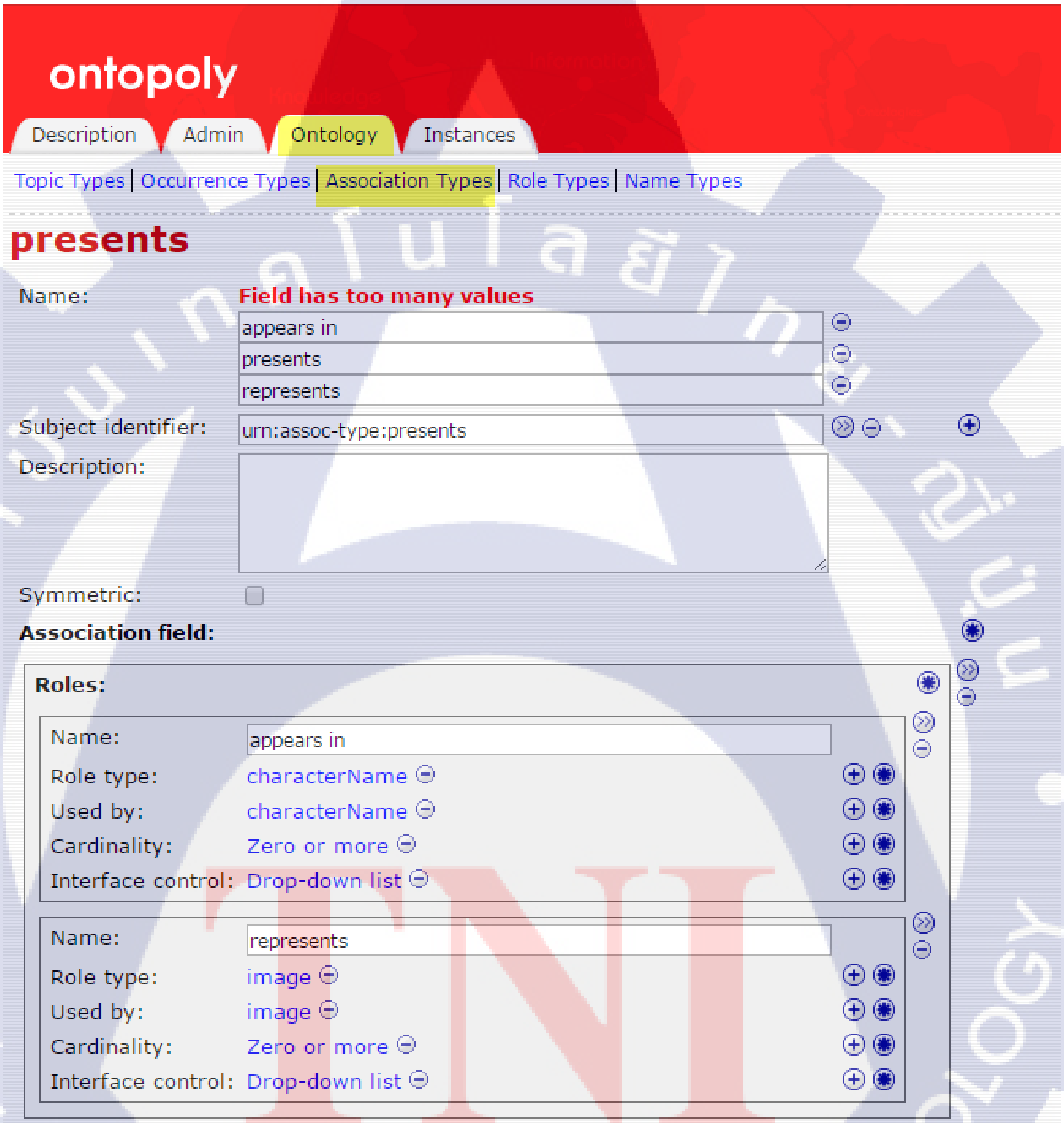

Figure 1.14 Ontopoly - Association of "Presents"

According to above snapshot, the main association term is “presents” linked between terms “characterName” and “image”. However, Ontopoly allows to create specific term for each scope of use for example if this association is used by a character name then it displays as "appears in" but if it is used from image point of view then the association term displays as "represents" in order to best suit for semantic implication.

(4) Assign correct instance to the association

To enable association between each term, some instances or classes must be assigned as shown below.

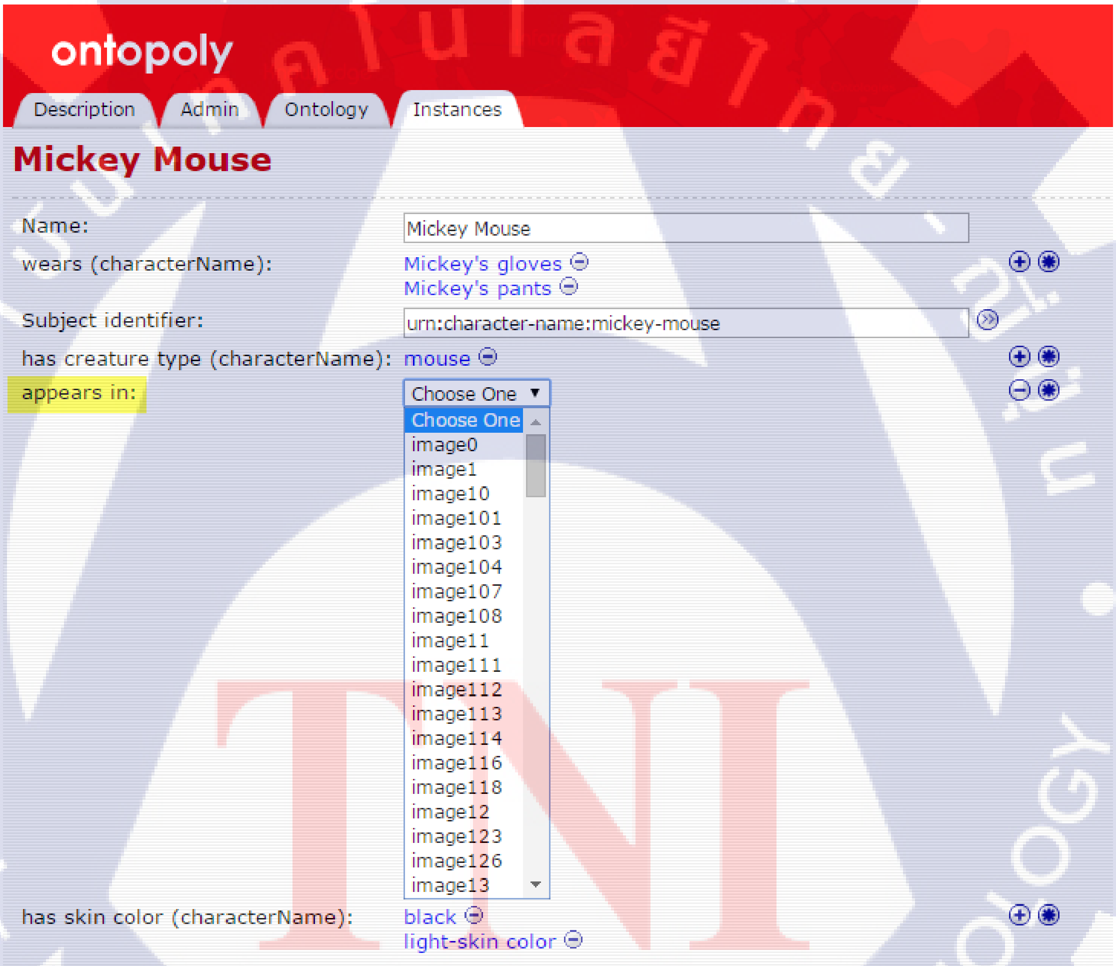

Figure 1.15 Ontopoly - Association of "presents" from Character Point of View

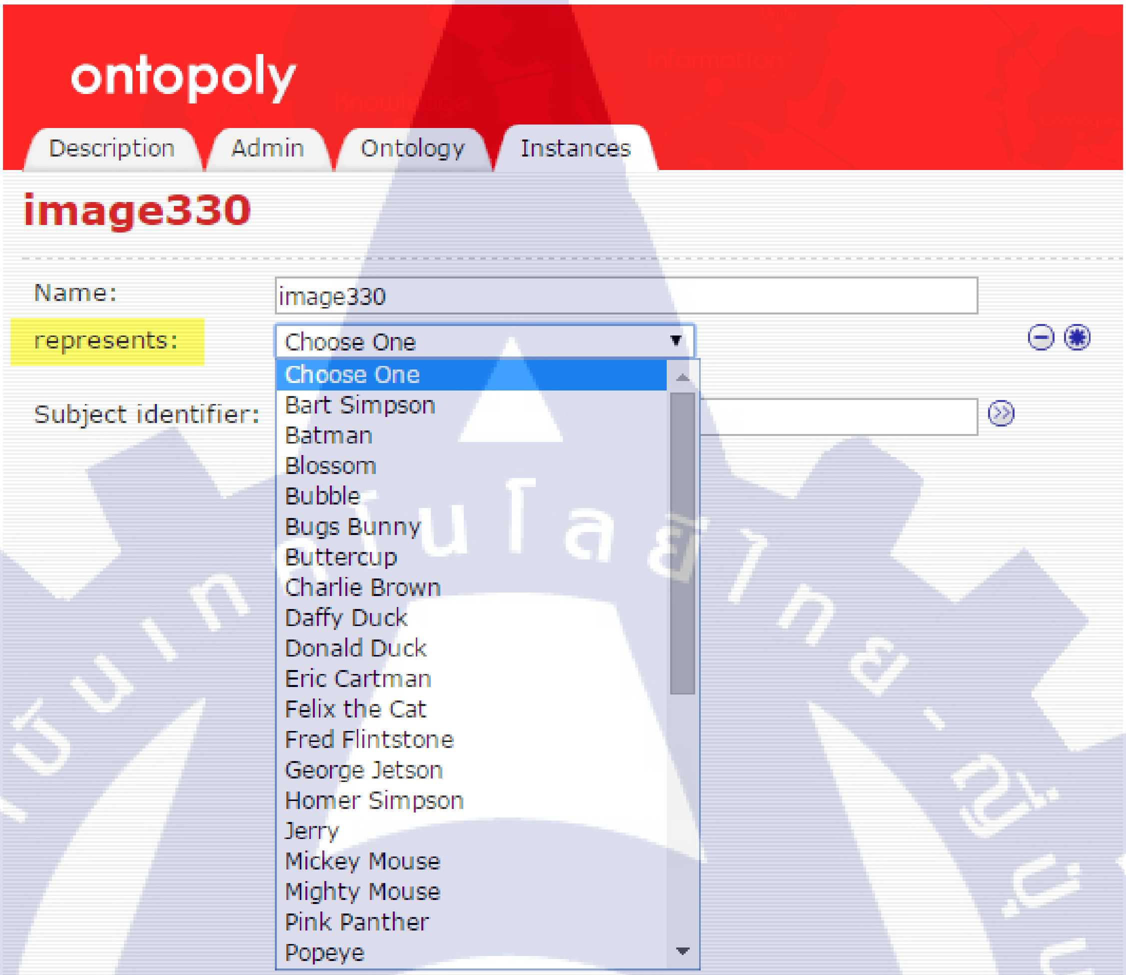

Figure 1.16 Ontopoly - Association of "presents" from Image Point of View

After completing to assign all topics and associations, a network of Topic Maps is constructed as shown in Figure 1.17. This Topic Maps network can be used as a dataset for the Topic Maps engine.

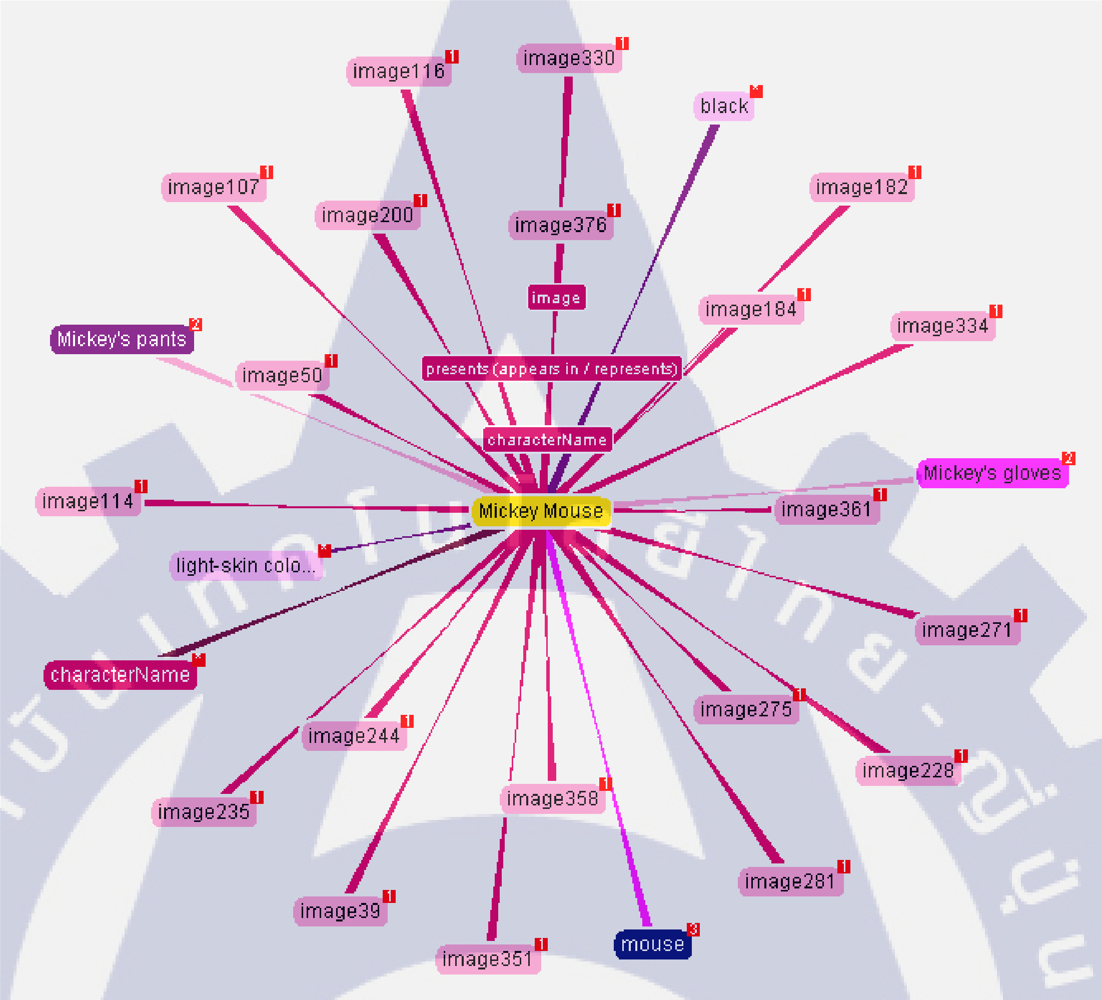

Figure 1.17 Topic Maps Network - Semantic-Based Dataset

Besides, this ontological network is fundamentally driven by LTM or Linear Topic Maps Notation which is the XML-based topic map interchange format representing the constructs in the Topic Maps standard as text [59]. Thus, LTM is considered as a core of Topic Maps ontological-structured data enhancing machine logical processing while maintaining readability for human. The LTM used in this study can be represented as below.

Table 1.6 Core LTM Structure

```
#VERSION "1.3"
/*
   Generator: Ontopia
   Date:      2014-03-06 18:16
*/

/* ----------------- TOPIC MAP ----------------- */

#TOPICMAP~ id64
[id64 = "cartoononto_test2-copy13"
    @"file:/D:/ontopia-5.3.0/apache-tomcat/webapps/ontopoly/WEB-
INF/config/ontopoly-ontology.xtm#reified-id206"
    @"file:/D:/ontopia-5.3.0/apache-tomcat/webapps/ontopoly/WEB-
INF/config/ontopoly-ontology.xtm#reified-id68"]

/* ----------------- ONTOLOGY ------------------ */

/* ----------------- Topic Types --------------- */

[animal = "animal"
    @"urn:creature:animal"]
[belt = "belt"
    @"urn:costume-item:belt"]
[body-suit = "body suit"
    @"urn:costume-item:body-suit"]
[bow-tie = "bow tie"
    @"urn:costume-item:bow-tie"]
[character-name = "characterName"
    @"urn:topic-type:character-name"]
[cloak = "cloak"
    @"urn:costume-item:cloak"]
[collar = "collar"
    @"urn:costume-item:collar"]
[color = "color"
    @"urn:topic-type:color"]
[geometric-shape = "geometricShape"
    @"urn:topic-type:geometric-shape"]
```

```
[glasses = "glassess"
    @"urn:costume-item:glasses"]
[gloves = "gloves"
    @"urn:costume-item:gloves"]
[hat = "hat"
    @"urn:costume-item:hat"]
[helmet = "helmet"
    @"urn:costume-item:helmet"]
[high-top-boot = "high-top boot"
    @"urn:costume-item:high-top-boot"]
[human = "human"
    @"urn:creature:human"]
[image = "image"
    @"urn:topic-type:image"]
[mask = "mask"
    @"urn:costume-item:mask"]
[necktie = "necktie"
    @"urn:costume-item:necktie"]
[pants = "pants"
    @"urn:costume-item:pants"]
[pipe = "pipe"
    @"urn:costume-item:pipe"]
[shirt = "shirt"
    @"urn:costume-item:shirt"]
[shoes = "shoes"
    @"urn:costume-item:shoes"]
[socks = "socks"
    @"urn:costume-item:socks"]
[suit = "suit"
    @"urn:costume-item:suit"]
[vest = "vest"
    @"urn:costume-item:vest"]

/* ----------------- Type Hierarchy ------------ */

superclass-subclass( costume-item : superclass, belt : subclass )
superclass-subclass( costume-item : superclass, body-suit :
subclass )
```

```
superclass-subclass( costume-item : superclass, bow-tie : subclass
)
superclass-subclass( costume-item : superclass, cloak : subclass )
superclass-subclass( costume-item : superclass, collar : subclass )
superclass-subclass( costume-item : superclass, glasses : subclass
)
superclass-subclass( costume-item : superclass, gloves : subclass )
superclass-subclass( costume-item : superclass, hat : subclass )
superclass-subclass( costume-item : superclass, helmet : subclass )
superclass-subclass( costume-item : superclass, high-top-boot :
subclass )
superclass-subclass( costume-item : superclass, id92 : subclass )
superclass-subclass( costume-item : superclass, mask : subclass )
superclass-subclass( costume-item : superclass, necktie : subclass
)
superclass-subclass( costume-item : superclass, pants : subclass )
superclass-subclass( costume-item : superclass, pipe : subclass )
superclass-subclass( costume-item : superclass, shirt : subclass )
superclass-subclass( costume-item : superclass, shoes : subclass )
superclass-subclass( costume-item : superclass, socks : subclass )
superclass-subclass( costume-item : superclass, suit : subclass )
superclass-subclass( costume-item : superclass, vest : subclass )
superclass-subclass( creature : superclass, animal : subclass )
superclass-subclass( creature : superclass, human : subclass )

/* ----------------- Role Types ---------------- */

[costume-item = "costumeItem"
    @"urn:topic-type:costume-item"]
[creature = "creature"
    @"urn:topic-type:creature"]
[subclass = "Subclass"
    @"http://www.topicmaps.org/xtm/1.0/core.xtm#subclass"
    @"http://psi.ontopia.net/ontology/subclass"]
[superclass = "Superclass"
    @"http://www.topicmaps.org/xtm/1.0/core.xtm#superclass"
    @"http://psi.ontopia.net/ontology/superclass"]
```

```
/* ----------------- Association Types --------- */

[has-costumecolor = "has costume color"
                  = "used in" / color
                  = "colored with" / costume-item
    @"urn:assoc-type:has-costumecolor"]
[has-creature-type = "has creature type"
                   = "is a" / character-name
                   = "is type of" / creature
    @"urn:assoc-type:has-creature-type"]
[has-skincolor = "has skin color"
               = "has skin color of" / character-name
               = "used as skin color of" / color
    @"urn:assoc-type:has-skincolor"]
[presents = "presents"
    @"urn:assoc-type:presents"]
[superclass-subclass = "Superclass/subclass"
                     = "Subclass of" / subclass
                     = "Superclass of" / superclass
    @"http://psi.ontopia.net/ontology/superclass-subclass"
    @"http://www.topicmaps.org/xtm/1.0/core.xtm#superclass-
subclass"]
[wears = "wears"
       = "wear" / character-name
       = "worn by" / costume-item
    @"urn:assoc-type:wears"]

/* ----------------- Occurrence Types ---------- */

/* ----------------- INSTANCES ----------------- */

/* ----------------- Topics -------------------- */


/* -- TT: animal -- */
[bear : animal = "bear"
    @"urn:animal:bear"]
[bird : animal = "bird"
```

```
    @"urn:animal:bird"]
[cat : animal = "cat"
    @"urn:animal:cat"]
[dog : animal = "dog"
    @"urn:animal:dog"]
[duck : animal = "duck"
    @"urn:animal:duck"]
[moose : animal = "moose"
    @"urn:animal:moose"]
[mouse : animal = "mouse"
    @"urn:animal:mouse"]
[panther : animal = "panther"
    @"urn:animal:panther"]
[pig : animal = "pig"
    @"urn:animal:pig"]
[rabbit : animal = "rabbit"
    @"urn:animal:rabbit"]
[sponge : animal = "sponge"
    @"urn:animal:sponge"]
[squirrel : animal = "squirrel"
    @"urn:animal:squirrel"]

/* -- TT: belt -- */
[batmans-belt : belt = "Batman's belt"
    @"urn:costume-item:belt:batmans-belt"]
[georges-belt : belt = "George's belt"
    @"urn:costume-item:belt:georges-belt"]
[mightymouses-belt : belt = "Mighty Mouse's belt"
    @"urn:costume-item:belt:mightymouses-belt"]
[supermans-belt : belt = "Superman's belt"
    @"urn:costume-item:belt:supermans-belt"]

/* -- TT: body-suit -- */
[batmans-body-suit : body-suit = "Batman's body suit"
    @"urn:costume-item:body-suit:batmans-body-suit"]
[mightymouses-body-suit : body-suit = "Mighty Mouse's body suit"
    @"urn:costume-item:body-suit:mightymouses-body-suit"]
[spidermans-body-suit : body-suit = "Spider-Man's body suit"
```

```
    @"urn:costume-item:body-suit:spidermans-body-suit"]
[supermans-body-suit : body-suit = "Superman's body suit"
    @"urn:costume-item:body-suit:supermans-body-suit"]
[underdogs-body-suit : body-suit = "Underdog's body suit"
    @"urn:costume-item:body-suit:underdogs-body-suit"]

/* -- TT: bow-tie -- */
[porkys-bow-tie : bow-tie = "Porky's bowtie"
    @"urn:costume-item:bow-tie:porkys-bow-tie"]

/* -- TT: character-name -- */
[bart-simpson : character-name = "Bart Simpson"
    @"urn:character-name:bart-simpson"]
[batman : character-name = "Batman"
    @"urn:character-name:batman"]
[bugs-bunny : character-name = "Bugs Bunny"
    @"urn:character-name:bugs-bunny"]
[bullwinkle : character-name = "Rocky and Bullwinkle (Bullwinkle)"
    @"urn:character-name:bullwinkle"]
[charlie-brown : character-name = "Charlie Brown"
    @"urn:character-name:charlie-brown"]
[daffy-duck : character-name = "Daffy Duck"
    @"urn:character-name:daffy-duck"]
[donald-duck : character-name = "Donald Duck"
    @"urn:character-name:donald-duck"]
[eric-cartman : character-name = "Eric Cartman"
    @"urn:character-name:eric-cartman"]
[felix-the-cat : character-name = "Felix the Cat"
    @"urn:character-name:felix-the-cat"]
[fred-flintstone : character-name = "Fred Flintstone"
    @"urn:character-name:fred-flintstone"]
[george-jetson : character-name = "George Jetson"
    @"urn:character-name:george-jetson"]
[homer-simpson : character-name = "Homer Simpson"
    @"urn:character-name:homer-simpson"]
[jerry : character-name = "Jerry"
    @"urn:character-name:jerry"]
[mickey-mouse : character-name = "Mickey Mouse"
```

```
    @"urn:character-name:mickey-mouse"]
[mighty-mouse : character-name = "Mighty Mouse"
    @"urn:character-name:mighty-mouse"]
[pink-panther : character-name = "Pink Panther"
    @"urn:character-name:pink-panther"]
[popeye : character-name = "Popeye"
    @"urn:character-name:popeye"]
[porky-pig : character-name = "Porky Pig"
    @"urn:character-name:porky-pig"]
[rocky : character-name = "Rocky and Bullwinkle (Rocky)"
    @"urn:character-name:rocky"]
[scooby-doo : character-name = "Scooby Doo"
    @"urn:character-name:scooby-doo"]
[speed-racer : character-name = "Speed Racer"
    @"urn:character-name:speed-racer"]
[spider-man : character-name = "Spider-Man"
    @"urn:character-name:spider-man"]
[spongebob : character-name = "SpongeBob SquarePants"
    @"urn:character-name:spongebob"]
[superman : character-name = "Superman"
    @"urn:character-name:superman"]
[sylvester : character-name = "Sylvester"
    @"urn:character-name:sylvester"]
[the-grinch : character-name = "The Grinch"
    @"urn:character-name:the-grinch"]
[the-powerpuff-girls : character-name = "The Powerpuff Girls"
    @"urn:character-name:the-powerpuff-girls"]
[tom : character-name = "Tom"
    @"urn:character-name:tom"]
[top-cat : character-name = "Top Cat"
    @"urn:character-name:top-cat"]
[tweety : character-name = "Tweety"
    @"urn:character-name:tweety"]
[underdog : character-name = "Underdog"
    @"urn:character-name:underdog"]
[winnie-the-pooh : character-name = "Winnie the Pooh"
    @"urn:character-name:winnie-the-pooh"]
```

```
/* -- TT: cloak -- */
[batmans-cloak : cloak = "Batman's cloak"
    @"urn:costume-item:cloak:batmans-cloak"]
[mightymouses-cloak : cloak = "Mighty Mouse's cloak"
    @"urn:costume-item:cloak:mightymouses-cloak"]
[supermans-cloak : cloak = "Superman's cloak"
    @"urn:costume-item:cloak:supermans-cloak"]
[underdogs-cloak : cloak = "Underdog's cloak"
    @"urn:costume-item:cloak:underdogs-cloak"]

/* -- TT: collar -- */
[scoobys-collar : collar = "Scooby's collar"
    @"urn:costume-item:collar:scoobys-collar"]

/* -- TT: color -- */
[black : color = "black"
    @"urn:color:black"]
[blue : color = "blue"
    @"urn:color:blue"]
[brown : color = "brown"
    @"urn:color:brown"]
[dark-blue : color = "dark blue"
    @"urn:color:dark-blue"]
[green : color = "green"
    @"urn:color:green"]
[grey : color = "grey"
    @"urn:color:grey"]
[light-skin-color : color = "light-skin color"
    @"urn:color:light-skin-color"]
[orange : color = "orange"
    @"urn:color:orange"]
[pink : color = "pink"
    @"urn:color:pink"]
[red : color = "red"
    @"urn:color:red"]
[violet : color = "violet"
    @"urn:color:violet"]
[white : color = "white"
```

```
    @"urn:color:white"]
[yellow : color = "yellow"
    @"urn:color:yellow"]

/* -- TT: geometric-shape -- */
[arc : geometric-shape = "arc"
    @"urn:geometric-shape:arc"]
[circle : geometric-shape = "circle"
    @"urn:geometric-shape:circle"]
[crescent : geometric-shape = "crescent"
    @"urn:geometric-shape:crescent"]
[diamond : geometric-shape = "diamond"
    @"urn:geometric-shape:diamond"]
[heart : geometric-shape = "heart"
    @"urn:geometric-shape:heart"]
[hexagon : geometric-shape = "hexagon"
    @"urn:geometric-shape:hexagon"]
[octagon : geometric-shape = "octagon"
    @"urn:geometric-shape:octagon"]
[oval : geometric-shape = "oval"
    @"urn:geometric-shape:oval"]
[rectangle : geometric-shape = "rectangle"
    @"urn:geometric-shape:rectangle"]
[square : geometric-shape = "square"
    @"urn:geometric-shape:square"]
[star : geometric-shape = "star"
    @"urn:geometric-shape:star"]
[trapezoid : geometric-shape = "trapezoid"
    @"urn:geometric-shape:trapezoid"]
[triangle : geometric-shape = "triangle"
    @"urn:geometric-shape:triangle"]

/* -- TT: glasses -- */
[rockys-glasses : glasses = "Rocky's glassess"
    @"urn:costume-item:glasses:rockys-glasses"]

/* -- TT: gloves -- */
[bullwinkles-gloves : gloves = "Bullwinkle's gloves"
```

```
    @"urn:costume-item:gloves:bullwinkles-gloves"]
[mickeys-gloves : gloves = "Mickey's gloves"
    @"urn:costume-item:gloves:mickeys-gloves"]
[mightymouses-gloves : gloves = "Might Mouse's gloves"
    @"urn:costume-item:gloves:mightymouses-gloves"]
[porkys-gloves : gloves = "Porky's gloves"
    @"urn:costume-item:gloves:porkys-gloves"]
[speedracers-gloves : gloves = "Speed Racer's gloves"
    @"urn:costume-item:gloves:speedracers-gloves"]

/* -- TT: hat -- */
[donalds-hat : hat = "Donald's hat"
    @"urn:costume-item:hat:donalds-hat"]
[erics-hat : hat = "Eric's hat"
    @"urn:costume-item:hat:erics-hat"]
[thegrinchs-hat : hat = "The Grinch's hat"
    @"urn:costume-item:hat:thegrinchs-hat"]
[topcats-hat : hat = "Top Cat's hat"
    @"urn:costume-item:hat:topcats-hat"]

/* -- TT: helmet -- */
[speedracers-helmet : helmet = "Speed Racer's helmet"
    @"urn:costume-item:helmet:speedracers-helmet"]

/* -- TT: high-top-boot -- */
[batmans-high-top-boot : high-top-boot = "Batman's high-top boot"
    @"urn:costume-item:high-top-boot:batmans-high-top-boot"]

/* -- TT: human -- */
[man : human = "man"
    @"urn:creature:human:man"]
[woman : human = "woman"
    @"urn:creature:human:woman"]

/* -- TT: image -- */
[image1 : image = "image1"
    @"urn:image:image1"]
[image10 : image = "image10"
```

```
    @"urn:image:image10"]
[image11 : image = "image11"
    @"urn:image:image11"]
[image12 : image = "image12"
    @"urn:image:image12"]
[image13 : image = "image13"
    @"urn:image:image13"]
[image14 : image = "image14"
    @"urn:image:image14"]
[image15 : image = "image15"
    @"urn:image:image15"]
[image16 : image = "image16"
    @"urn:image:image16"]
[image17 : image = "image17"
    @"urn:image:image17"]
[image18 : image = "image18"
    @"urn:image:image18"]
[image19 : image = "image19"
    @"urn:image:image19"]
[image2 : image = "image2"
    @"urn:image:image2"]
[image20 : image = "image20"
    @"urn:image:image20"]
[image3 : image = "image3"
    @"urn:image:image3"]
[image4 : image = "image4"
    @"urn:image:image4"]
[image5 : image = "image5"
    @"urn:image:image5"]
[image6 : image = "image6"
    @"urn:image:image6"]
[image7 : image = "image7"
    @"urn:image:image7"]
[image8 : image = "image8"
    @"urn:image:image8"]
[image9 : image = "image9"
    @"urn:image:image9"]
```

```
/* -- TT: mask -- */
[batmans-mask : mask = "Batman's mask"
    @"urn:costume-item:mask:batmans-mask"]

/* -- TT: necktie -- */
[freds-necktie : necktie = "Fred's necktie"
    @"urn:costume-item:necktie:freds-necktie"]
[spongebobs-necktie : necktie = "Spongebob's necktie"
    @"urn:costume-item:necktie:spongebobs-necktie"]

/* -- TT: pants -- */
[barts-pants : pants = "Bart's pants"
    @"urn:costume-item:pants:barts-pants"]
[batmans-pants : pants = "Batman's pants"
    @"urn:costume-item:pants:batmans-pants"]
[charlies-pants : pants = "Charlie's pants"
    @"urn:costume-item:pants:charlies-pants"]
[erics-pants : pants = "Eric's pants"
    @"urn:costume-item:pants:erics-pants"]
[georges-pants : pants = "George's pants"
    @"urn:costume-item:pants:georges-pants"]
[homers-pants : pants = "Homer's pants"
    @"urn:costume-item:pants:homers-pants"]
[mickeys-pants : pants = "Mickey's pants"
    @"urn:costume-item:pants:mickeys-pants"]
[mightymouses-pants : pants = "Mighty Mouse's pants"
    @"urn:costume-item:pants:mightymouses-pants"]
[popeyes-pants : pants = "Popeye's pants"
    @"urn:costume-item:pants:popeyes-pants"]
[speedracers-pants : pants = "Speed Racer's pants"
    @"urn:costume-item:pants:speedracers-pants"]
[spongebobs-pants : pants = "Spongebob's pants"
    @"urn:costume-item:pants:spongebobs-pants"]
[supermans-pants : pants = "Superman's pants"
    @"urn:costume-item:pants:supermans-pants"]

/* -- TT: pipe -- */
[popeyes-pipe : pipe = "Popeye's pipe"
```

```
    @"urn:costume-item:pipe:popeyes-pipe"]

/* -- TT: shirt -- */
[barts-shirt : shirt = "Bart's shirt"
    @"urn:costume-item:shirt:barts-shirt"]
[charlies-shirt : shirt = "Charlie's shirt"
    @"urn:costume-item:shirt:charlies-shirt"]
[donalds-shirt : shirt = "Donald's shirt"
    @"urn:costume-item:shirt:donalds-shirt"]
[erics-shirt : shirt = "Eric's shirt"
    @"urn:costume-item:shirt:erics-shirt"]
[freds-shirt : shirt = "Fred's shirt"
    @"urn:costume-item:shirt:freds-shirt"]
[georges-shirt : shirt = "George's shirt"
    @"urn:costume-item:shirt:georges-shirt"]
[homers-shirt : shirt = "Homer's shirt"
    @"urn:costume-item:shirt:homers-shirt"]
[poohs-shirt : shirt = "Pooh's shirt"
    @"urn:costume-item:shirt:poohs-shirt"]
[popeyes-shirt : shirt = "Popeye's shirt"
    @"urn:costume-item:shirt:popeyes-shirt"]
[speedracers-shirt : shirt = "Speed Racer's shirt"
    @"urn:costume-item:shirt:speedracers-shirt"]
[spongebobs-shirt : shirt = "Spongebob's shirt"
    @"urn:costume-item:shirt:spongebobs-shirt"]
[thegrinchs-shirt : shirt = "The Grinch's shirt"
    @"urn:costume-item:shirt:thegrinchs-shirt"]

/* -- TT: shoes -- */
[barts-shoes : shoes = "Bart's shoes"
    @"urn:costume-item:shoes:barts-shoes"]
[charlies-shoes : shoes = "Charlie's shoes"
    @"urn:costume-item:shoes:charlies-shoes"]
[erics-shoes : shoes = "Eric's shoes"
    @"urn:costume-item:shoes:erics-shoes"]
[georges-shoes : shoes = "George's shoes"
    @"urn:costume-item:shoes:georges-shoes"]
[homers-shoes : shoes = "Homer's shoes"
```

```
    @"urn:costume-item:shoes:homers-shoes"]
[mightymouses-shoes : shoes = "Mighty Mouse's shoes"
    @"urn:costume-item:shoes:mightymouses-shoes"]
[popeyes-shoes : shoes = "Popeye's shoes"
    @"urn:costume-item:shoes:popeyes-shoes"]
[speedracers-shoes : shoes = "Speed Racer's shoes"
    @"urn:costume-item:shoes:speedracers-shoes"]
[spongebobs-shoes : shoes = "Spongebob's shoes"
    @"urn:costume-item:shoes:spongebobs-shoes"]
[thegrinchs-shoes : shoes = "The Grinch's shoes"
    @"urn:costume-item:shoes:thegrinchs-shoes"]

/* -- TT: socks -- */
[charlies-socks : socks = "Charlie's socks"
    @"urn:costume-item:socks:charlies-socks"]
[spongebobs-socks : socks = "Spongebob's socks"
    @"urn:costume-item:socks:spongebobs-socks"]

/* -- TT: suit -- */
[porkys-suit : suit = "Porky's suit"
    @"urn:costume-item:suit:porkys-suit"]

/* -- TT: vest -- */
[topcats-vest : vest = "Top Cat's vest"
    @"urn:costume-item:vest:topcats-vest"]

/* -- TT: (untyped) -- */
[has-geometric-shape = "has geometric shape"
    @"urn:assoc-type:has-geometric-shape"]
[id1]
[id10]
[id100]
[id101]
[id102]
[id103]
[id104]
[id105]
[id106]
```

```
[id107]
[id108]
[id109]
[id11]
[id110]
[id111]
[id112 = "wear"]
[id113]
[id114]
[id115]
[id116]
[id117]
[id118]
[id119 = "Version"
    @"http://psi.ontopia.net/ontology/version"
    @"http://purl.org/dc/elements/1.1/version"]
[id12]
[id120]
[id121]
[id122]
[id123]
[id124]
[id125]
[id126]
[id127]
[id128]
[id129]
[id13]
[id130]
[id131]
[id132]
[id133 = "colored with"]
[id134]
[id135]
[id136]
[id137 = "is a"]
[id138 = "has skin color of"]
[id139]
```

```
[id14]
[id140]
[id141]
[id15]
[id16]
[id17]
[id18]
[id19]
[id2]
[id20]
[id21]
[id22]
[id23]
[id24]
[id25]
[id26 = "presents"]
[id27]
[id28]
[id29]
[id3 = "presents in"]
[id30]
[id31]
[id32]
[id33]
[id34]
[id35]
[id36]
[id37]
[id38]
[id39]
[id4]
[id40]
[id41]
[id42]
[id43]
[id44]
[id45]
[id46]
```

```
[id47 = "used in"]
[id48]
[id49]
[id5]
[id50]
[id51]
[id52]
[id53 = "Name"
    @"http://psi.topicmaps.org/iso13250/model/topic-name"]
[id54]
[id55 = "is type of"]
[id56]
[id57]
[id58]
[id59]
[id6]
[id60 = "used as skin color of"]
[id61]
[id62]
[id63]
[id65]
[id66]
[id67]
[id68]
[id69]
[id7]
[id70 = "has character shape of"]
[id71]
[id72]
[id73]
[id74]
[id75]
[id76 = "build up character of"]
[id77]
[id78]
[id79]
[id8]
[id80]
```

```
[id81]
[id82 = "Creator"
    @"http://purl.org/dc/elements/1.1/creator"
    @"http://psi.ontopia.net/ontology/creator"]
[id83 = "Description"
    @"http://purl.org/dc/elements/1.1/description"
    @"http://psi.ontopia.net/ontology/description"]
[id84]
[id85]
[id86]
[id87]
[id88]
[id89]
[id9]
[id90]
[id91]
[id92 = "scarf"
    @"urn:costume-item:scarf"]
[id93]
[id94]
[id95]
[id96]
[id97]
[id98]
[id99 = "worn by"]

/* ----------------- Associations -------------- */

/* -- AT: has-costumecolor */
has-costumecolor( black : color, batmans-body-suit : costume-item )
has-costumecolor( black : color, batmans-cloak : costume-item )
has-costumecolor( black : color, batmans-high-top-boot : costume-
item )
has-costumecolor( black : color, batmans-mask : costume-item )
has-costumecolor( black : color, batmans-pants : costume-item )
has-costumecolor( black : color, charlies-pants : costume-item )
has-costumecolor( black : color, erics-shoes : costume-item )
has-costumecolor( black : color, homers-shoes : costume-item )
```

```
has-costumecolor( black : color, popeyes-shirt : costume-item )
has-costumecolor( black : color, spongebobs-shoes : costume-item )
has-costumecolor( blue : color, barts-pants : costume-item )
has-costumecolor( blue : color, barts-shoes : costume-item )
has-costumecolor( blue : color, donalds-hat : costume-item )
has-costumecolor( blue : color, donalds-shirt : costume-item )
has-costumecolor( blue : color, erics-hat : costume-item )
has-costumecolor( blue : color, freds-necktie : costume-item )
has-costumecolor( blue : color, georges-pants : costume-item )
has-costumecolor( blue : color, homers-pants : costume-item )
has-costumecolor( blue : color, porkys-suit : costume-item )
has-costumecolor( blue : color, rockys-glasses : costume-item )
has-costumecolor( blue : color, scoobys-collar : costume-item )
has-costumecolor( blue : color, speedracers-shirt : costume-item )
has-costumecolor( blue : color, spidermans-body-suit : costume-item
)
has-costumecolor( blue : color, supermans-body-suit : costume-item
)
has-costumecolor( blue : color, underdogs-cloak : costume-item )
has-costumecolor( brown : color, charlies-shoes : costume-item )
has-costumecolor( brown : color, erics-pants : costume-item )
has-costumecolor( brown : color, popeyes-shoes : costume-item )
has-costumecolor( brown : color, speedracers-shoes : costume-item )
has-costumecolor( brown : color, spongebobs-pants : costume-item )
has-costumecolor( dark-blue : color, popeyes-pants : costume-item )
has-costumecolor( green : color, georges-belt : costume-item )
has-costumecolor( orange : color, freds-shirt : costume-item )
has-costumecolor( orange : color, popeyes-pipe : costume-item )
has-costumecolor( orange : color, speedracers-gloves : costume-item
)
has-costumecolor( red : color, barts-shirt : costume-item )
has-costumecolor( red : color, donalds-shirt : costume-item )
has-costumecolor( red : color, erics-shirt : costume-item )
has-costumecolor( red : color, mickeys-pants : costume-item )
has-costumecolor( red : color, mightymouses-cloak : costume-item )
has-costumecolor( red : color, mightymouses-pants : costume-item )
has-costumecolor( red : color, mightymouses-shoes : costume-item )
has-costumecolor( red : color, poohs-shirt : costume-item )
```

```
has-costumecolor( red : color, porkys-bow-tie : costume-item )
has-costumecolor( red : color, spongebobs-necktie : costume-item )
has-costumecolor( red : color, supermans-cloak : costume-item )
has-costumecolor( red : color, supermans-pants : costume-item )
has-costumecolor( red : color, thegrinchs-hat : costume-item )
has-costumecolor( red : color, thegrinchs-shirt : costume-item )
has-costumecolor( red : color, thegrinchs-shoes : costume-item )
has-costumecolor( red : color, underdogs-body-suit : costume-item )
has-costumecolor( violet : color, topcats-hat : costume-item )
has-costumecolor( violet : color, topcats-vest : costume-item )
has-costumecolor( white : color, bullwinkles-gloves : costume-item
)
has-costumecolor( white : color, georges-shirt : costume-item )
has-costumecolor( white : color, georges-shoes : costume-item )
has-costumecolor( white : color, mickeys-gloves : costume-item )
has-costumecolor( white : color, mightymouses-belt : costume-item )
has-costumecolor( white : color, mightymouses-gloves : costume-item
)
has-costumecolor( white : color, porkys-gloves : costume-item )
has-costumecolor( white : color, speedracers-helmet : costume-item
)
has-costumecolor( white : color, speedracers-pants : costume-item )
has-costumecolor( white : color, spongebobs-shirt : costume-item )
has-costumecolor( white : color, spongebobs-socks : costume-item )
has-costumecolor( white : color, thegrinchs-hat : costume-item )
has-costumecolor( white : color, thegrinchs-shirt : costume-item )
has-costumecolor( white : color, thegrinchs-shoes : costume-item )
has-costumecolor( yellow : color, batmans-belt : costume-item )
has-costumecolor( yellow : color, charlies-shirt : costume-item )
has-costumecolor( yellow : color, charlies-socks : costume-item )
has-costumecolor( yellow : color, donalds-shirt : costume-item )
has-costumecolor( yellow : color, mightymouses-body-suit : costume-
item )
has-costumecolor( yellow : color, supermans-belt : costume-item )

/* -- AT: has-creature-type */
has-creature-type( bear : creature, winnie-the-pooh : character-
name )
```

```
has-creature-type( bird : creature, tweety : character-name )
has-creature-type( cat : creature, felix-the-cat : character-name )
has-creature-type( cat : creature, sylvester : character-name )
has-creature-type( cat : creature, tom : character-name )
has-creature-type( cat : creature, top-cat : character-name )
has-creature-type( dog : creature, scooby-doo : character-name )
has-creature-type( dog : creature, underdog : character-name )
has-creature-type( duck : creature, daffy-duck : character-name )
has-creature-type( duck : creature, donald-duck : character-name )
has-creature-type( man : creature, bart-simpson : character-name )
has-creature-type( man : creature, batman : character-name )
has-creature-type( man : creature, charlie-brown : character-name )
has-creature-type( man : creature, eric-cartman : character-name )
has-creature-type( man : creature, fred-flintstone : character-name
)
has-creature-type( man : creature, george-jetson : character-name )
has-creature-type( man : creature, homer-simpson : character-name )
has-creature-type( man : creature, popeye : character-name )
has-creature-type( man : creature, speed-racer : character-name )
has-creature-type( man : creature, spider-man : character-name )
has-creature-type( man : creature, superman : character-name )
has-creature-type( moose : creature, bullwinkle : character-name )
has-creature-type( mouse : creature, jerry : character-name )
has-creature-type( mouse : creature, mickey-mouse : character-name
)
has-creature-type( mouse : creature, mighty-mouse : character-name
)
has-creature-type( panther : creature, pink-panther : character-
name )
has-creature-type( pig : creature, porky-pig : character-name )
has-creature-type( rabbit : creature, bugs-bunny : character-name )
has-creature-type( sponge : creature, spongebob : character-name )
has-creature-type( squirrel : creature, rocky : character-name )
has-creature-type( woman : creature, the-powerpuff-girls :
character-name )

/* -- AT: has-skincolor */
has-skincolor( black : color, daffy-duck : character-name )
```

```
has-skincolor( black : color, felix-the-cat : character-name )
has-skincolor( black : color, mickey-mouse : character-name )
has-skincolor( black : color, pink-panther : character-name )
has-skincolor( black : color, scooby-doo : character-name )
has-skincolor( black : color, sylvester : character-name )
has-skincolor( brown : color, bullwinkle : character-name )
has-skincolor( brown : color, mighty-mouse : character-name )
has-skincolor( brown : color, scooby-doo : character-name )
has-skincolor( green : color, the-grinch : character-name )
has-skincolor( grey : color, bugs-bunny : character-name )
has-skincolor( grey : color, rocky : character-name )
has-skincolor( grey : color, tom : character-name )
has-skincolor( light-skin-color : color, charlie-brown : character-
name )
has-skincolor( light-skin-color : color, eric-cartman : character-
name )
has-skincolor( light-skin-color : color, fred-flintstone :
character-name )
has-skincolor( light-skin-color : color, jerry : character-name )
has-skincolor( light-skin-color : color, mickey-mouse : character-
name )
has-skincolor( light-skin-color : color, mighty-mouse : character-
name )
has-skincolor( light-skin-color : color, popeye : character-name )
has-skincolor( light-skin-color : color, speed-racer : character-
name )
has-skincolor( light-skin-color : color, superman : character-name
)
has-skincolor( light-skin-color : color, underdog : character-name
)
has-skincolor( orange : color, bullwinkle : character-name )
has-skincolor( orange : color, daffy-duck : character-name )
has-skincolor( orange : color, jerry : character-name )
has-skincolor( orange : color, tweety : character-name )
has-skincolor( pink : color, pink-panther : character-name )
has-skincolor( pink : color, porky-pig : character-name )
has-skincolor( red : color, sylvester : character-name )
has-skincolor( white : color, bugs-bunny : character-name )
```

```
has-skincolor( white : color, donald-duck : character-name )
has-skincolor( white : color, felix-the-cat : character-name )
has-skincolor( white : color, sylvester : character-name )
has-skincolor( white : color, tom : character-name )
has-skincolor( yellow : color, homer-simpson : character-name )
has-skincolor( yellow : color, pink-panther : character-name )
has-skincolor( yellow : color, spongebob : character-name )
has-skincolor( yellow : color, top-cat : character-name )
has-skincolor( yellow : color, tweety : character-name )
has-skincolor( yellow : color, winnie-the-pooh : character-name )

/* -- AT: presents */
presents( bart-simpson : character-name, image1 : image )
presents( bart-simpson : character-name, image2 : image )
presents( bart-simpson : character-name, image3 : image )
presents( bart-simpson : character-name, image4 : image )
presents( bart-simpson : character-name, image5 : image )
presents( bugs-bunny : character-name, image10 : image )
presents( bugs-bunny : character-name, image6 : image )
presents( bugs-bunny : character-name, image7 : image )
presents( bugs-bunny : character-name, image8 : image )
presents( bugs-bunny : character-name, image9 : image )
presents( charlie-brown : character-name, image11 : image )
presents( charlie-brown : character-name, image12 : image )
presents( charlie-brown : character-name, image13 : image )
presents( charlie-brown : character-name, image14 : image )
presents( charlie-brown : character-name, image15 : image )
presents( daffy-duck : character-name, image16 : image )
presents( daffy-duck : character-name, image17 : image )
presents( daffy-duck : character-name, image18 : image )
presents( daffy-duck : character-name, image19 : image )
presents( daffy-duck : character-name, image20 : image )

/* -- AT: wears */
wears( bart-simpson : character-name, barts-pants : costume-item )
wears( bart-simpson : character-name, barts-shirt : costume-item )
wears( bart-simpson : character-name, barts-shoes : costume-item )
wears( batman : character-name, batmans-belt : costume-item )
```

```
wears( batman : character-name, batmans-body-suit : costume-item )
wears( batman : character-name, batmans-cloak : costume-item )
wears( batman : character-name, batmans-high-top-boot : costume-
item )
wears( batman : character-name, batmans-mask : costume-item )
wears( batman : character-name, batmans-pants : costume-item )
wears( bullwinkle : character-name, bullwinkles-gloves : costume-
item )
wears( charlie-brown : character-name, charlies-pants : costume-
item )
wears( charlie-brown : character-name, charlies-shirt : costume-
item )
wears( charlie-brown : character-name, charlies-shoes : costume-
item )
wears( charlie-brown : character-name, charlies-socks : costume-
item )
wears( donald-duck : character-name, donalds-hat : costume-item )
wears( donald-duck : character-name, donalds-shirt : costume-item )
wears( eric-cartman : character-name, erics-hat : costume-item )
wears( eric-cartman : character-name, erics-pants : costume-item )
wears( eric-cartman : character-name, erics-shirt : costume-item )
wears( eric-cartman : character-name, erics-shoes : costume-item )
wears( fred-flintstone : character-name, freds-necktie : costume-
item )
wears( fred-flintstone : character-name, freds-shirt : costume-item
)
wears( george-jetson : character-name, georges-belt : costume-item
)
wears( george-jetson : character-name, georges-pants : costume-item
)
wears( george-jetson : character-name, georges-shirt : costume-item
)
wears( george-jetson : character-name, georges-shoes : costume-item
)
wears( homer-simpson : character-name, homers-pants : costume-item
)
wears( homer-simpson : character-name, homers-shirt : costume-item
)
```

```
wears( homer-simpson : character-name, homers-shoes : costume-item
)
wears( mickey-mouse : character-name, mickeys-gloves : costume-item
)
wears( mickey-mouse : character-name, mickeys-pants : costume-item
)
wears( mighty-mouse : character-name, mightymouses-belt : costume-
item )
wears( mighty-mouse : character-name, mightymouses-body-suit :
costume-item )
wears( mighty-mouse : character-name, mightymouses-cloak : costume-
item )
wears( mighty-mouse : character-name, mightymouses-gloves :
costume-item )
wears( mighty-mouse : character-name, mightymouses-pants : costume-
item )
wears( mighty-mouse : character-name, mightymouses-shoes : costume-
item )
wears( popeye : character-name, popeyes-pants : costume-item )
wears( popeye : character-name, popeyes-pipe : costume-item )
wears( popeye : character-name, popeyes-shirt : costume-item )
wears( popeye : character-name, popeyes-shoes : costume-item )
wears( porky-pig : character-name, porkys-bow-tie : costume-item )
wears( porky-pig : character-name, porkys-gloves : costume-item )
wears( porky-pig : character-name, porkys-suit : costume-item )
wears( rocky : character-name, rockys-glasses : costume-item )
wears( scooby-doo : character-name, scoobys-collar : costume-item )
wears( speed-racer : character-name, speedracers-gloves : costume-
item )
wears( speed-racer : character-name, speedracers-helmet : costume-
item )
wears( speed-racer : character-name, speedracers-pants : costume-
item )
wears( speed-racer : character-name, speedracers-shirt : costume-
item )
wears( speed-racer : character-name, speedracers-shoes : costume-
item )
```

```
wears( spider-man : character-name, spidermans-body-suit : costume-item )
wears( spongebob : character-name, spongebobs-necktie : costume-item )
wears( spongebob : character-name, spongebobs-pants : costume-item )
wears( spongebob : character-name, spongebobs-shirt : costume-item )
wears( spongebob : character-name, spongebobs-shoes : costume-item )
wears( spongebob : character-name, spongebobs-socks : costume-item )
wears( superman : character-name, supermans-belt : costume-item )
wears( superman : character-name, supermans-body-suit : costume-item )
wears( superman : character-name, supermans-cloak : costume-item )
wears( superman : character-name, supermans-pants : costume-item )
wears( the-grinch : character-name, thegrinchs-hat : costume-item )
wears( the-grinch : character-name, thegrinchs-shirt : costume-item )
wears( the-grinch : character-name, thegrinchs-shoes : costume-item )
wears( top-cat : character-name, topcats-hat : costume-item )
wears( top-cat : character-name, topcats-vest : costume-item )
wears( underdog : character-name, underdogs-body-suit : costume-item )
wears( underdog : character-name, underdogs-cloak : costume-item )
wears( winnie-the-pooh : character-name, poohs-shirt : costume-item )
```

3.3.4 Testing Phase

The test is conducted following comparison approach as illustrated in Figure 1.6 in order to observe result for both conventional CBIR and proposed Topic Maps-based method. Test scenario and result are explained in the next chapter to simulate various environment relating to the output results. These results are used for further assessment in evaluation phase.

3.3.5 Evaluation Phase

This phase evaluates the retrieved results acquired by the conventional CBIR and the proposed Topic Maps-based method in each test scenario in terms of precision rate.

To summarize, the experimental design is composed of five phases with some main detail as listed in table below.

Table 1.7 Summary of Experimental Design

Phase	Input	Output	Related Tools
Infrastructure Preparation Phase	• Infrastructure plan	System infrastructure divided into backend and frontend	• JRE6u27 • JRE7u60 • JRE8u71
Pre-processing Phase	• Training images	Images with theses following properties: • .jpg image renamed as consecutive numerical numbers starting from "0.jpg" • .jpg image resized to 384 x 256 • Solid white background	• Microsoft Picture Manager • Snagit12
Training Phase	• A folder of training images • Ontological terms of all training images	• .mat file containing extracted image low-level feature vectors • Topic Maps network of training images	• MATLAB and additional toolbox • Ontopoly
Testing Phase	• Query image	• Retrieved images	• MATLAB and additional toolbox • Omnigator,

Table 3.7 Summary of Experimental Design (Cont.)

Phase	Input	Output	Related Tools
Evaluation Phase	• Numbers of all retrieved results • Numbers of relevant retrieved results	• Precision rate (0-1)	• Equation of precision rate calculation

3.4 Chapter Summary

The detail of this chapter is separated to 3 parts including system design, comparison approach and experimental design. The system diagram part illustrates system related elements and process flows to provide overall concept of the system functions. Next, the comparison approach explain how this study is going to conduct an experiment to compare between conventional CBIR and the proposed Topic Maps-based system. The other part is experimental design which is separated into four consecutive phases aiming to provide steps of action to proceed through experiment. Result of the experiment is demonstrated in the next chapter to prove that the proposed Topic Maps-based environment is able to optimize retrieved results acquired from the conventional CBIR method.

Chapter 4

Result, Assessment and Remarkable Features

This chapter is mainly divided into three parts. The experiment result is first mentioned to report the test conducted following comparison approach as illustrated in previous chapter in order to compare result between conventional CBIR and the proposed Topic Maps-based method. Next, an assessment is calculated in order to prove that the proposed Topic Maps-based approach does increase accuracy and recall. Remarkable features are presented later to point out relevant techniques which are able to enhance retrieval performance and functionality by employing the proposed Topic Maps-based method.

4.1 Experiment Result

According to **Error! Reference source not found.** below, database used in the experiment consists of up to 480 images of 12 cartoon characters. Each of them comprises of 40 different gestures.

Table 1.1 Cartoon Characters Used for Testing

Character Name	Example
Bart Simpson	
Blossom	

Table 4.1 Cartoon Characters Used for Testing (Cont.)

Character Name	Example
Bubble	
Bugs Bunny	
Butter Cup	
Fred Flintstones	
Jerry	

Table 4.1 Cartoon Characters Used for Testing (Cont.)

Character Name	Example
Mickey Mouse	
Pink Panther	
Pluto	
Spider-Man	
Spongebob	

Regarding the experiment, CBIR will extract low-level feature vector of the query image then calculate similarity distance with existing images in database to find the best results with closest distance. The first retrieved image acquiring from CBIR will be selected as a key image to get the ontological term which is hereby an image name. This term is set as a beginning point to traverse along related terms and association in the Topic Maps network in order to get relevant result images. In other words, the Topic Maps-based approach does not require any calculation of similarity distance to get the results as CBIR does. The results of Topic Maps-based approach depends on association not calculation.

The result is observed at different points to compare between conventional CBIR and the proposed Topic Maps-based method following comparison approach as illustrated in previous chapter. The observation for both cases is done based on the same test scenario which is defined to cover all possible environments that may occur during the experiment.

There are 4 test scenarios for this experiment including:

- Scenario 1:
  Query image is exactly the same as one of images in database.
- Scenario 2:
  Query image is not exactly the same image but it is still one of character types as existing in database. Only some relevant results can be retrieved by CBIR in this case.
- Scenario 3:
  Query image is one of character types as existing in database. No relevant results can be retrieved by CBIR in this case.
- Scenario 4:
  Query image is not one of existing character type in database.

Therefore, the experiment and result can be explained into two main parts which are conventional CBIR result and Topic Maps-based result. The numbers of result image are limited to 20 images for demonstration and assessment purpose.

Test Scenario #1:

Query image

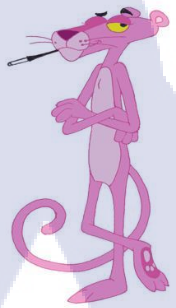

Figure 1.1 Query Image for Test Scenario #1

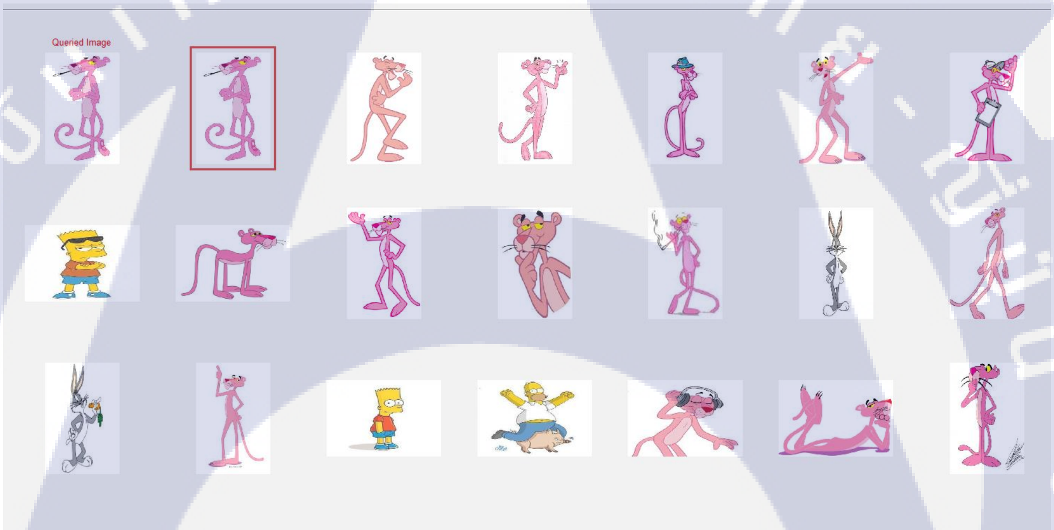

Figure 1.2 Test Scenario #1: Top 20 Images with Closest Distance

Table 1.2 Test Scenario #1: Similarity Distance of CBIR for Top 20 Images

Rank	Distance	Image Name
1	0	image399
2	9.4551	image303
3	9.8737	image401
4	9.8939	image369
5	11.0615	image320
6	11.7895	image381
7	12.078	image53
8	12.2688	image101
9	12.3894	image37
10	12.4129	image48
11	12.7577	image205
12	12.874	image152
13	12.9494	image363
14	13.2537	image113
15	13.3136	image464
16	13.6004	image58
17	13.7092	image341
18	14.0125	image0
19	14.0783	image56
20	14.2227	image63

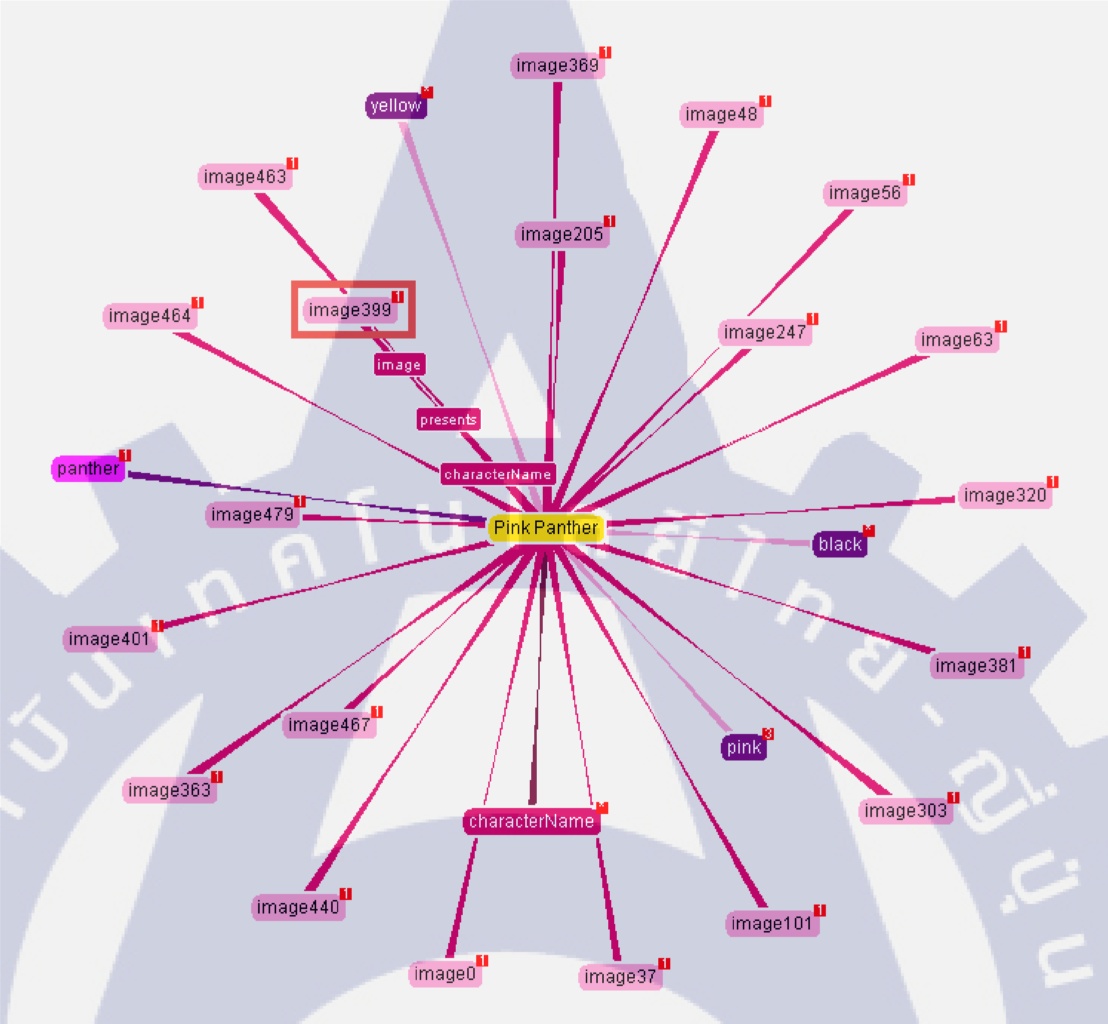

Figure 1.3 Test Scenario #1: Topic Maps Network and Related Images

Figure 1.4 Test Scenario #1: Result Images for Topic Maps-Based Method

For CBIR, in case that there are some images in database which are exactly the same as query image, the first result can be matched with zero distance meaning that both query image and the result are exactly the same in terms of low-level feature value as shown in Figure 1.2 and Table 4.2 Some other results are also matched correctly following distance order. This environment can be occurred when the query image is clear and its low-level feature vector can be apparently extracted without any noise or interrupted by background.

According to Table 1.2, the first retrieved result of CBIR which is “image399” is used as a key ontological term to set starting point in Topic Maps network as shown in Figure 1.3. The “image399” is an instance of “image” and connected to a character name “Pink Panther” with an association named “presents”. With this way, the semantic engine will traverse through all relationship defined with “presents” and retrieve images related to this association as shown in Figure 1.4.

Test Scenario #2:

Query image

Figure 1.5 Query Image for Test Scenario #2

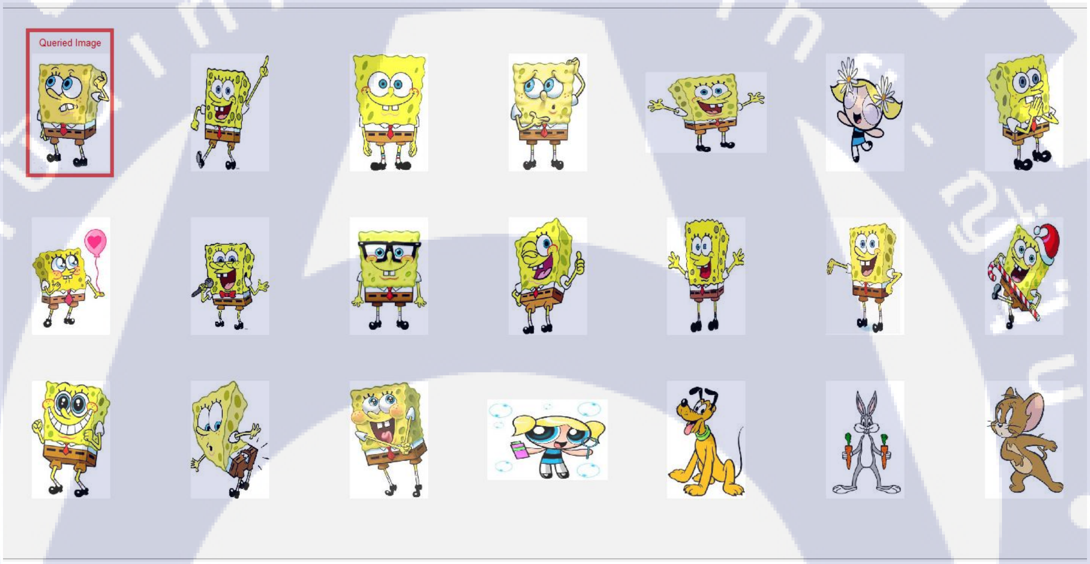

Figure 1.6 Test Scenario #2: Top 20 Images with Closest Distance

Table 1.3 Test Scenario #2: Similarity Distance of CBIR for Top 20 Images

Rank	Distance	Image Name
1	8.8015	image172
2	9.5881	image126
3	9.6395	image322
4	9.6406	image342
5	9.7836	image272
6	9.87	image313
7	10.1366	image283
8	10.2493	image290
9	10.4824	image111
10	10.7693	image224
11	10.9474	image390
12	11.1307	image382
13	11.4035	image367
14	11.6378	image308
15	11.6402	image149
16	11.6731	image206
17	11.684	image295
18	11.7346	image36
19	11.7921	image26
20	12.2046	image112

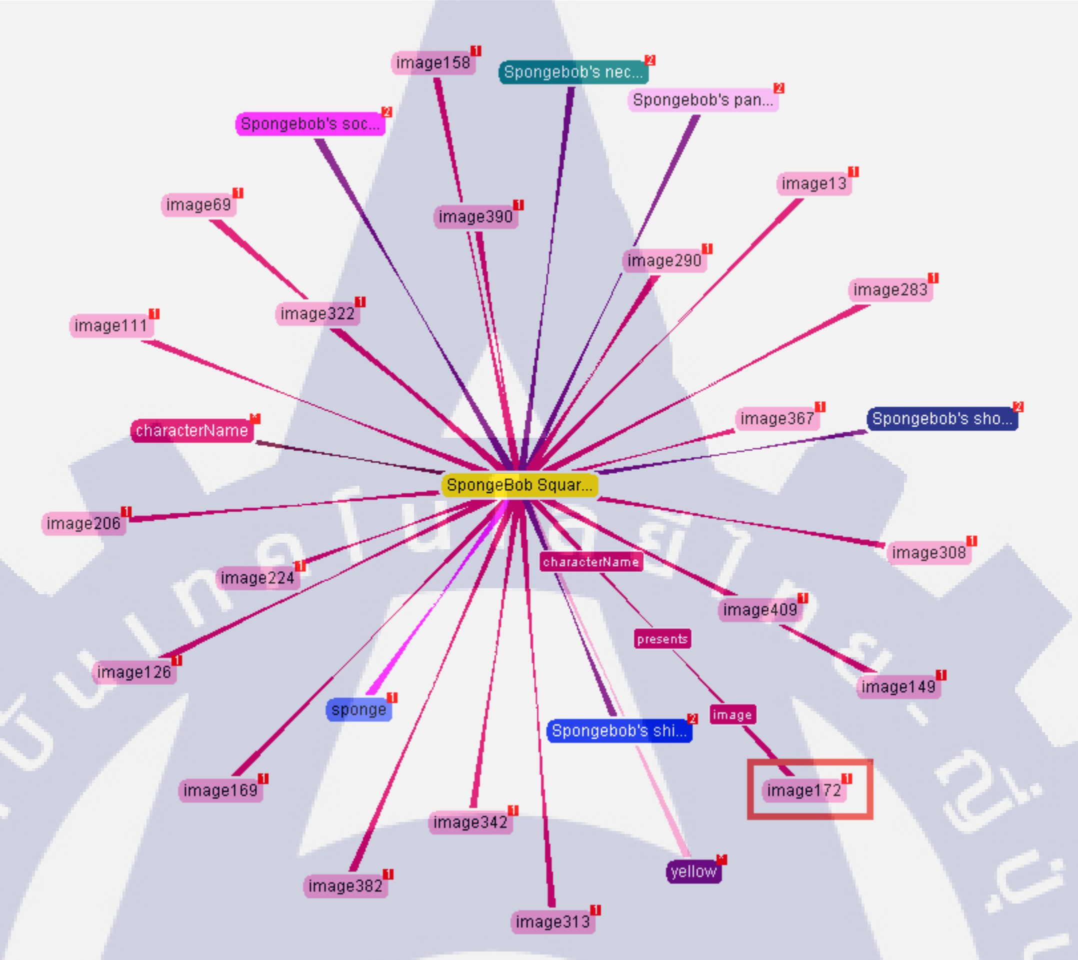

Figure 1.7 Test Scenario #2: Topic Maps Network and Related Images

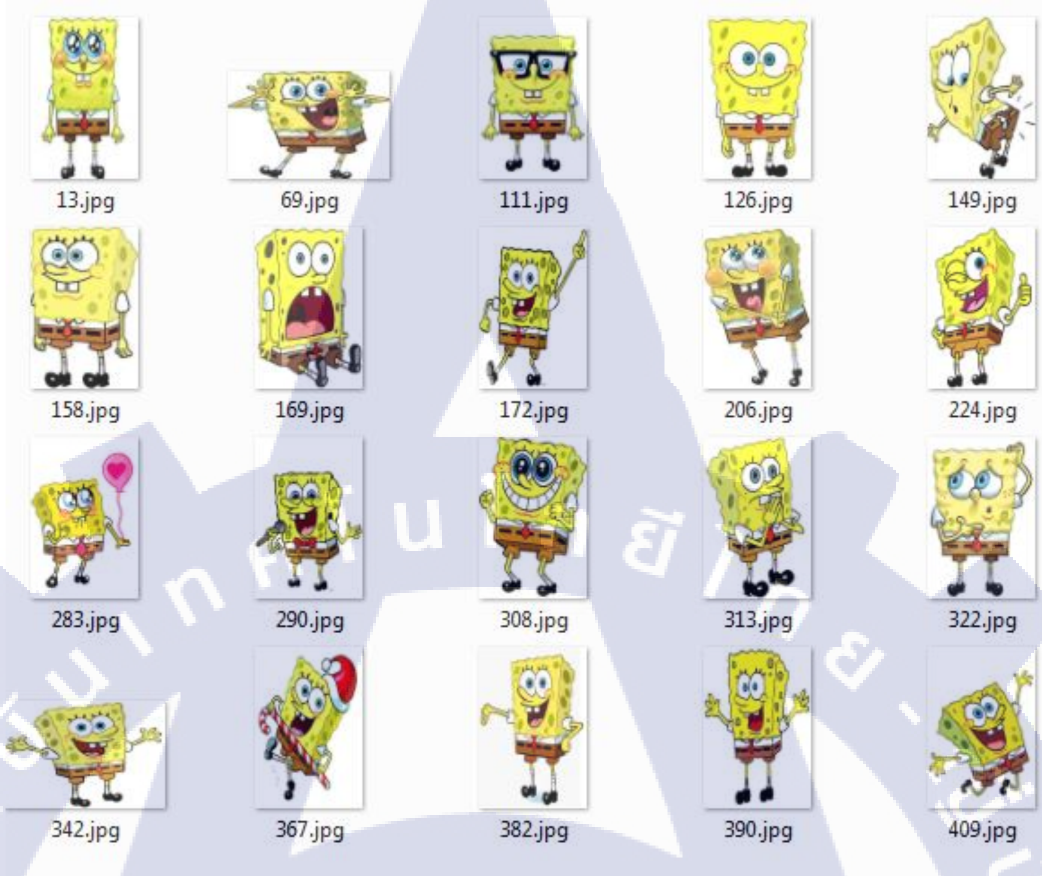

Figure 1.8 Test Scenario #2: Result Images for Topic Maps-Based Method

In case that the query image is not exactly the same image as existing in database but it is still the same character type, CBIR is then able to retrieve some relevant images as shown in Figure 1.6. This environment can be occurred if the query image is one of existing character type which is quite distinguished and stay out among other characters. This factor allows the query image to be correctly matched with other images of the same character in database more easily.

After that, semantic engine will acquire a key ontological term from the first retrieved image of CBIR which is hereby "image172" as defined in

Table 1.3. The “image172” is an instance of “image” and connected to a character name “Spongebob Squarepants” with an association named “presents” as shown in Figure 1.7. With this way, the semantic engine will traverse through all relationship defined with “presents” and retrieve images related to this association as displayed in Figure 1.8.

Test Scenario #3:

Query image

Figure 1.9 Query Image for Test Scenario #3

Figure 1.10 Test Scenario #3: Top 20 Images with Closest Distance

Table 1.4 Test Scenario #3: Similarity Distance of CBIR for Top 20 Images

Rank	Distance	Image Name
1	12.0682	image431
2	14.0434	image19
3	14.7067	image262
4	14.923	image274
5	14.9651	image345
6	14.9999	image85
7	15.3221	image305
8	15.3422	image28
9	15.8121	image25
10	15.8634	image103
11	15.9558	image191
12	15.9973	image286
13	16.033	image287
14	16.0811	image444
15	16.1546	image199
16	16.2408	image69
17	16.2515	image31
18	16.2971	image150
19	16.3114	image148
20	16.3233	image42

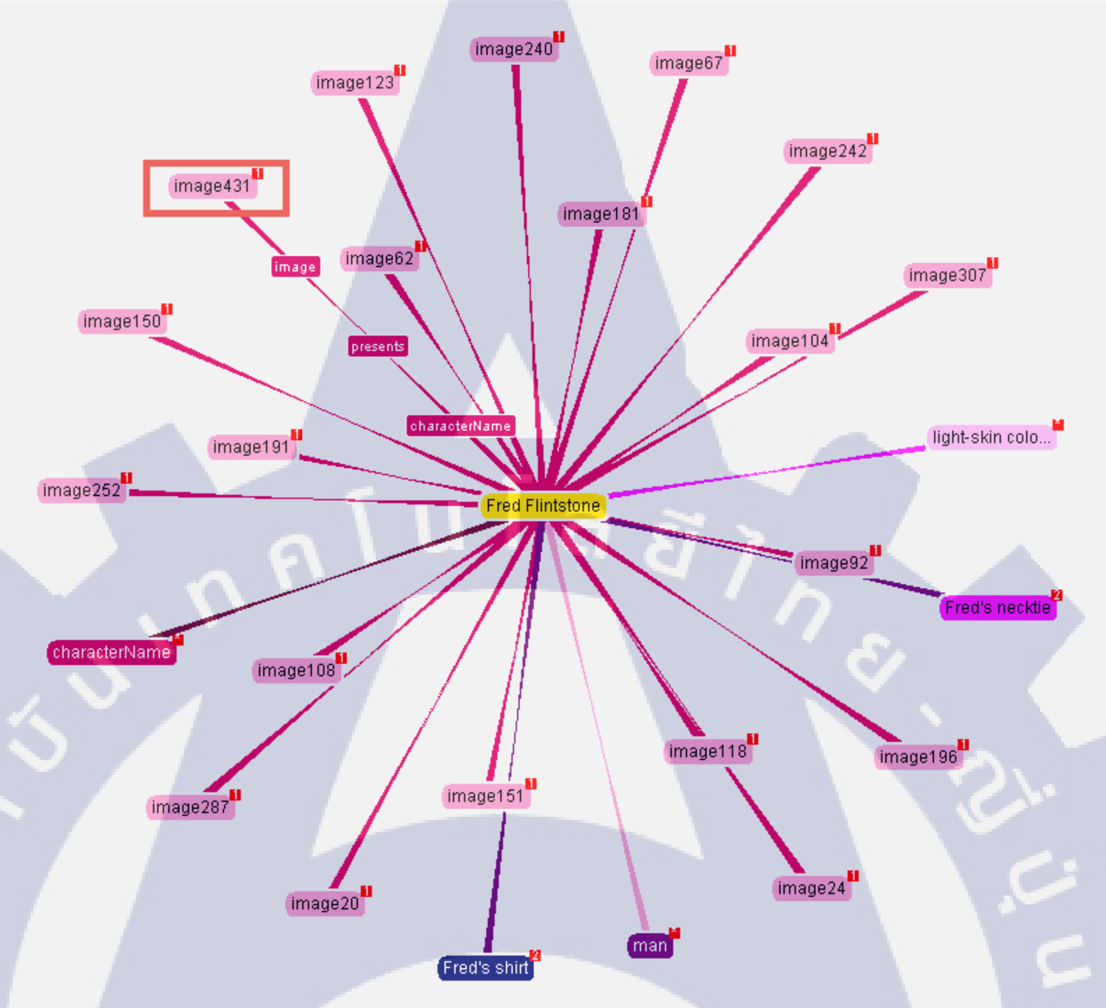

Figure 1.11 Test Scenario #3: Topic Maps Network and Related Images

Figure 1.12 Test Scenario #3: Result Images for Topic Maps-Based Method

In case that query image is the same character as existing in database but it is some kinds of enlargement at specific points or contains different color of background, CBIR is then unable to correctly match with relevant images in database as illustrated in Figure 1.10. This is because low-level feature extracted from the query image is interrupted by irrelevant elements resulting in mis-calculation of similarity distance.

However, semantic engine still works on its mechanism which is to acquire the first retrieved image of CBIR as a key image to get an ontological term. Thus in this case, the key ontological term is acquired from “image431” representing a character of “Fred Flintstones” with an association named “presents” as shown in Figure 1.11. With this way, the semantic engine will traverse through all relationship defined with “presents” and retrieve images related to this association as displayed in Figure 1.12 even though all of them are not the same character as the query image.

Test Scenario #4:

Query image

Figure 1.13 Query Image for Test Scenario #4

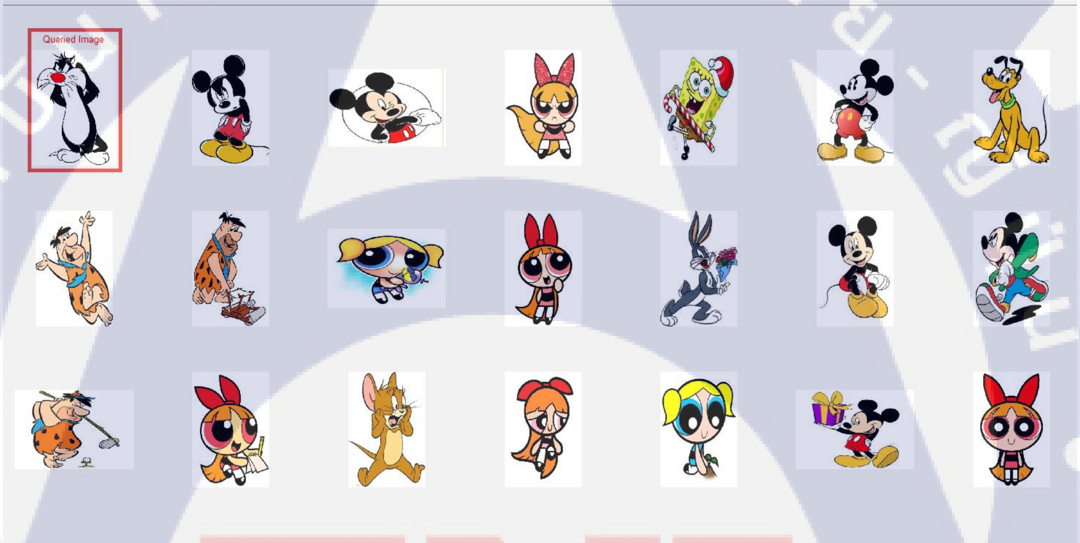

Figure 1.14 Test Scenario #4: Top 20 Images with Closest Distance

Table 1.5 Test Scenario #4: Similarity Distance of CBIR for Top 20 Images

Rank	Distance	Image Name
1	9.4852	image107
2	10.567	image351
3	10.9578	image301
4	10.9752	image367
5	10.9874	image114
6	11.2663	image36
7	11.5093	image282
8	11.5363	image14
9	11.784	image461
10	12.0265	image46
11	12.0845	image304
12	12.1724	image235
13	12.193	image244
14	12.222	image151
15	12.2791	image412
16	12.2956	image457
17	12.4074	image471
18	12.4653	image156
19	12.4943	image376
20	12.5229	image285

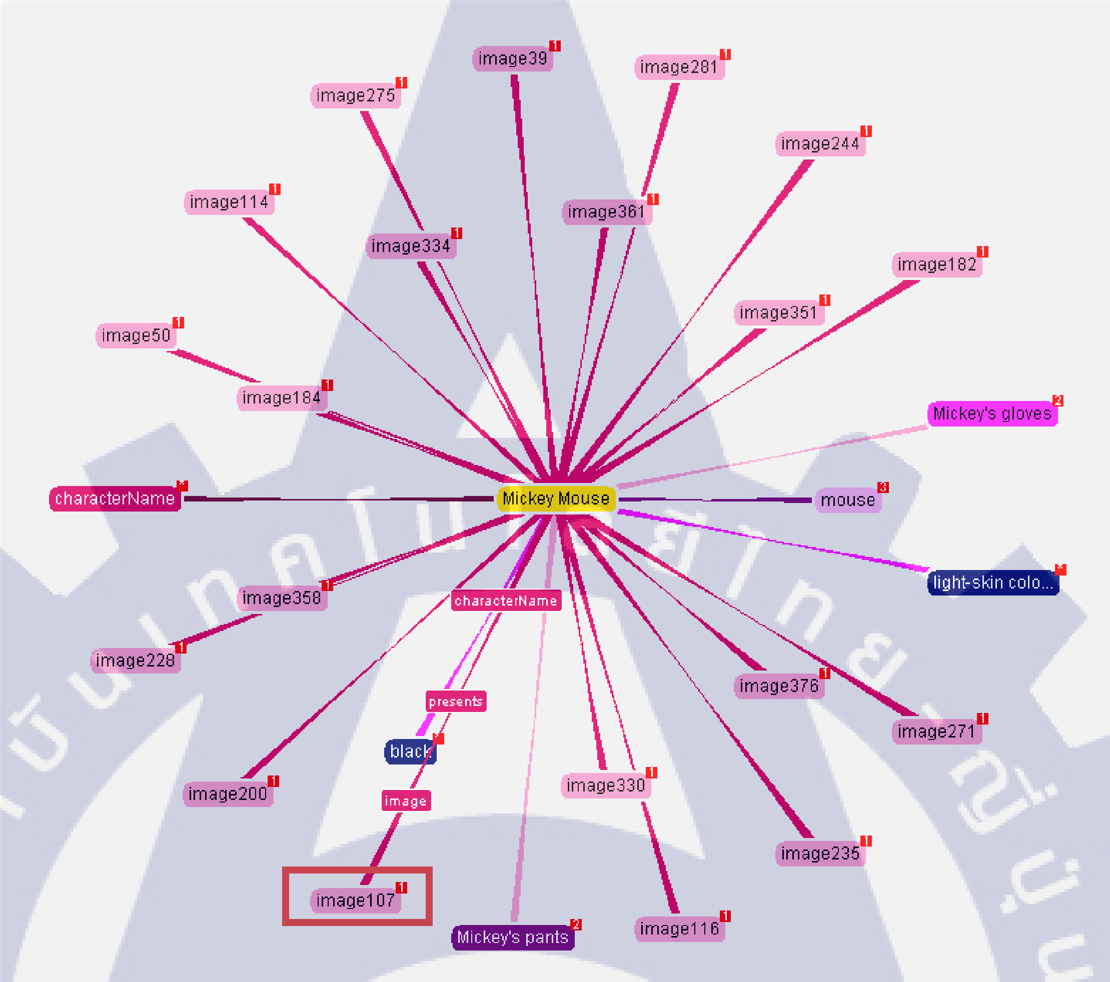

Figure 1.15 Test Scenario #4: Topic Maps Network and Related Images

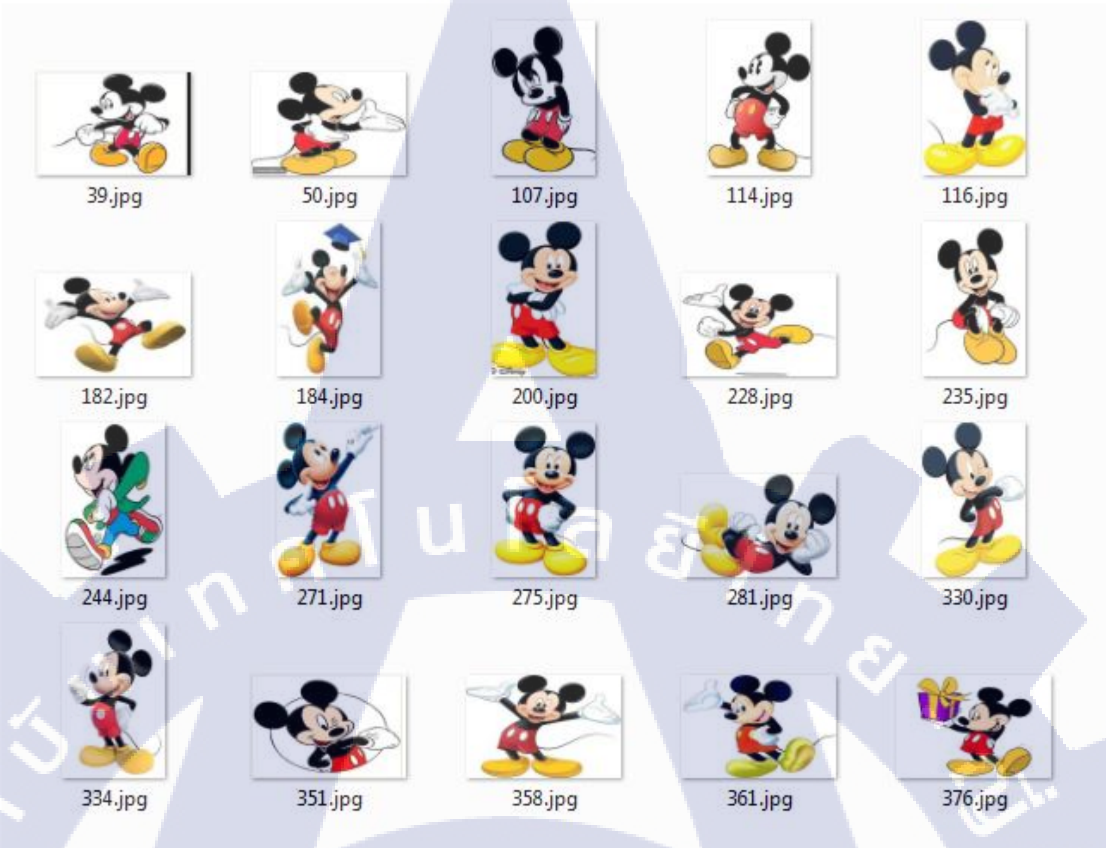

Figure 1.16 Test Scenario #4: Result Images for Topic Maps-Based Method

In case that query image is not the same character as existing in database, CBIR normally extract and calculate similarity distance between the query image and images in database as usual. However, no relevant results is matched as no such characters existing in database as shown in Figure 1.14.

Semantic engine also performs its normal function by acquiring a key ontological terms from the first retrieved image of CBIR wqhich is hereby “image107” representing a character of “Mickey Mouse” with an association named “presents” as shown in Figure 1.15. With this way, the semantic engine will traverse through all relationship defined with “presents” and retrieve images related to this association as displayed in Figure 1.16 even though all of them are not the same character as the query image.

To summarize all test scenarios and results, the Topic Maps-based results are always a collection of same character, which is the first character retrieved by conventional CBIR. This is because the system employs CBIR first retrieved result as selected image to acquire a key ontological term for further lookup in the Topic Maps network. Therefore, the system is able to find related images by using network of ontological association. On the other hand, in case that there is no correct match acquired by CBIR, the Topic Maps is then unable to retrieve any relevant results. However, the semantic based image retrieval engine allows human intervention to provide correct feedback to the query image and annotate it with appropriate ontological terms. With this way, the unknown query image will become already known image existing in database which can be used in next retrieval. This process is called "Annotation" which will be explained in the next parts.

4.2 Assessment of Precision and Recall

The assessment has been done based on the results of the test scenarios as mentioned earlier in order to concretely evaluate the results of conventional CBIR and the proposed Topic-Maps based approach. Regarding this matter, precision and recall or PR Test are calculated as below [60].

$$Precision = \frac{\text{Numbers of Relevant Images Retrieved}}{\text{Numbers of Retrieved Images}}$$

$$Recall = \frac{\text{Numbers of Relevant Images Retrieved}}{\text{Numbers of Relevant Images in Database}}$$

Figure 1.17 Calculation of Precision and Recall

Table 1.6 Comparison of Precision Rate

Case #	CBIR				Topic Maps-Based Method			
	Precision		Recall		Precision		Recall	
	No.s of Images	Rate	No.s of Images	Rate	No.s of Images	Rate	No.s of Images	Rate
1	15/20	0.75/1	15/40	0.38/1	20/20	1/1	20/40	0.5/1
2	15/20	0.75/1	15/40	0.38/1	20/20	1/1	20/40	0.5/1
3	1/20	0.05/1	1/40	0.025/1	0/20	0/1	0/40	0/1
4	0/20	0/1	0/40	0/1	0/20	0/1	0/40	0/1
Total	**31/80**	**1.55/4**	**31/160**	**0.79/4**	**40/80**	**2/4**	**40/160**	**1/4**

Referring to above table, the proposed Topic Maps-based approach is mostly able to enhance retrieval performance in terms of both precision and recall. The reason behind this is the Topic-Maps network is connected with ontological relationship allowing the same collection of targeted character type to be retrieved.

4.3 Remarkable Features of Topic Maps

4.3.1 Annotation

The proposed Topic Maps-based image retrieval is not implemented to replace the conventional CBIR, but to optimize or increase a chance that relevant images can be retrieved as much as possible. Referring to the results of some test scenarios, none of relevant image is retrieved by the Topic Maps-based method when the first retrieved result of CBIR is not correctly matched. Regarding this matter, the annotation feature is introduced as a part of semantic-based image retrieval mechanism in order to allow human intellectual to manipulate retrieval process in more semantic manner which encourages more accuracy.

Regarding the test scenarios conducted in previous part, the annotation can be performed to enhance precision especially the test scenario #3 and #4 as the first retrieved image of CBIR is not the correct one. The test scenario #1 is not necessary to do further annotation as the first retrieved image of CBIR is correct and query image is exactly the same image as existing in database. However, it is good practice to do annotation for the test scenario #2 as although the first retrieved result of CBIR is correct, query image is still not exactly the same image as existing in database. Thus performing annotation in this case can expand the image database as well as the Topic Maps ontological network.

Next in this part, the study provides demonstration of annotation only for the test scenario #3 and 4 by using Ontoply, a module of Ontopia Knowledge Suite, which is used for creating and modifying ontological term and structural network. The results are then re-processed for both CBIR and the Topic maps-based approach. However, there is no assessment conducted for the annotation as the objective of this demonstration is to point out additional concept that can enhance retrieval accuracy by accumulating human intellectual and growing database in more semantic manner.

For the test scenario #3, the first retrieved image of CBIR is actually not the same character type as query image as mentioned earlier. Regarding this matter, annotation process is conducted in order to save the query image into database and allow recognition when it is queried again next time. Besides, the query image is annotated with an ontological term using Ontopoly to expand Topic Maps network as shown in Figure 1.18. Thus, CBIR is able to find “image480” as the first correct retrieved image with zero distance after performing annotation as shown in Figure 1.19 and Table 1.7. Semantic engine then employs the “image480” as a key ontological term to start traversing along Topic Maps network and acquire a set of relevant images as illustrated in Figure 1.20 and Figure 1.21.

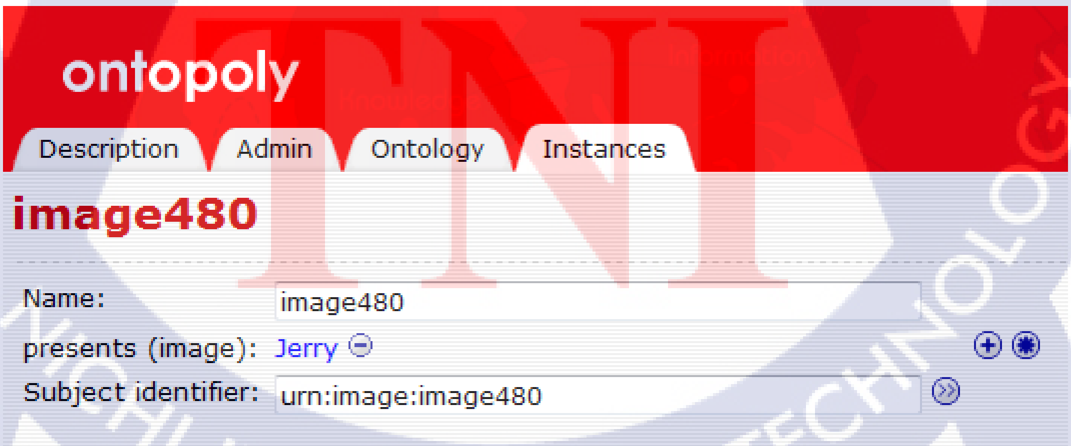

Figure 1.18 Test Scenario #3: Image Annotation in Ontopoly

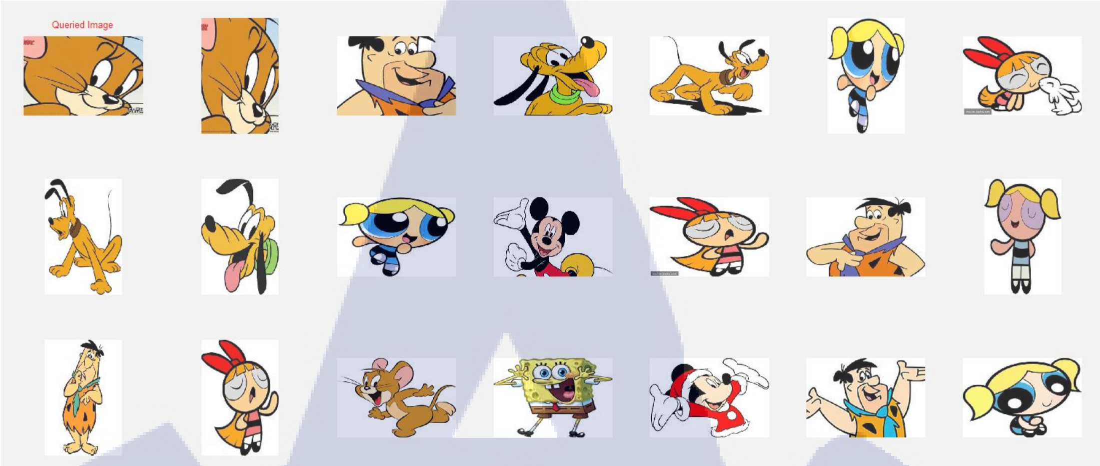

Figure 1.19 Test Scenario #3: New Results after Conducting Annotation

Table 1.7 Test Scenario #3: Similarity Distance after Conducting Annotation

Rank	Distance	Image Name
1	0	image480
2	12.0682	image431
3	14.0434	image19
4	14.7067	image262
5	14.923	image274
6	14.9651	image345
7	14.9999	image85
8	15.3221	image305
9	15.3422	image28
10	15.8121	image25
11	15.8634	image103
12	15.9558	image191
13	15.9973	image286
14	16.033	image287
15	16.0811	image444
16	16.1546	image199
17	16.2408	image69

Table 4.7 Test Scenario #3: Similarity Distance after Conducting Annotation (Cont)

Rank	Distance	Image Name
18	16.2515	image31
19	16.2971	image150
20	16.3114	image148

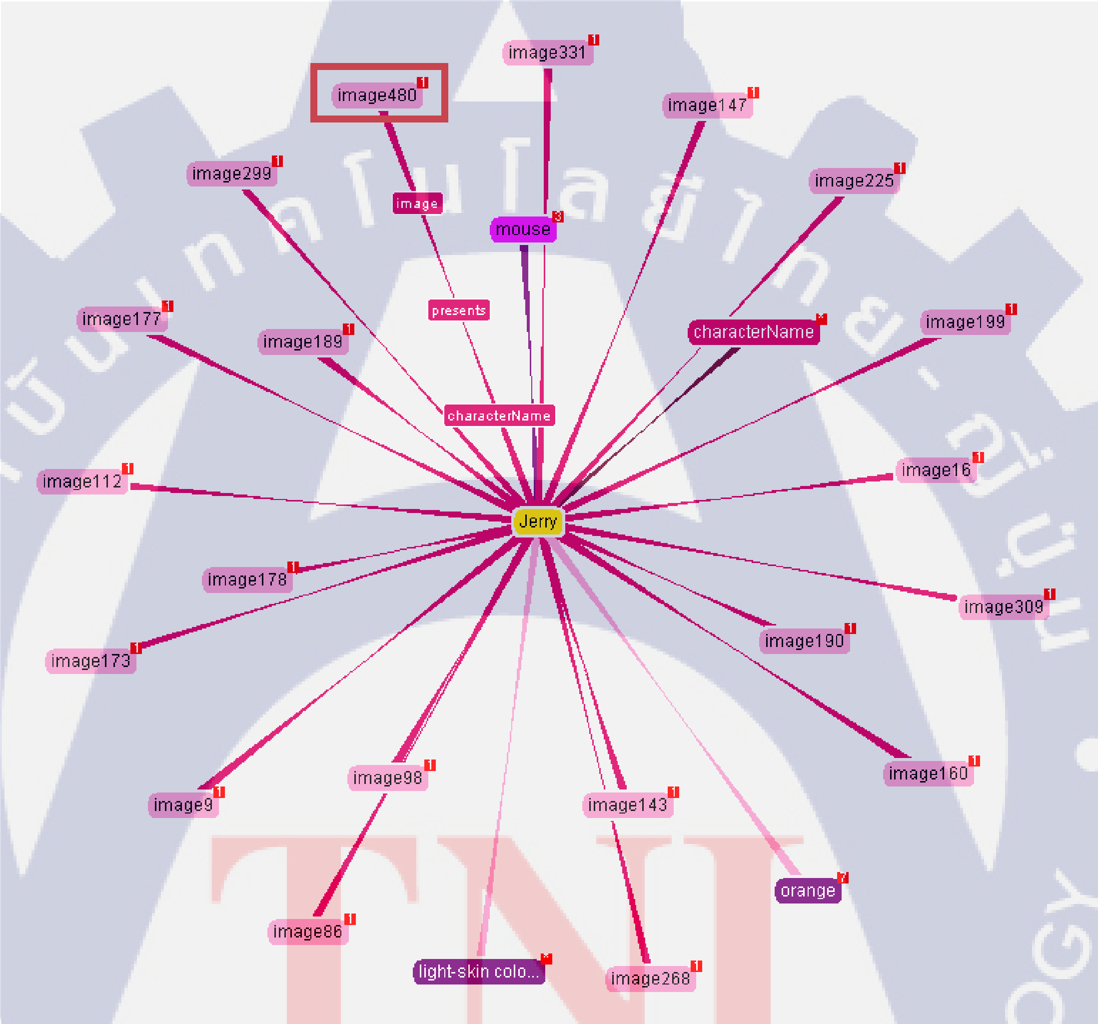

Figure 1.20 Jerry: Topic Maps Network and Related Images

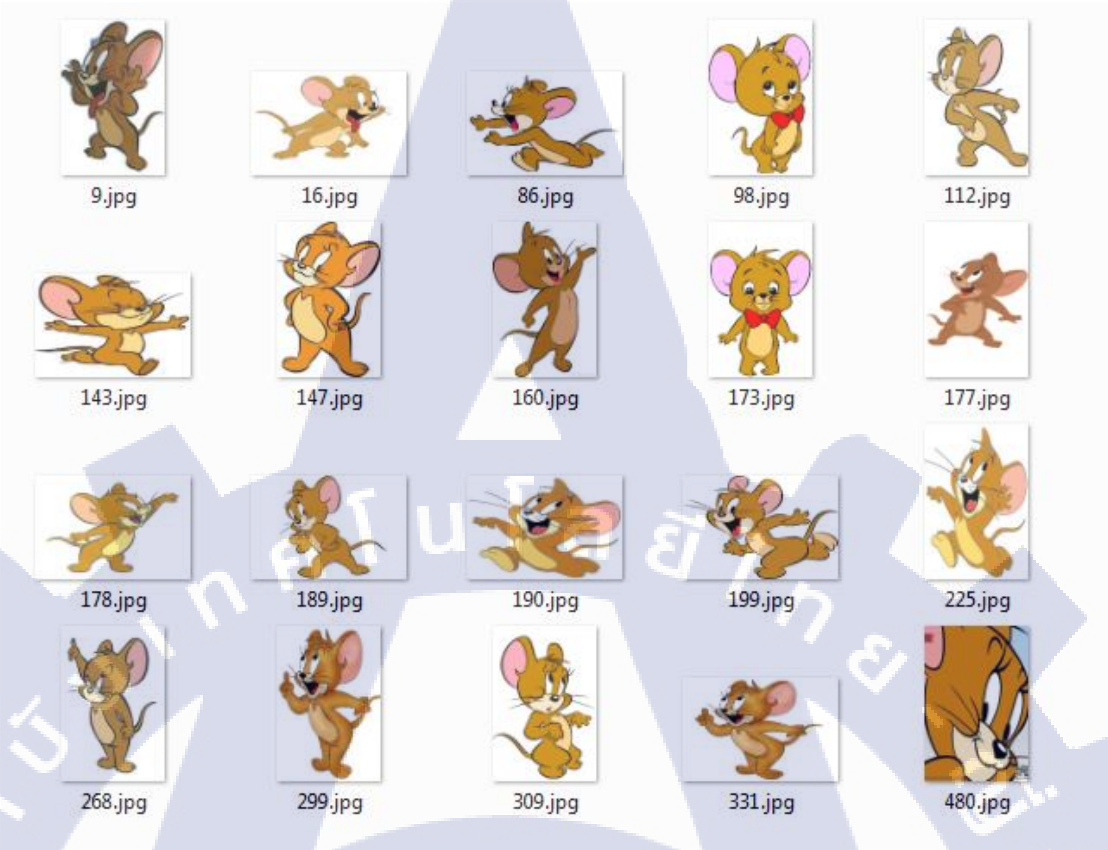

Figure 1.21 Jerry Related Images

On the other hand, the query image for the test scenario #4 is not one of existing character types available in database so the first retrieved image of CBIR is not correct as mentioned in Figure 1.14. To do annotation, the query image is saved into image database and an ontological term is added to the Topic Maps network as shown in Figure 1.22. After that, CBIR is able to retrieve the first correct result which is “image481” as shown in Figure 1.23 and Table 1.8

. Semantic engine then acquires a key ontological term from “image481” and traverses along Topic Maps related association in order to get a correct set of relevant images as shown in Figure 1.24 and Figure 1.25. However, since the query image is not one of characters existing in database so the Topic Maps finds only a single matched result, which is the query image that just has been saved during the annotation process.

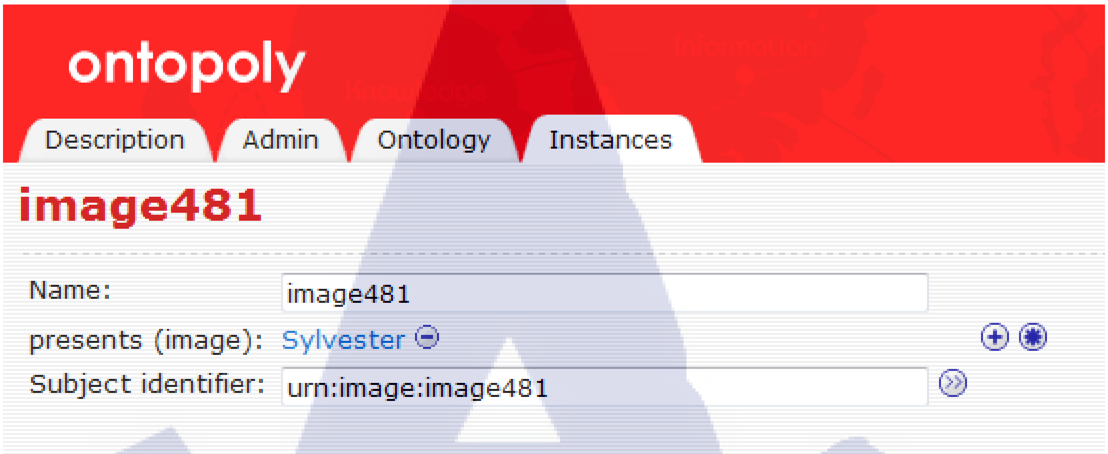

Figure 1.22 Test Scenario #4: Image Annotation in Ontopoly

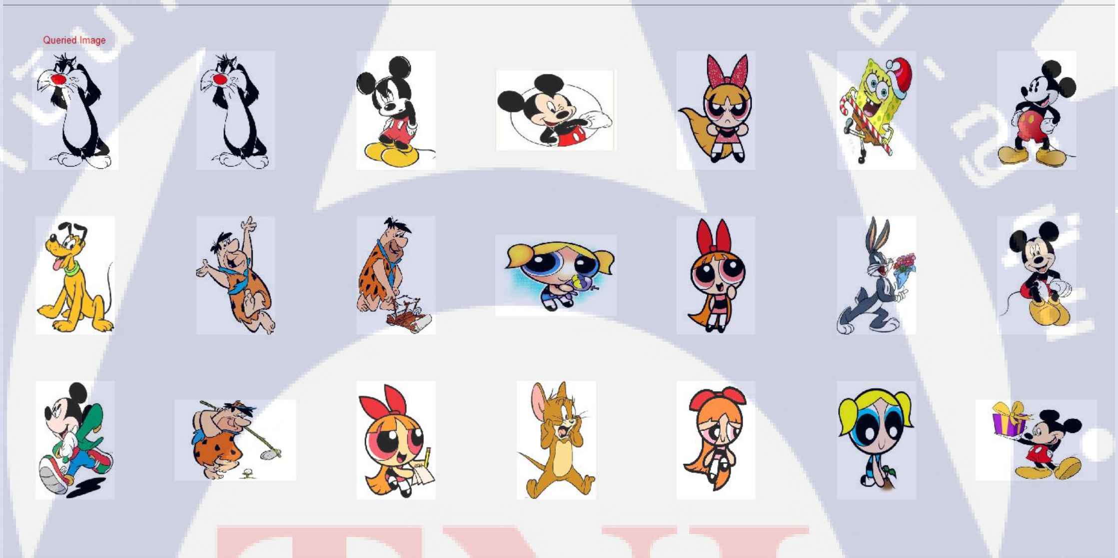

Figure 1.23 Test Scenario #4: New Results after Conducting Annotation

Table 1.8 Test Scenario #4: Similarity Distance after Conducting Annotation

Rank	Distance	Image Name
1	0	image481
2	9.4852	image107
3	10.567	image351
4	10.9578	image301
5	10.9752	image367
6	10.9874	image114
7	11.2663	image36
8	11.5093	image282
9	11.5363	image14
10	11.784	image461
11	12.0265	image46
12	12.0845	image304
13	12.1724	image235
14	12.193	image244
15	12.222	image151
16	12.2791	image412
17	12.2956	image457
18	12.4074	image471
19	12.4653	image156
20	12.4943	image376

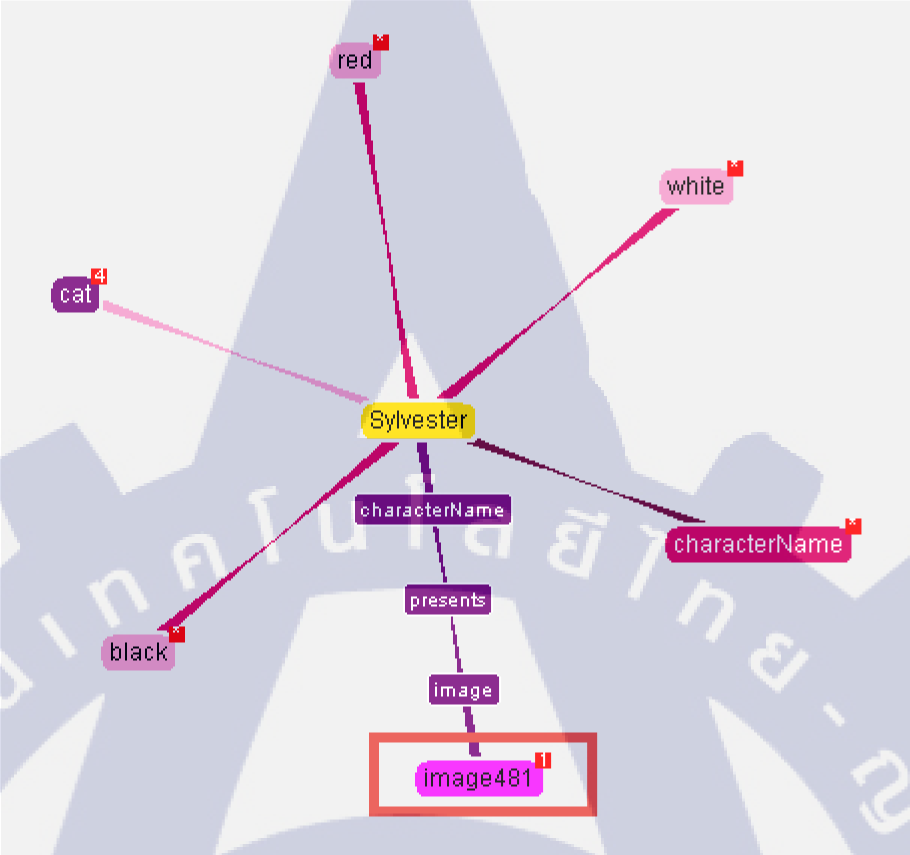

Figure 1.24 Sylvester: Topic Maps Network and Related Images

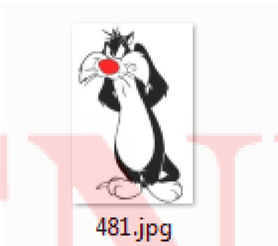

Figure 1.25 Sylvester Related Images

To summarize, the annotation is included as a part of semantic-based image retrieval mechanism as it is able to enhance retrieved results acquired by CBIR by accumulating human intellectual and growing database in more semantic manner.

4.3.2 Image Understanding Using TMQL

The proposed Topic Maps-based method can be used as another tool to perform a computer vision related technique as called Image Understanding. Actually, the origin concept of image understanding is to know the object inside an image based on well-analyzed image low-level feature. The proposed Topic Maps-based method can perform similarly by using defined ontological terms in some kinds of image indexing. This mechanism enhances the expressiveness of query languages [61] and thus allows machine to figure out what the objects are, what relationship is between each object and eventually making some decision for further action [62]. An important tool behind this calls tolog which is the Topic Maps Query Language (TMQL) inspired by Prolog [63]. Tolog is a logic-based query language used specifically for querying Topic Maps network. This language enables sophisticated query behind a complex network of objects and their relationships that the traditional relational database query language like SQL will not be able to handle. Query can be executed diversely from basis to complex filtering as exemplified in below tables

Table 1.9 Query for all related instances of a topic

Query Lines:
`instance-of($TOPIC, necktie)?`
Result:

`TOPIC`
`"Spongebob's necktie"`
`"Fred's necktie"`

Table 1.10 Query for character names wearing necktie

Query Lines:

`select $CHARACTER from`
`instance-of($TOPIC, necktie),`
`wears($CHARACTER : character-name, $TOPIC : costume-item)?`

Result:

`CHARACTER`
`"SpongeBob SquarePants"`
`"Fred Flintstone"`

Table 1.11 Query for character names wearing necktie and being human

Query Lines:

`select $CHARACTER from`
`instance-of($TOPIC, necktie),`
`wears($CHARACTER : character-name, $TOPIC : costume-item),`
`has-creature-type($CHARACTER : character-name, man : creature)?`

Result:
`CHARACTER` `"Fred Flintstone"`

Table 1.12 Query for character names wearing red clothes

Query Lines:
select $CHARACTER from wears($CHARACTER : character-name, $COSTUME : costume-item), has-costumecolor($COSTUME : costume-item , red : color)?
Result:
CHARACTER "Donald Duck" "Winnie the Pooh" "Superman" "Porky Pig" "Mighty Mouse" "Eric Cartman" "The Grinch" "Bart Simpson" "SpongeBob SquarePants" "Blossom" "Mickey Mouse" "Underdog"

Although tolog language allows logical-dynamic manner to perform complex query and filtering, the result may depend on how ontological network of Topic Maps is structured. This requires some knowledge of the Topic Maps domain to provide well-analyzed data categorization enhancing Topic Maps tracing and more accuracy of retrieved results. In other words, the better ontological network structured, the better precision acquired.

4.5 Chapter Summary

This chapter demonstrates experiment result of CBIR and the proposed Topic Maps-based method. The experiment is conducted based on test scenarios to control the environment and input factor. The assessment is carried out to compare between the results acquired from conventional CBIR and the proposed Topic Maps-based method by calculating precision and recall rate of each test scenario. The result of calculation presents that the proposed Topic Maps-based approach is able to enhance retrieval performance in terms of both accuracy and recall especially in the cases that image low-level feature of the first retrieved image of CBIR is correctly matched with images in database. The experiment also introduces the image annotation feature as a part of semantic-based image retrieval mechanism so as to deal with irrelevant results occurred by unknown or complicated query image. The annotation allows human intervention to assign ontological terms relating to the query image in the way which is similar to human knowledge accumulation manner. Regarding this matter, the irrelevant results can be optimized and the system is then able to retrieve more relevant images. Apart from that, some lines of tolog query language are exemplified to concretely present another feature of Topic Maps towards image understanding area.

Chapter 5
Conclusion

This chapter provides overall summary of this study including additional analysis and possible future works that can be extended.

5.1 Overall Summary

This study has been done with the objective to develop image retrieval system which is able to optimize the retrieved results by using Topic Maps-based approach to increase precision rate. Regarding this purpose, the implementation and experiment require diverse theories and knowledge which mainly are content-based image retrieval, semantic-based image retrieval and knowledge representation. Some specific tools are used in this study such as Ontopia software suite for Topic Maps ontological terms creation, navigation and visualization. The developed system is designed to consist of two main parts which are content-based image retrieval engine and Topic Maps engine including image database layer and ontological network of Topic Maps. The experiment is done based on different test scenarios to control environment and input factor. The result of experiment is observed twice for each case: once after the conventional CBIR processing and the other after the Topic Maps-based mechanism, following comparison approach in order to compare between the conventional CBIR and the proposed Topic Maps-based method. The experiment result is used to calculate precision and recall rate which represents that the proposed Topic Maps-based approach is able to enhance retrieval performance in terms of both accuracy and recall especially in the cases that image low-level feature of the first retrieved image of CBIR is correctly matched with images in database. However, the study also introduces the image annotation feature as a part of semantic-based image retrieval mechanism to compromise with irrelevant results occurred in some cases that the first retrieved result of CBIR is not correctly matched. The annotation allows human intervention to assign the ontological terms to unknown image and add it to the Topic Maps network for processing next time. Regarding this matter, relevant results are retrieved by both the conventional CBIR and the Topic Maps-based method. On the other hand, some lines of tolog query language are exemplified to demonstrate another feature of Topic Maps regarding image context understanding. The study leads to the conclusion that the Topic Maps-based approach does optimize the retrieved results acquired by the conventional CBIR method.

5.2 Additional Analysis

Although the proposed Topic Maps-based approach proves for impressive result in terms of enhanced precision and recall rate, it is remarkable that the Topic Maps-based image retrieval in this research is not implemented solely but extended from the conventional method. Some limitations are therefore still presented as shown in the test scenario and this prevents the proposed semantic-based approach from full functionality. In other words, the proposed method possibly produces better result if it can work independently.

5.3 Future Works

This study can be extended to some kinds of automatic ontological representation without employing human intervention. With this way, the semantic engine will be able to automatically generate proper ontological terms for each case depending solely on machine intelligence and reasoning. However, this kind of system requires well-complex structured data to drive machine logical calculation. Besides, the most important challenge on this area is that it is not easy to let machine think and understand the world as human does, thus to automatically generate the ontological terms which involves with high-level interpretation of human is still such the iceberg in this research area.

References

References

[1] J. Yu et al., “On combining multiple features for cartoon character retrieval and clip synthesis,” *IEEE Transaction on Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 42, no. 5, pp. 1,413-1,427, April 2012.

[2] Y. Yang et al., “Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning,” *Proceedings of The 17th ACM International Conference on Multimedia,* New York, USA, October 2009, pp. 311-320.

[3] N. Basile, “*Top 50 Cartoon Characters of All Time*,” [Online]. Available: http://animatedtv.about.com/od/showsaz/tp/top50chrctrs.04.htm. [Accessed: June 10, 2013].

[4] Wikipedia, “*Computer Vision*,” [Online]. Available: http://en.wikipedia.org/wiki/Computer_vision. [Accessed: March 12, 2013].

[5] G. Miller et al., “A conceptual structure for computer vision,” *Proceedings of 2011 Canadian Conference on Computer and Robot Vision, CRV*, St. Johns, Canada, May 25-27, 2011, pp. 168 - 174.

[6] Szeliski, “*Computer Vision: Algorithms and Applications*,” [Online]. Available: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf. [Accessed: July 12, 2013].

[7] Image Processing Learning Resources, “*Intensity Histogram*,” [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/histgram.htm. [Accessed: January 15, 2016].

[8] R. C. Gonzalez et al., *Digital Image Processing*, 3rd ed., New Jersey : Pearson Prentice Hall, 2008.

[9] R. C. Gonzalez et al., *Digital Image Processing Using MATLAB*, 2nd ed., New Jersey : McGrawHill, 2011.

[10] MathWorks, “*Create Image Histogram*,” [Online]. Available: http://www.mathworks.com/help/images/create-image-histogram.html. [Accessed: January 15, 2016].

[11] J. Huang et al., “Image indexing using color correlograms,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, San Juan, Puerto Rico, June 17-19, 1997, pp. 762-768.

[12] J. Huang and R. Zabih, "*Combining Color and Spatial Information for Content-based Image Retrieval*," [Online]. Available: http://www.cs.cornell.edu/rdz/Papers/ecdl2/spatial.htm. [Accessed: February 25, 2016].

[13] S. M. Singh and K. Hemachandran, "Content-based image retrieval using color moment and gabor texture feature," *International Journal of Computer Science*, vol. 9, no. 5, pp. 299-309, September 2012.

[14] W. L. Chao, "*Gabor Wavelet Transform and Its Application*," [Online]. Available: http://disp.ee.ntu.edu.tw/~pujols/Gabor%20wavelet%20transform%20and%20its%20application.pdf. [Accessed: January 25, 2016].

[15] M. A. Akinlar et al., "A novel matching of MR images using gabor wavelets," *IETE Technical Review*, vol. 30, no. 2, pp. 129-133, March-April 2013.

[16] H. Lidong et al., "Combination of contrast limited adaptive histogram equalisation and discrete wavelet transform for image enhancement," *IET Image Processing*, vol. 9, no. 10, pp. 908-915, September 2015.

[17] D. R. Nayak et al., "Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests," *Neurocomputing*, vol. 177, pp. 188-197, February 2016.

[18] X. Zhao et al., "A wavelet-based image preprocessing method for illumination insensitive face recognition," *Journal of Information Science & Engineering*, vol. 31, no. 5, pp. 1,711-1,731, September 2015.

[19] F. Hassan and M. Sajad, "Colour and texture feature-based image retrieval by using Hadamard matrix in discrete wavelet transform," *IET Image Processing*, vol. 7, no. 3, pp. 212-218, April 2013.

[20] Math.NET Numerics, "*Distance Metrics*," [Online]. Available: http://numerics.mathdotnet.com/Distance.html. [Accessed: January 25, 2016].

[21] H. H. Wang et al., "Image retrieval: techniques, challenge, and trend," *World Academy of Science, Engineering and Technology*, vol. 36, pp. 649-651, December 2009.

[22] V. Roth, "Content-based retrieval from digital video," *Image and Vision Computing*, vol. 17, no. 7, pp. 531-540, May 1999.

[23] X. Yu et al., “The related techniques of content-based image retrieval,” *2008 International Symposium on Computer Science and Computational Technology, ISCSCT 2008*, Shanghai, China, December 20-22, 2008, pp. 154-158.

[24] M. Haseyama and A. Matsumura, “A cartoon character retrieval system including trainable scheme,” *Proceedings of 2003 International Conference on Image Processing, ICIP 2003*, Barcelona, Spain, September 14-17, 2003, pp. 1,522-4,880.

[25] H. H. Wang et al., “Semantic gap in CBIR: automatic objects spatial relationships semantic extraction and representation,” *International Journal Of Image Processing*, vol. 4, no. 3, pp. 192-204, July 2010.

[26] N. R. Milton, *Knowledge Technology*, Italy : Polimetrica, 2008.

[27] V. Janev and S. Vraneš, “Applicability assessment of Semantic Web technologies,” *Information Processing and Management*, vol. 47, no.4, pp. 507–517, July 2011.

[28] S. Chongpratheep et al., “Enhancing application interoperability using semantic technology for E&P industry: a literature review,” *2012 International Conference on Business and Industrial Research, ICBIR 2012*, Bangkok, Thailand, May 17-18, 2012, pp. 98-102.

[29] H. P. Alesso and C. F. Smith, *Thinking on the Web: Berners-Lee, Godel and Turing*, New York : Wiley, 2008.

[30] D. A. Ali and Sh. A. Noah, “Semantically indexed and searched of digital images using lexical ontologies and named entity recognition,” *2010 International Symposium on Information Technology, ITSim 2010*, Kuala Lumpur, Malaysia, June 15-17, 2010, pp. 1,308-1,314.

[31] Ch. Y. Xu et al., “Image semantics recognition based on ontology and description logics,” *2009 International Conference on Multimedia Information Networking and Security, MINES 2009*, Hubei, China, November 18-20, 2009, pp. 33-36.

[32] B. Zheng et al., “Ontology for the object recognition in images,” *2009 International Conference on Multimedia Information Networking and Security, MINES 2009*, Hubei, China, November 18-20, 2009, pp. 472-479.

[33] Wikipedia, "*Knowledge Representation and Reasoing*," [Online]. Available: http://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning. [Accessed: July 15, 2014].

[34] A. Chebotko et al., "OntoELAN: an ontology-based linguistic multimedia annotator," *The IEEE Sixth International Symposium on Multimedia Software Engineering, ISMSE 2004*, Miami, USA, December 13-15, 2004, pp. 329-336.

[35] R. Davis et al., "What is a knowledge representation?," *AI Magazine*, vol. 14, no. 1, pp. 17-33, 1993.

[36] L. Stanescu et al., "A software system for viewing and querying automatically generated topic maps in the e-learning domain," *Informatica*, vol. 34, no. 4, pp. 441-450, 2010.

[37] Topic Maps Community, "*Topic Maps*," [Online]. Available: http://www.topicmaps.org/. [Accessed: August 15, 2012].

[38] ITSCJ, "*Information Processing*," [Online]. Available: http://www.itscj.ipsj.or.jp/sc34/open/1045.htm. [Accessed: September 12, 2012].

[39] K. Steiner et al., "Topic maps - an enabling technology for knowledge management," *12th International Workshop on Database and Expert Systems Applications, DEXA 2001*, Munich, Germany, September 3-7, 2001, pp. 472-476.

[40] N. Ellouze1 et al., "Overview of topic map construction approaches," *The 22nd International Conference on Advanced Information Networking and Applications – Workshops, AINA 2008*, Okinawa, Japan, March 25-28, 2008, pp. 1,642-1,647.

[41] A. Papastergiou et al., "Introducing automated melanoma detection in a topic map based image retrieval system," *The 6th WSEAS International Conference on Applied Computer Science, ACS 2010*, Hangzhou, China, April 15-17, 2007, pp. 452-456.

[42] A. Papastergiou et al., "Presenting a 3-tier system for managing melanoma images with topic maps," *WSEAS Transactions on Information Science and Applications*, vol. 4, no. 4, pp. 820-826, 2007.

[43] A. Papastergiou et al., “Applying topic map technology to create a knowledge base of melanoma images for diagnostic support,” *WSEAS Transactions on Information Science and Applications*, vol. 4, no. 2, pp. 406-412, 2007.

[44] A. Papastergiou et al., “Introducing an advanced topic map software tool towards the deployment of a TM-based system for managing melanoma cases images,” *WSEAS Transactions on Information Science and Applications*, vol. 4, no. 3, pp. 452-459, March 2007.

[45] A. Badii et al., “Semi-automatic knowledge extraction, representation and context-sensitive intelligent retrieval of video content using collateral context modelling with scalable ontological networks,” *Signal Processing: Image Communication*, vol. 24, no. 9, pp. 759-773, October 2009.

[46] A. Badii et al., “Semi-automatic annotation and retrieval of visual content using the topic map technology,” *The 1st WSEAS International Conference on Visualization, Imaging and Simulation, VIS 2008*, Wisconsin, USA, pp. 77-82, 2008.

[47] A. Badii et al., “Semantic-driven context-aware visual information indexing and retrieval: applied in the film post-production domain," in *IEEE Workshop on Computational Intelligence for Visual Intelligence, CIVI 2009*, Nashville, USA, March 30 - April 2, 2009, pp, 44-51.

[48] A. Badii et al., “The DREAM framework: using a network of scalable ontologies for intelligent indexing and retrieval of visual content,” *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology – Workshops, WI-IAT 2009*, Milan, Italy, September 15-18, 2009, pp. 551-554.

[49] D. Wang et al., “Retrieving information in topic maps: the case of TM4L,” *The 45th Annual Southeast Regional Conference, ACM-SE 45*, New York, USA, March 23-24, 2007, pp. 88-93.

[50] S. Chongpratheep and S. Mruetusatorn, “Strengthen image retrieval performance using knowledge-intensive network of topic maps: Survey and initial experiment,” *2014 International Conference on Business and Industrial Research, ICBIR 2014*, Bangkok, Thailand, May 15-16, 2014, pp. 194-199.

[51] Y. Xu, “Research of intelligent teaching system based on knowledge grid,” *The 2nd IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2010*, Beijing, China, September 24-26, 2010, pp. 826-830.

[52] R. Ota and M. Kimura, “Proposal of a topic map database indexing method,” *2012 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2012*, Seoul, South Korea, October 14-17, 2012, pp. 2,650-2,655.

[53] H. Lu et al., “Extended topic map: Knowledge collaborative building for distributed knowledge resource,” *2010 IEEE Network Operations and Management Symposium, NOMS 2010*, Osaka, Japan, April 19-23, 2010, pp. 128-135.

[54] A. Korthaus and M. Schader, “Using a topic grid and semantic wikis for ontology-based distributed knowledge management in enterprise software development processes,” *10th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOCW 2006*, Hong Kong, China, October 16-20, 2006, pp. 4.

[55] S. Pepper, “*The TAO of Topic Maps Finding the Way in the Age of Infoglut*,” [Online]. Available: http://www.ontopia.net/topicmaps/materials/tao.html# pep1999. [Accessed: July 25, 2012].

[56] Mimir, “*Online Knowledge Management System with Topic Map Technology*,” [Online]. Available: http://www.mimir.com.my/www/lang/en/okp?articleid =2122. [Accessed: January 25, 2016].

[57] MathWorks, “*File Exchange: Content Based Image Retrieval by Chez*,” [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/ 42008-content-based-image-retrieval. [Accessed: November 7, 2013].

[58] Ontopia, “*Welcome to the Land of Ontopia*,” [Online]. Available: http://www.ontopia.net/. [Accessed: September 22, 2012].

[59] L. M. Garshol, “*The Linear Topic Map Notation*,” [Online]. Available: http://www.ontopia.net/download/ltm.html. [Accessed: May 7, 2016].

[60] D. Gowsikhaa et al., “Construction of Image Ontology using low-level features for Image Retrieval,” *2012 International Conference on Computer Communication and Informatics, ICCCI 2012*, Coimbatore, India, January 10-12, 2012, pp. 1-7.

[61] B. Mohammed and F. Fariza, “Image understanding and the web: a state-of-the-art review,” *Journal of Intelligent Information Systems*, vol. 43, no. 2, pp. 271-306, October 2014.

[62] B. S. Morse, “*Lecture 21: Image Understanding*,” [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/iu.pdf. [Accessed: May 7, 2016].

[63] L. M. Garshol, “*A Topic Map Query Language*,” [Online]. Available: http://www.ontopia.net/topicmaps/materials/tolog.html#N224. [Accessed: May 7, 2016].

Appendix

Appendix 1

Image Low-Level Feature Vector

Image Low-Level Feature Vector

Below is example of low-level feature vector of an image no. 330 constructed from:

- HSV Histogram (Starting from column 1 to 32)
- Color Auto-Correlogram (Starting from column 33 to 97)
- Two First Color Moments for RGB (Starting from column 98 to 116)
- Gabor Wavelet (Starting from column 117 to 169)
- Discrete Wavelet Transform (Starting from column 170 to 190)
- Image Name (At column 191)

Col.	Value	Col.	Value	Col.	Value	Col.	Value
1	0.004610762	51	0	101	179.0622762	151	14.9325783
2	0.003004766	52	0.020862309	102	92.46119241	152	14.95510828
3	0.000794363	53	1.139082058	103	13.60125155	153	14.49207412
4	0.001882296	54	0	104	55.37095552	154	13.30809076
5	0.049215998	55	0	105	124.202618	155	13.56214101
6	0.011483733	56	0.020862309	106	230.6639852	156	12.21479041
7	0.001985909	57	0	107	13.73914606	157	10.66922008
8	0.003540098	58	0	108	56.61457064	158	11.67864828
9	0.007943635	59	0	109	124.2605093	159	13.32400866
10	0.003764592	60	0.008344924	110	197.2409651	160	12.7493105
11	0.003056572	61	0.035465925	111	11.16363037	161	12.3050472
12	0.00139877	62	0	112	46.30427993	162	12.95647594
13	0.149530289	63	0	113	87.02115745	163	12.88929271
14	0.01643987	64	0	114	147.70295	164	14.18755037
15	0.001899565	65	0.003477051	115	13.30617689	165	12.84149105
16	0.006769358	66	0	116	56.58298688	166	14.01601226
17	0.15218968	67	0	117	98.51629535	167	14.00860809
18	0.100037991	68	0.003477051	118	149.563552	168	14.95113754
19	0.01124197	69	0	119	13.13668773	169	15.25111156
20	0.021206051	70	0	120	54.42981195	170	15.26092604
21	0.116546937	71	0.055632823	121	108.09134	171	3.063205812
22	0.019461905	72	0	122	203.5452333	172	3.004437767
23	0.025419631	73	0.020862309	123	14.24703551	173	3.724010037
24	0.012398978	74	0	124	55.83797243	174	5.242936732
25	0.055536368	75	0	125	124.4021593	175	4.975335695

Col.	Value	Col.	Value	Col.	Value	Col.	Value
26	0.197053948	76	0	126	260.2188217	176	4.646738922
27	0	77	0	127	0.00458458	177	5.525651651
28	0	78	0.006954103	128	0.009700428	178	5.332010082
29	0	79	0	129	0.017560958	179	3.910142377
30	0	80	0	130	0.030200628	180	4.179034719
31	0	81	0.155771905	131	0.004563238	181	5.835032851
32	0.021585964	82	0.082753825	132	0.009252486	182	4.840006805
33	0.014603616	83	0	133	0.01694759	183	4.528827643
34	0	84	0.233657858	134	0.027604339	184	3.933012053
35	0.002781641	85	0.004867872	135	0.004311039	185	4.252556564
36	0.001390821	86	2.657858136	136	0.00871175	186	1.972556466
37	0.006954103	87	0.019471488	137	0.014850876	187	1.588671368
38	0.001390821	88	0	138	0.02485206	188	1.418664842
39	0.003477051	89	0	139	0.004434712	189	1.793856584
40	0.00069541	90	0.04867872	140	0.009460511	190	1.773900737
41	0	91	0	141	0.015395101	191	330
42	0.022253129	92	0.029207232	142	0.025070736		
43	0	93	0	143	0.004808859		
44	0.022253129	94	0	144	0.009970184		
45	0	95	0.194714882	145	0.016743274		
46	0	96	0	146	0.029542052		
47	0	97	208.2745667	147	0.004854019		
48	0.006954103	98	82.80768186	148	0.009955487		
49	0.004172462	99	199.5257365	149	0.01762254		
50	0	100	83.42021731	150	0.031869044		