Generating visual concept maps based on image annotation and commonsense knowledge acquisition

Alireza, Kashian

2008

Alireza, K. (2008). Generating visual concept maps based on image annotation and commonsense knowledge acquisition. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/20650>

<https://doi.org/10.32657/10356/20650>

GENERATING VISUAL CONCEPT MAPS BASED ON IMAGE ANNOTATION AND COMMONSENSE KNOWLEDGE ACQUISITION

ALIREZA KASHIAN

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2008

Acknowledgments

I wish to acknowledge all those people that have contributed to the work described in this thesis. I express my gratitude to all those who provided me their great ideas and helped me to design, implement and test our applications. Those who analyzed, criticized and sponsored the work also deserve my warm thanks.

This work would not have been possible without the support and encouragement of Prof. Robert Kheng Leng Gay, under whose supervision I chose this topic and began the thesis. He also helped me by providing resources and subjects, and offering direction and penetrating criticisms.

I would also like to thank Dr. Habib Mir Hosseini for bountiful commitment in every step of this project and his generous and wise guidance during its development.

My parents deserve special mention for their inseparable support and prayers. Their unflinching courage and conviction always inspire me, and I hope to continue, in my own small way, the noble mission to which they gave their lives.

I would like to acknowledge the help of many people during my research period:

Assoc Prof. Chris Khoo Soo Guan for the interesting dialogs and dedication in sharing his beliefs and ideas. He was so much kind to support our discussions.

Armineh Nourbakhsh for her great support with analyzing our data and commenting on the developed concepts. She did a great job on evaluation of algorithms.

Hashem Tatari for the superb code developing on server side especially for SYGY^{TAG} and SYGY^{Map}.

Saeed Akhavan for his breathtaking graphic design and Flash work for SYGY^{TAG} and SYGY^{Map}. His elegant interface design for the games is appreciated many times by the game players. His art is coined in our project.

I also extend my thanks to *Ehsan Zaery Moghadam, Sona Hassani, Shayan Shahand, Ahmad Sabouri, Mehdi Milanifard, Alireza Vazifedoost, Mohammad Shabestary, Shahram Sharif, Safoora Kashian* who shared their knowledge, proficiency and time with me to push this work forward.

Finally, I would like to thank NTU and Division of ICIS at school of electrical engineering as a great host for my research activities. I express my gratitude to ICIS division head, staffs at ICIS labs and graduate studies office.

Table of Contents

CHAPTER 1 INTRODUCTION.....	1
1.1 What is a Concept Map?.....	1
1.1.1 Usefulness of Concept Maps	2
1.1.2 Structure of Concept Maps	2
1.1.3 Concept Mapping as Learning Tool	5
1.2 Trends	6
1.2.1 Emergence of Collaborative Tagging Systems	6
1.2.2 Image Annotation	8
1.2.3 Human-based Computation	10
1.3 The Power of Visual Learning.....	11
1.4 Commonsense Knowledge Acquisition.....	12
1.5 Objectives	13
1.6 Research workflow and challenges	14
1.7 Organization of the Thesis.....	17
 CHAPTER 2 COMMONSENSE KNOWLEDGE ACQUISITION	 19
2.1 What is a Commonsense Fact?	19
2.1.1 Identifying the Commonsense Knowledge.....	19
2.1.2 Motivation for Collecting Commonsense Knowledge	20
2.1.3 Previous Works	20
2.2 Collecting from Contributors over the Web	20
2.3 Knowledge Acquisition	21
2.3.1 Knowledge Engineering	21
2.4 Methods of Knowledge Acquisition.....	23
2.4.1 Interviews	24
2.4.2 Structured and Unstructured.....	24
2.5 Bottleneck in Commonsense Knowledge Acquisition	25
2.6 Validating Commonsense Knowledge	26
2.7 Ambiguity in Commonsense knowledge.....	27
2.7.1 Number of Contributors	27
2.7.2 Lexical Ambiguity & Word Senses.....	28
2.7.3 Models of Data Entry	29
 CHAPTER 3 DEVELOPED TOOLS – COLLIMATOR.....	 30
3.1 Motivations.....	30
3.2 Versioning of Collimator.....	32
3.3 Collimator Terminology and Access Point	35

Table of Contents

3.4 Implementation and Architecture	35
3.4.1 Scalable Vector Graphics	35
3.4.2 AJAX for Data Communication	37
3.4.3 System Architecture	38
3.5 Collimator Features	41
3.5.1 User Interface Features.....	41
3.5.2 Semantic Web Features	42
3.5.3 General Capabilities	43
3.6 How to Use Collimator.....	46
3.7 Form of Data Entry.....	49
3.8 Applications.....	49
3.9 Possible Extension Areas.....	51
3.10 Related Work.....	51
 CHAPTER 4 GAMES FOR COLLECTING COMMONSENSE KNOWLEDGE	53
4.1 Design of a Useful Game.....	54
4.2 Fairness Analysis.....	56
4.3 SYGY ^{TAG}	56
4.3.1 How to Play SYGY ^{TAG}	58
4.3.2 Scoring algorithm in SYGY ^{TAG}	59
4.3.3 Tools for Players in SYGY ^{TAG}	61
4.3.4 Center of Attention	62
4.3.5 Application of Labels and Regions	63
4.3.6 Cheating in SYGY ^{TAG}	64
4.3.7 Quality of Labels and Selections.....	66
4.3.8 Experiment 1: Analyzing Database of SYGY ^{TAG}	67
4.3.9 Experiment 2: Commonsense Measurement through Time	68
4.3.10 Experiment 3: Change of Threshold for Center of Attention.....	70
4.3.11 Possible Improvements.....	71
4.4 SYGY ^{Map}	73
4.4.1 Basic Mechanism of SYGY ^{Map}	74
4.4.2 Game Conductor and Templates for the Facts	75
4.4.3 Using Templates or Natural Language.....	76
4.4.4 Scoring and Handling the Cheaters	77
4.4.5 Related Works	78
 CHAPTER 5 GENERATING VISUAL CONCEPT MAPS	80
5.1 What is a Visual Concept Map?	80
5.1.1 Focus Question for Visual Concept Maps.....	82
5.2 Frequency and Support of the Facts in SYGY ^{Map}	82
5.3 Structure of Visual Concept Maps.....	84
5.4 Technical Perspective	86

Table of Contents

5.5 Administrator Perspective	91
5.6 Similarity Measurement of Objects	95
5.6.1 Tversky's Contrast Model	97
5.7 Center-Based Visual Concept Maps	101
5.8 Similarity Map	105
5.9 Maximum Spanning Tree and Prim's Algorithm	108
5.10 Applications of Visual Concept Maps	110
5.11 Enhancing the Learning Process	111
5.12 Saving Visual Concept Maps	112
 CHAPTER 6 CASE STUDIES FOR EVALUATION	116
6.1 Center-Based Visual Concept Map for Fruits	116
6.2 Similarity Map for Mammals	119
6.3 Similarity Map for Vehicles	124
6.4 Expanded Similarity Maps	128
 CHAPTER 7 CONCLUSION AND FUTURE WORK	131
 PUBLICATIONS AND LECTURES	135
 REFERENCES	136
 BIBLIOGRAPHY	142
 APPENDIX I – USER INTERFACE FOR SYGY^{TAG} AND SYGY^{MAP}	148
 APPENDIX II – SCORING ALGORITHM FOR SYGY^{TAG}	155

Summary

Representing a domain of knowledge in a computer is a highly complex task that involves the knowledge representation itself, domain-specific particularities, and problems in extracting knowledge from people, among others. This problem becomes more complex when knowledge is dynamically introduced by a user who is not necessarily a computer expert. According to Novak [1], a formalism named “concept map” is extremely powerful for human education and communication. These simple nets would be auxiliary to computers, more specifically in tasks where structural domain knowledge is important. Concept maps are also useful for reasoning, information retrieval and they even can be used for knowledge simulation and learning purposes.

This research is a new approach for generating visual concept maps using annotated digital images. We have developed an online environment where a group of users can enter their knowledge collaboratively. We analyze the entries and generate a new visual map in which the nodes and links are based on common consensus of the users.

Briefly, the following steps are taken to reach a new set of results in this thesis:

- 1- A survey on Concept Maps, Online Collaborative Tools, Manual Annotation of Digital Images, Bottlenecks for Commonsense Knowledge Acquisition is done.
- 2- A new online tool (Collimator)¹ was developed for annotating digital images.
- 3- A new online game (SYGY)² was developed for locating, labeling and generating facts (statements) about visual objects inside digital images
- 4- The concept of Visual Concept Maps is proposed.
- 5- A new formula for similarity measurement between two objects is derived.
- 6- Two sample models for constructing Visual Concept Maps are proposed.

Collimator is short for Collaborative Image Annotator. Collimator has been our first endeavor to introduce a new web-based semantic application for annotating digital images. Users could load any web accessible image and start drawing regions (part of an image) using shapes such as rectangles or circles. Later, the user has to assign labels to those regions. Collimator demonstrates a collaborative tagging environment, since users are able to tag different regions in any image collaboratively.

¹ <http://www.xmlweb.info/collimator>

² <http://www.sygy.org>

In the next phase of our research, we introduced a better interface for higher rate of participation in labeling the digital images. Following a recent successful approach for collecting commonsense knowledge in Carnegie Mellon University³, we designed a new online game to motivate users to label images while having fun. The game, that is named SYGY, is a commonsense collaborative tool which is divided into two parts: SYGY^{TAG} and SYGY^{Map}.

In SYGY^{TAG}, players have to draw arbitrary regions in a digital image and assign labels to that region. Score is given based on matching each player with previous players who have played on the same image.

In SYGY^{Map}, the players generate facts about objects in the digital images using the labels created in SYGY^{TAG}. The facts are small pieces of knowledge representing the features of the objects. For example: “The lion is Wild”.

The algorithms for generating two models of visual maps are proposed in this work. These two models are called Center-Based Maps and Similarity Maps. Center-Based Map is supposed to express the features and attributes of objects while Similarity Map is illustrating the similarity degree between pairs of objects. By saying objects, we refer to those entities which annotators (volunteers) would find in the digital images while they are using our annotation tools.

Generated Visual Concept Maps are customizable for different bodies of knowledge, depending on the purpose of the user. A teacher would use them to teach concepts visually in a class while a researcher would benefit it for surveying the children’s brain learning abilities. Our research could be expanded in the future through mining labels and facts in SYGY databases to discover the potential of synergistic knowledge acquisition over the web. It is also helpful to use current system architecture in order to run online surveys about commercial products and services to collect comments from consumers and finally generating a meaningful map to illustrate a consensus of users for buying or not buying a product or service.

During the course of this study, two sample knowledge acquisition systems were developed (Collimator and SYGY). We found that using games to collect small pieces of knowledge is promising, as the level and quality of contribution improves over time and is accumulative and sustainable. We were also able to test and validate an algorithm for measuring the similarity between two objects based on their features vector. A few

³ <http://www.espgame.org>

tests for comparing different sets of objects were performed and the outputs showed that the algorithm was effective in achieving the desired results. One of the important achievements was the discovery of an innovative method for generating dynamic scores for the image labeling game. This method was fully implemented and tested in SYGY^{TAG} and proved to be an efficient way for making fair and dynamic collaborative games. The scoring method is fully explained in Appendix II.

As a final goal for our study, we explored a new generation of concept maps called Visual Concept Maps. We successfully integrated our major modules in the last phase of the research to produce a richer structure to represent a map of relationships between objects and their features. The new integrated system proved to be practical for building new knowledge maps based on consensus of all collaborators. Two sample outputs of the system are the Similarity Map and the Center-Based Map which were proposed and evaluated through three case studies (see section 5.7 and 5.8).

The most significant and interesting results of the work were: A new toggling feature between nodes of a concept map which we believe is completely original (see chapter 5), a new dynamic way of rating players where the scores of a player would be changed over time (see chapter 4), and finally the unique definition of the top-down structure of Similarity Map which leads to a new method of classifying objects into similar categories. This method has been introduced for the first time here in this work (see chapter 5).

List of Figures

Figure 1-1: Simple Concept Map	3
Figure 1-2: Complex Concept Map	3
Figure 1-3: Hierarchical and Nonhierarchical Concept Maps	4
Figure 1-4: A portion of Tag cloud as displayed by well-known collaborative tagging system “del.icio.us”	7
Figure 1-5 Relationship between main concepts of current thesis in a form of concept map	16
Figure 3-1: Primitive design of Collimator (Version 1)	33
Figure 3-2: Screenshot of the 2nd version of Collimator	34
Figure 3-3: Screenshot of the 3rd version of Collimator	34
Figure 3-4: Output of an example SVG Code on the screen	37
Figure 3-5: Traditional web application model compared to AJAX model	38
Figure 3-6: AJAX Technology and the Layering System of Collimator	39
Figure 3-7: Core of Collimator	40
Figure 3-8: Overlapping message for two regions	45
Figure 3-9: Illustration of the annotation procedure in Collimator	48
Figure 4-1: Screenshot from SYGY ^{TAG} session. The Ball is selected	58
Figure 4-2: Using Spray tools to select face region	61
Figure 4-3: Created Labels and Regions in SYGY ^{TAG}	62
Figure 4-4: Sample for Center of Attention – higher than 50% agreement	67
Figure 4-5: Screenshot from SYGY ^{TAG} Analyzer	68
Figure 4-6: Change of total score for a meticulous player over 30 days	70
Figure 4-7: Change of total score for an inaccurate player over 30 days	70
Figure 4-8: Exploring different thresholds for Center of Attention for “Stairs”	71
Figure 4-9: Screenshot from SYGY ^{Map} graphical user interface	74
Figure 4-10: Fact-generation sheet in SYGY ^{Map} game for sample object “Frog”	75
Figure 5-1: Sample visual concept map about red fruits	81
Figure 5-2: Elephant as a object selected in SYGY ^{TAG}	83
Figure 5-3: Sample two nodes of VCM	84
Figure 5-4: Flowchart of VCM Generation system	86
Figure 5-5: Sample centre-based diagram showing toggling process	90
Figure 5-6: Sample Similarity Map, visualizing the classification of garment based on the season they’re made for	91
Figure 5-7: Sample object pair and their common and distinct features	99
Figure 5-8: Sample centre-based VCM at the initial stage	103
Figure 5-9: Center-Based VCM after selecting a feature to explore (second stage)	104
Figure 5-10: Toggling between Object and Feature in sample Center-Based VCM	105
Figure 5-11: Sample similarity tree showing classes of animals	108
Figure 5-12: Similarity Map for similar animals to horse	114
Figure 6-1 Center-based Visual Map for circular fruits	118
Figure 6-2: 12 Mammals selected for the games	120
Figure 6-3: Complete graph for similarity of the objects	122
Figure 6-4: Similarity Map for animals	123
Figure 6-5: Similarity Map for vehicles	127
Figure 6-6: Expanded Similarity Map for animals	129
Figure 6-7: Expanded Similarity Map for animals	130

List of Tables

Table 3-1: Characteristics of Collimator	46
Table 5-1 Sample Facts for a VCM.....	85
Table 5-2: Fact frequency percentiles and their corresponding adverbs for edge labels in the centre-based VCM.....	102
Table 5-3: Adverbs of degree for Similarity measures between two objects to be used for the links.....	107
Table 5-4: Pseudo code for Maximum spanning tree using Prim's Algorithm.....	109
Table 5-5: XML file Generated for sample Visual Concept Map of Animals.....	115
Table 6-1: List of selected fruits.....	116
Table 6-2: List of selected 16 features for fruits	117
Table 6-3: Fact frequency for each fruit.....	117
Table 6-4: List of Animals	119
Table 6-5: List of Features	119
Table 6-6: Fact frequency for animals.....	120
Table 6-7: Matrix of Similarity Map for Animals.....	121
Table 6-8: List of vehicles	124
Table 6-9: List of features for vehicles.....	124
Table 6-10: Fact frequency for vehicles.....	125

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript + XML
CBIR	Content Based Image Retrieval
CM	Concept Map
CoA	Center of Attention
Collimator	COLlaborative IMAge annotaTOR
CSS	Cascading Style Sheets
CXL	Concept mapping eXtensible Language
DOM	Document Object Model
GIF	Graphics Interchange Format
GWT	Google Web Toolkit
HTML	Hypertext Markup Language
ICIS	Information & Communication Institute of Singapore
JPEG	Joint Photographic Experts Group
JVM	Java Virtual Machine
MRT	Mass Rapid Transportation - City train in Singapore
MST	A Minimum Spanning Tree
NTU	Nanyang Technological University
OSI	Open Systems Interconnection
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
SYGY	SYncretic synerGY
VCM	Visual Concept Map
W3C	The World Wide Web Consortium
XML	eXtensible Markup Language

Chapter 1 Introduction

Representing a domain of knowledge in a computer is a highly complex task, involving the knowledge representation itself, domain-specific particularities, and problems in extracting knowledge from people, among others. This problem becomes even more complex when knowledge is dynamically introduced by the user, who is not necessarily a computer expert. According to Novak [1], a formalism named “concept map” is extremely powerful for education and communication in humans. These simple nets may also be useful to computers, more specifically in tasks where structural domain knowledge is important. Concept maps are even useful for reasoning and information retrieval. They can also be used for knowledge simulation in free format style. In this research we are proposing new series of concept maps which are generated collaboratively over the web. The maps visually represent the concepts and that is why we decided to call them visual concept maps (in short VCM).

1.1 What is a Concept Map?

A concept map is “a two-dimensional, hierarchical, node-link diagram that depicts verbal, conceptual, or declarative knowledge in succinct visual or graphic forms” [2]. They are actually graphical tools for organizing and representing knowledge. The map is composed of concept labels, each enclosed in a box or oval; a series of labeled linking lines, and an inclusive, general-to-specific organization.

A concept could be defined as a conceivable regularity in events or objects. We may designate a concept through a label and this label is usually a word but it could be also a symbol or some other kind of visual illustration. When two or more concepts are connected to each other through linking phrases, a proposition is constructed. Propositions are meaningful statements and sometimes they are called semantic units, or units of meaning. In this thesis we use the term “Facts” instead of propositions. Facts are representing the realities in our real life. For example, a statement like “Atoms are located inside Molecules” is a fact. Atoms and Molecules are basic concepts.

Concept maps were developed in 1972 in the course of Novak’s research program at Cornell where he sought to follow and understand changes in children’s knowledge of science [3]. In that time, Novak was using Ausubel’s theory [4] of cognitive

assimilation which declares that learning takes place by the incorporation of new concepts and propositions into existing concept and propositional frameworks held by the learner. This knowledge structure is usually called cognitive structure and each person holds such structure in his mind. Novak tried to draw simple cognitive structures on the paper and that was the time concept maps were born as a tool. Nowadays, this tool is not only for use in research, but also is useful in many other applications.

1.1.1 Usefulness of Concept Maps

Concept mapping can be used to build up ideas and make a connection between complex knowledge structures and help people to communicate their own ideas and transfer the knowledge. Once we build a concept map, it can be expanded to include new concepts. People can use concept maps to make a link between new knowledge and their existing knowledge [5]. Concept maps can also be used to compare the old knowledge with new knowledge and even help to assess the level of understanding, especially in classes. Teachers can use concept maps both in teaching and assessing. If a student has the ability to draw comprehensive concept maps, it means he has understood the domain of knowledge and the link between different ideas and knowledge elements. Usually science teachers at primary school and secondary school use concept maps for teaching. There have been several academic research activities to find out what a good concept map is [6] and how would it be possible to assess students' learning based on concept maps.

In short concept maps are used:

- To generate ideas (brain storming, etc.)
- To design a complex structure (long texts, hypermedia, large web sites, etc.)
- To communicate complex ideas
- To aid learning by explicitly integrating new and old knowledge
- To assess understanding or diagnose misunderstanding

1.1.2 Structure of Concept Maps

Concept maps can be a simple or complex. The number of nodes and links depends on the amount of information intended to be reflected in the concept map and also the required granularity. There are several methods to draw concept maps. Usually it is

```

graph TD
    A[Easing Student Fears] -- by --> B[Showing support]
    A -- includes --> C[Breaking assignments down into steps]
    A -- relate to --> D[familiar concepts]
    B -- provides --> E[confidence]
    C -- includes --> F[goals]
    C -- includes --> G[purpose]
    C -- includes --> H[steps]
    D -- relate to --> I[past material]
    D -- encourage --> J[familiarization]
    E -- with --> K[material]
    H -- provides --> L[simplification]
    I -- provides --> L
    L -- of --> K
    J -- of --> K
  
```

[illegible]

Page 3

As mentioned earlier, the core element of a concept map is a **proposition**, which consists of two or more concepts connected by a labeled link. In a concept map, propositions are connected to each other to form a hierarchical, branching, and dendritic structure that represents the organization of knowledge in long-term memory (Zeilik [7]).

Different geometric shapes of nodes like rectangles, ovals, circles and other shapes, as well as different spatial configurations, icons, colors and sizes, may be used for symbolizing different semantic aspects of knowledge elements and for conveying meaning. Multiple linkages between concepts may depict how each concept is related to other concepts belonging to different sections of a concept map. Concept maps are particularly useful for representing networks of concepts, where links do not only connect adjacent concepts, but are also linked to concepts in different sections of the concept map.

The early descriptions of concept maps explain that they are hierarchical (as Novak describes). The hierarchical map has a dominant concept which is what the concept map is 'about' and this often appears at the top where the other concepts are represented in a hierarchical fashion, but nowadays there is no rigid rule on this. One can often find non-hierarchical maps, for example a concept map of the Nitrogen cycle. Figure 1-3 illustrates two types of concept maps.

Figure 1-3: Hierarchical and Nonhierarchical Concept Maps

Different types of concept mapping are proposed: Freestyle mapping (if a map is self-produced), guided mapping (if students use a template for mapping), and information mapping (if an expert map is presented to students by an instructor). Research has shown that all types of concept mapping can be useful in improving outcomes in different domains, e.g. education and counseling, depending on variables

like task requirement, structure of content and individual prerequisites of the user (e.g. [8]; [9]).

Due to variations in mapping conceptual knowledge structures, authors of mapping tools use different labels, e.g. cognitive maps, knowledge maps, network maps or concept maps. For example, concept mapping is sometimes called cognitive mapping, because of the network structure of represented concepts. The labels "knowledge mapping" and "semantic networking" are used with a quite similar meaning as "concept mapping" [10]. Concept mapping combines features of "networking" and "mapping", because it is intended to represent both the network structure of concepts in semantic memory and its visualization in a map by means of the spatial configuration and special representational features to represent and emphasize meaning.

1.1.3 Concept Mapping as Learning Tool

Many learners and teachers are surprised to see how this simple tool facilitates meaningful learning and the creation of powerful knowledge frameworks that not only permit utilization of the knowledge in new contexts, but also the retention of the knowledge for long periods of time [11], [12]. There is still relatively little known about memory processes and how knowledge finally gets incorporated into our brain, but it seems evident from diverse sources of research that our brain works to organize knowledge in hierarchical frameworks and that learning approaches that facilitate this process significantly enhance the learning capability of all learners [13].

Concept Maps have a wide range of usage in Teaching & Learning environments. By reading the concept map, an instructor can:

1. Find out how students have understood the scientific topic.
2. Check if students have any misunderstanding about concepts.
3. Evaluate the complexity of relationships between concepts which the student depicts.

Students can also benefit from concept maps in following ways:

1. To learn course material: Students can use concept maps to take notes during class time and organize them into independent concepts. Any time they refer back to their concept map, they can review it much faster and remember the concepts much easier.

2. To integrate course content: Students can integrate different material they have learned throughout the semester. It means the old and new knowledge will be combined and gives them a bigger picture of what they have learned during the semester.
3. To integrate material across different courses: Sometimes students have different courses in different semesters but usually they cannot find the relations between these courses. Concept maps can help them to connect different material they have learned in different courses.
4. To assess their own learning: Students can assess themselves each time they look on their own concept maps which have been extended in different time periods. Since their conceptual understanding will grow during the semester, they can check their old concept maps and find out how much they have learned and what misconceptions they held.
5. To provide feedback to the students: Students can show their concept maps to the instructor and receive his comments regarding their concept maps. This way they can use feedback efficiently.

1.2 Trends

During the course of this research project, we progressively got engaged with a few interesting topics in the computer science field. We were realizing that each topic has something valuable to be considered that could be used as a piece for our bigger puzzle. We benefited from studying and learning a few concepts such as collaborative tagging, image annotation and collective intelligence. These concepts helped us fundamentally to plan our road map for this work. Before we embark on explaining our goals and objectives of the project, it would be useful to review these topics briefly.

1.2.1 Emergence of Collaborative Tagging Systems

If many users add metadata in the form of terms, keywords (also called tags) to share content like web pages or online digital images then we have a mechanism which we can call “Collaborative Tagging System”. It is collaborative since users are collaborating to mark a shared content. The content which is collaboratively tagged

could be used for future navigation, filtering and search. We know that organizing contents on the web is not a new phenomenon, but the collaborative form of organizing and tagging is something new which is gaining popularity on the web during the past few years. Two famous collaborative tagging systems are Flickr[14] and De.li.cio.us[15] which were both developed in academic environments but later Yahoo! bought them for commercial purposes.

Metadata helps to identify, describe and manage the digital information resources on the web. The kind of metadata to tag a digital resource could be controlled or uncontrolled. If a controlled vocabulary is used to tag information resources, it means that a predefined set of keywords is applied in which the relationships between keywords are set and they are assigned into specific categories. The user is limited to these defined categories and keywords. But in uncontrolled vocabulary, the users are free to choose any keyword when they want to tag a resource. In collaborative tagging systems we usually prefer to use uncontrolled vocabularies since organization of the information will be combined with one's personal information space. Since people would annotate a single concept with different keywords, the number of used keywords will be expanded. The more often a tag (keyword) is used to describe different resources; it will be rated to be more popular. As seen in Figure 1-4, popular tags are shown in bigger size.

Figure 1-4: A portion of Tag cloud as displayed by well-known collaborative tagging system "del.icio.us"

In one of the few research studies to date Scott et al [16] have analyzed the data gathered from a few Collaborative Tagging systems including del.icio.us. They tried to study the regularities in user activities, tag frequencies and the nature of tags used. They have reported that collaborative tagging users exhibit a great variety in their sets of tags. It means some users have many tags in their history and some have few. Tags

themselves vary in frequency of use, as well as in what they describe. It is also interesting to see that minority opinions can coexist alongside extremely popular ones. The prevalence of tagging with a very large number of tags demonstrates that a large amount of tagging is done for personal use rather than public benefit. Nevertheless, even information tagged for personal use can benefit other users. For example if many users find something funny the likelihood is that someone else would also find it interesting and funny. [16]

1.2.2 Image Annotation

By growing the World Wide Web and development of technology, more non-textual information has been generated. This non-textual information includes digital images, video and voice files. With the advent of digital cameras and other devices such as PDAs and mobile phones which are equipped with cameras, more and more digital images are created each day. Large amounts of these digital images are uploaded to servers and are published in blogs and image collection websites. This means searching the web to find a desired image has turned into a challenging task. Search engines are using sophisticated algorithms to find the most relevant images to each query but since these algorithms are mostly keyword based, they are inefficient to answer semantic questions.

In the past few years more interest has been shown in image annotation to develop different methods of annotating images. Image annotation is the process of assigning metadata to digital images. This metadata will describe the image and could be used for finding images. The annotation level depends on how someone would like to describe an image. Sometimes it is just intended to describe the whole concept of an image and sometimes it is important to pay more attention to context of the image. Annotating images on a small scale for personal usage can be relatively simple; however large scale image annotation is notoriously complex.

There are three main methods to annotate images. These methods are different based on human involvement in the annotation process. Every method has its own pros and cons. Here we do not claim which method is the best, since according to our experience, each method is just fitted to special category of applications. Some researchers claim that manual annotation is tedious and time consuming so they are not going to survive in the long term while some others claim that automatic methods are

not accurate and they do not provide image understanding. It is good to have a closer look at these three methods.

Manual Annotation: Manual annotation of image content is considered a “best case” in terms of accuracy, since human will make decisions to choose the best keywords. Humans can identify different objects easily and describe the relationship between objects. Image understanding is also another factor which humans are capable of doing and as a result, manual annotation produces much more fruitful metadata for image search and retrieval. It is obvious that manual annotation is time consuming and involves a tedious process. In addition, manual annotation may also introduce retrieval errors due to incorrect user inputs or perhaps misunderstanding the subject which is annotated.

Semi-Automatic Annotation: We call this method semi-automatic annotation because it depends on the user’s interaction to provide an initial query and feedback and the system’s capability for using these annotations as well as image features in retrieval. In this method, the user provides feedback about the retrieved images then the system annotates them. The annotation process and the relevance feedback process are usually integrated together. The system uses automatic feature extraction methods to find relevant images to queries. This automatic process will be combined with user feedback to produce the annotation. [17]

Automatic Annotation: “Automatic image annotation is the process by which a computer system automatically assigns metadata in the form of captioning or keywords to a digital image.” [18] Automatic Image annotation is a technique to provide semantic image retrieval via text descriptions. It is the process of automatically assigning labels to images with pre-defined set of keywords [19]. The automatic means that humans will not be involved in the annotation process. The computer will use image processing methods to analyze the content and compare it with a set of predefined patterns. Automatic annotation is usually used to identify visual objects. These visual objects should be clearly depicted within the image. Automatic annotation is also unable to understand the relations between objects. These relations are important factors for image understanding.

1.2.3 Human-based Computation

Although computer science researchers have made tremendous progress to enhance the power of processing over the past 50 years, computers are still lagging in conceptual intelligence. They even have restricted perceptual capabilities that make them absolutely feeble compared to a human's cognitive level. Many jobs for human's brain are taken for granted while in some cases, using the computers to do the same job does not guarantee a proper result in a needed short time.

If we assume that the human brain is a processor and we have a large distributed system, each person is a small computation cell. All these cells together can come up with massive computational power. Such a "human computation" paradigm has enormous potential to address problems that computers cannot yet tackle on their own.

Human-based computation is a technique when a computational process performs its function via outsourcing certain steps to humans [20]. This approach leverages differences in abilities and alternative costs between humans and computer agents to achieve symbiotic human-computer interaction.

In traditional computation, a human employs a computer to solve a problem: a human provides a formalized problem description to a computer, and receives a solution to interpret. In human-based computation, the roles are often reversed: the computer asks a person or a large number of people to solve a problem, then collects, interprets, and integrates their solutions.

Unlike computer processors, humans require some incentive to become part of a collective computation. In different human-based computation projects people are motivated for different reasons. They may be curious and like to see if their contribution works or they simply prefer to be a part of a project. Some people like to be monetarily compensated while some are esthetically satisfying themselves.

In our project, we designed an online game package called SYGY which collects small pieces of commonsense knowledge. In our game, players are asked to find visual objects in digital images and assign labels to them. According to latest image processing and pattern recognition research results, it is impossible for computers to analyze and identify all sets of objects in different shapes, sizes and colors in most digital images. We still do not have such large database of training sets for patterns. Such a database could have billions of records. We believe that this kind of knowledge (location of objects and their names) could be easily collected from humans. We also found that

using games would be a great method for encouraging people to participate in the process and that was the reason for designing and developing SYGY in the context of our research. The game is discussed in more detail in chapter 4.

1.3 The Power of Visual Learning

As much as the efficiency and reliability of the processes of information collection and manipulation plays an important role in information management systems, the efficiency of information representation techniques will affect the brain's ability to interpret and comprehend the body of knowledge. In our increasingly visually driven society, the ability to create and interpret imagery is as imperative as the ability to read and write and to listen and speak.

Visual representation of information is a commonly employed technique for large data collections due to its noticeable expressiveness and summarization potential. Visual features such as shape, color, size and texture are analyzed using the pre-attentive memory of the human brain which highly accelerates the process of observation and interpretation compared to other forms of representation of data for human beings. These features are also processed in parallel which adds the advantage of simultaneity compared to textual representation where information is sequentially presented and processed by human readers.

The very basic meaning of “visual learning” is described through visual psychophysics: visual learning is the recognition of objects, patterns as well as seeing movements and colors, initially without regarding aspects of cognitive psychology. Visual learning is a teaching method in which ideas and concepts are represented and illustrated through graphical depictions such as images, signs and shapes. By saying a powerful method we refer to its ability to transfer significant amounts of information to the human brain in a very short time, where the information is expected to be stored and maintained for longer periods of time since vision has proved more reliable in learning than other senses [21]. Visual learning techniques help the learner clarify thoughts, organize and analyze information, integrate new knowledge and think critically. Many systems also use this type of information representation to interact with illiterate, young or non-native users, unable to make use of data represented in textual or audio format [22].

As we are focused on constructing concept maps to be used in learning environments, we have chosen to investigate how concept maps would be beneficial for students. We identified that by representing information spatially and with images, students are able to focus on meaning, reorganize and group similar ideas easily, and make better use of their visual memory. Even teachers and instructors can assess and express the meaning of concepts in a prolific way.

In our proposed framework, an instructor may challenge the students to work collaboratively and label the images online and at the end; students will be able to review the results through constructing visual concept maps. We realized that the constructed visual map had a great effect on efficiency for students to compare the concepts and even understand the relation between two concepts in a bigger map.

1.4 Commonsense Knowledge Acquisition

Commonsense, as the term itself tries to reveal, is what people in common would agree upon. It is a common natural understanding among humans. In reality, commonsense knowledge is a true statement about the world that is known to most humans. The knowledge and experience most people have, or are believed to have is called commonsense knowledge and commonsense experience. In contrast to commonsense knowledge, we have expert knowledge which is not owned by ordinary people.

Knowledge acquisition is the transformation of knowledge from the forms in which it is available in the world into forms that can be used by a knowledge system and interpreted by machines. It is commonly said that Knowledge Acquisition is about extracting knowledge from sources of expertise. These sources are in fact those people who are expert in the specified domain of knowledge. Recently, by growing collaborative knowledge systems over the web, knowledge acquisition is not just a job for experts but even non-experts could contribute and stay there as the source of knowledge. For example, commonsense knowledge could be acquired through ordinary contributors who voluntarily like to share their knowledge in an online website. Recently, researchers are mainly focused on the methods of acquiring knowledge and how to make the job easier and fun for contributors. [23]

In our project, we are trying to collect the commonsense knowledge using our developed online game which entertains the users while they contribute. It was important for us to make sure that the knowledge we collect is common among the community of users. There are basically two types of knowledge that we collect: 1- Location of objects in digital images 2- Some basic facts about the objects. The facts are about attributes and functionality of objects through studying our primary users, we understood that almost all users (volunteers) are able to identify objects in digital images and they can simply provide a label for those objects. We even tested the ability of users to create a few basic facts related to those identified objects. The test proves that all users have the ability to come up with basic knowledge about attributes and functionality of the objects in the digital images. The applications that we used to collect the knowledge are explained in chapters 3 and 4. At the final stage, we used this collected knowledge to develop visual concept maps which are explained in more detail in chapter 5.

1.5 Objectives

The primary goal of this study was to construct a new series of visual concept maps through collaboration of people. This also involved conception of the framework, development of the tools and the design of games/questionnaires to enable this. The two key questions which were leading us to main objective of our research were: 1) how to benefit from wisdom of a community and 2) how to represent commonsense knowledge in the form of concept maps. The key difference from previous work was the use of digital images as the nodes of the concept map instead of the usual verbal nodes. This intention led us to choose the option of developing our own image annotation system which could be used to collect small pieces of information. To achieve the main objective, we needed to define a list of smaller goals/requirements. These were:

1- An efficient online knowledge acquisition system. This system was to enable us to have a sustainable and reliable source of contribution from users.

2- A simple method for annotating the content of digital images. This goal led to the development of the online collaborative tagging system.

3- A new form of concept maps had to be defined. The new form had to be consistent with the nature of the knowledge acquisition system which is based on consensus of users.

4- The software and databases for testing the ideas had to be designed and implemented.

1.6 Research workflow and challenges

The way people can collaborate efficiently to share their knowledge has been a research topic for decades. Emergence of the internet brought new opportunities while at the same time, new scientific questions appeared and multi-dimensional research challenges arose. On the same path, our chosen research challenge is among those multidisciplinary topics.

The new maps which are defined in this thesis are called Visual Maps. The main feature of Visual Maps is their ability to represent concepts visually. As a result some of the nodes are replaced by digital images while others remain verbal for better clarity. This way, concept maps have stronger impact on human minds as they express the meanings visually. That is why it is much easier to learn and memorize concepts by reviewing the visual maps. People have the main role of sharing and contributing their knowledge using our developed system in order to construct such visual maps. The maps are actually representing the common consensus of a group of volunteers; thus we may even name the maps as collaborative visual maps. During our course of study we understood that having more contributors means stronger results so we were convinced that we had to seek for methods of attracting more people to use our system.

A visual map is a great tool to teach new concepts to children, illiterate and old people since their visual learning ability is much stronger than their capability to read and comprehend complex verbal expressions.

In our project, we passed through all the steps of collecting and processing pieces of knowledge to construct visual maps. The data collection is done through three developed online applications, namely Collimator, SYGY^{TAG} and SYGY^{Map}. We took volunteers (students) and an administrator (teacher) as our sample community of users. A group of students (with the help of the teacher) would go through the following steps to generate a sample visual concept map collaboratively:

Step 1 – Labeling (tagging) a set of digital images using the Collimator or SYGY^{TAG} game. The image corpus is prepared by the administrator.

Step 2 – Evaluating facts about identified objects in Step 1, using predefined fact templates in an online game called SYGY^{Map}. Fact templates are prepared by the administrator.

Step 3 – Using the visual map generator to construct a visual concept map. Those labels and facts which are created in steps 1 and 2 are used to generate a visual concept map.

The main scenario for generating visual maps is started from the administrator. He/she has to choose a topic and then prepare a sample image corpus which is actually the starting point. Later he/she has to configure the game and ask volunteers (i.e. students) to play the game in order to generate labels and facts for identified objects in digital images. For example, the admin would choose “Mammals” as the topic and select a few photos of mammals from his/her own resources. When the game is finished, the admin can use the generated labels and facts to build a visual concept map. Several approaches could be taken to process and generate a visual map out of a database of annotated objects and related facts. The algorithms for generating two models of visual maps are proposed in this work. These two models are called Center-Based Maps and Similarity Maps. A Center-Based Map is supposed to express the features and attributes of annotated objects while a Similarity Map is illustrating the similarity degree between pairs of objects. By saying objects, we refer to those entities which annotators (volunteers) would find in the digital images while they are using our annotation tools.

The main two challenges that we addressed are:

1. Methods and tools for collecting knowledge about visual objects inside digital images. This knowledge would tell us the names of objects, their location inside images and their features and characteristics.
2. New algorithms to process the database of annotated objects and related facts in order to generate two sample types of visual concept maps.

Some other sub-problems that we had to tackle were:

- A. Where could we find a ready labeled image corpus?
- B. If we prefer to have our own digital image corpus, who labels them for us?
- C. What are the latest trends in manual or automatic image annotation?
- D. How could we make incentives for volunteers (for example students) to contribute and label images for us?

E. How could we analyze and extract a meaningful concept map out of database of annotated objects and related facts?

As we found that we would not have easy access to a rich database of annotated objects, we planned to develop our own applications to collect and process the information about objects found inside the digital images. Two main applications were developed. The first application is called Collimator. Collimator is an advanced online tool to label digital images. The user would load the image into the interface and then draw a bounding box around a specific object and later he/she has to assign a label to that object. The tool has certain features such as multiple image handling, importing ontologies and exporting RDF files. Collimator preceded our second application which is an online game package called SYGY. We realized that the games are great incentives for people to play and contribute. SYGY is divided into two counterpart games: SYGY^{TAG} and SYGY^{Map}. SYGY^{TAG} is used to locate objects and assign labels to them while SYGY^{Map} is used to generate facts about objects in digital images.

Figure 1-5 demonstrates the relationship among elements of our research in the form of a concept map. It provides a bigger picture of what we have covered throughout the thesis and makes it easy to understand where each piece of the puzzle is standing. Here is one sample reading of our thesis concept map from Figure 1-5:

Volunteers may play SYGY^{TAG} or use Collimator to Locate and Label Objects inside images of Image Corpus.

Figure 1-5 Relationship between main concepts of current thesis in a form of concept map

As it is seen in the introductory chapter (chapter 1), the spirit of the multidisciplinary research activity led us through all the steps of this job. It worth to

mention that we benefited from the contributions of idea developers, strategists, consultants and code developers and every piece of our research puzzle is a bold footprint of our punctilious supervisors and colleagues.

1.7 Organization of the Thesis

The thesis consists of 7 chapters, 2 appendixes and the list of references and bibliography. The content is mainly divided into two parts. The first part including chapters 2,3 and 4 explains about our research experience with methods and challenges of collecting knowledge through the web, while in the second part including chapters 5 and 6, we primarily discuss methods and algorithms that we developed and used to generate sample visual concept maps and some sample case studies. For better understanding the structure of the thesis and the material covered in each chapter, we shortly review the following chapters here:

Chapter 1 - This chapter has a concise overview on some of the latest research trends which were involved in our work. Concept maps are the foundation of our research. We also benefited from studying the power of visual learning and the methods of acquiring small pieces of knowledge from contributors. Objectives and challenges of our research work are also explained at the end of this chapter.

Chapter 2 – A big part of our research work is focused on methods and challenges of collecting commonsense knowledge. As the topic was of great value to us, we have covered it in a separate chapter. In this chapter, we go deeper into bottlenecks of knowledge acquisition and discuss resolving ambiguity and validating the collected knowledge.

Chapter 3 - Collimator is our first experience with developing an online application for annotating digital images. The application is rich in nature and has certain valuable features which have been reviewed and presented in several academic events.[24] Motivations for developing collimator, features and applications of collimator are discussed in this chapter.

Chapter 4 - Following Collimator, we studied further to make the job of image labeling fun for contributors. In recent years, more human computation tasks are routed through online games and that is why we designed and developed the SYGY game as our second research product. SYGY is divided into two counterpart games each aimed at collecting a specific type of knowledge. In this chapter, we covered a wide range of

topics related to SYGY games including the basic game mechanism, data validation, avoiding cheaters and short evaluations.

Chapter 5 - In this chapter, we move inside the idea of visual concept maps and we explain in more detail methods and algorithms for generating visual concept maps. The procedures to generate two types of visual concepts maps are explained. These two visual maps are: Similarity Maps and Center-Based Visual Maps.

Chapter 6 - We have conducted case studies to evaluate the algorithms which are proposed in chapter 5. Sample data entries, matrix of similarities and even extended Similarity Maps are included in this chapter.

Chapter 7 - A short discussion about future research opportunities and a concise conclusion is covered in this chapter.

Appendix I - More screenshots and sample game sessions from SYGY^{TAG} and SYGY^{Map} are illustrated in this appendix.

Appendix II - Our unique algorithm and strategy to generate scores in SYGY^{TAG} is explained in more details.

Chapter 2 Commonsense Knowledge Acquisition

2.1 What is a Commonsense Fact?

Commonsense, as the term itself tries to reveal, is what people in common would agree. It is a common natural understanding among humans. A big difference between computers and humans is that computers do not have a full understanding of what humans rely on to solve problems. Facts like “Boiling water is dangerous” or “Poor people cannot buy expensive things” are trifling and implicit to us but are far from computer reasoning. In reality, a commonsense fact is a true statement about the world that is known to most humans. The knowledge and experience most people have, or are believed to have is called Commonsense Knowledge and Commonsense Experience. In contrast to commonsense knowledge, we have expert knowledge which is not owned by ordinary people. For example the knowledge which a computer scientist or a physician keeps in his mind is a kind of expert knowledge. Methods for collecting expert knowledge would be different from techniques for eliciting commonsense knowledge.

2.1.1 Identifying the Commonsense Knowledge

Beyond the definition, identifying those pieces of knowledge which are common among humans is a challenging task. The real problem is that computers do not know anything about us! The knowledge that people in our society share, and the things that humans commonly understand are still ambiguous for machines. We often hardly realize that we know this common knowledge and this knowledge is one of our fundamental elements for making daily decisions in our life. Computers do not know anything about this ordinary knowledge. They actually lack common sense. They do not have any imagination about the patterns of people’s lives, what emotions we have and the hopes and fears we may keep in our heart. By giving computers ways to represent and reason about this common knowledge, they can reveal those interesting things we would not know about our own knowledge and help us to do more reasoning tasks faster and easier.

2.1.2 Motivation for Collecting Commonsense Knowledge

The motivation for collecting a large database of "true statements" is the belief that such knowledge is necessary to create truly intelligent systems. There are also more immediate applications. For example, a search engine was prototyped that converts the query "my cat is sick" to "veterinarians, Boston, MA", by following a simple chain of reasoning based on an underlying network of commonsense facts [25].

2.1.3 Previous Works

Over the past two decades, there have been several efforts devoted to collecting a large database of "commonsense" knowledge [26], [27], [28].

McCarthy's advice taker proposal in 1958 was probably the first endeavor to use logic for representing commonsense knowledge in mathematical logic.

The Cyc project is an attempt to provide a basis of commonsense knowledge for artificial intelligence systems with the goal of enabling AI applications to perform human-like reasoning.

The Open Mind Commonsense project is similar to Cyc except that it was designed as an online knowledge collector. Even non-experts can contribute to Open Mind.

2.2 Collecting from Contributors over the Web

If we prefer to collect knowledge from volunteers (experts and non-experts), we must focus on ways to motivate them to contribute high-quality knowledge [29]. This challenge is exacerbated by the fact that the number of volunteers (contributors) drops over time and this has been experienced in many academic projects before. It is believed that many large-scale open problems could be solved by channeling human brain power into the computer training area.

Most People in the world do not know anything about knowledge representation and why it is important to computer scientists. Thus eliciting knowledge from these people is challenging. Millions of people have access to the Internet and this number is growing each day. This is important to us to be able to acquire knowledge from many possible contributors over the web. Designing user interfaces that would allow a user to enter his knowledge by filling data entry forms has been practiced by the knowledge

acquisition community for several years. More advanced user interfaces would even let users use a mouse to draw a shape or point to answers on the screen. While these approaches make knowledge entry easy, they also constrain what type of knowledge can be entered and they usually force the contributor to use a particular method for representing knowledge.

To date, one of the most ambitious projects to gather commonsense knowledge from ordinary contributors is Open Mind Commonsense [30] which uses pre-defined templates and prompts for free-form text to collect knowledge.

The Open Mind initiative focuses on creating a common platform of tools for gathering from ordinary netizens, sharing the collected data, cross-validating their input and even rewarding the best contributors.

In our project, we have designed our own data entry schemes for collecting knowledge from contributors. For labeling digital images, a user must use his mouse to draw a shape (in Collimator) or choose covering cells (in SYGY^{TAG}) and enter the English label into a text box. More details about forms of data entry are explained in chapter 3 and 4.

2.3 Knowledge Acquisition

Knowledge acquisition is the transformation of knowledge from the forms in which it is available in the world into forms that can be used by a knowledge system and interpreted by machines. It is commonly said that Knowledge Acquisition is about extracting knowledge from sources of expertise. These sources are in fact those people who are expert in the specified domain of knowledge. Recently, by growing collaborative knowledge systems over the web, the knowledge acquisition is not just a job for experts but even non-experts would contribute and stay there as the source of knowledge. For example, commonsense knowledge, as we delineated earlier in this chapter, could be acquired through ordinary contributors.

2.3.1 Knowledge Engineering

Knowledge Acquisition is the first step of the knowledge engineering process [31]. Knowledge Engineering is mainly used to build expert systems which are forms of Artificial Intelligence systems. The one who is involved in knowledge engineering

process is commonly called a knowledge engineer. His task is to elicit knowledge from experts and then transfer this knowledge into the knowledge base. This knowledge base is a database of related knowledge to be used for solving domain specific problems.

The role of Knowledge Engineer is critical in the process of collecting knowledge. The knowledge engineer should have several qualifications. Some of these qualifications are:

- Having computer skills
- Proficiency in effective Communication with contributors (For example experts)
- Having a good perspective on the Organization of knowledge
- Thinking logically
- Being Self Confident
- Being patient and trying to tolerate the mistakes

These qualifications are conceivable where we talk about a knowledge engineer as a human. But in our project, we tried to simulate this knowledge engineer as computer software which means that the software acts like a knowledge engineer to collect knowledge. This software has very similar qualifications as we discussed earlier. It also holds other specific attributes since it is a machine code. We tried to specify our requirements to clarify what type of knowledge it should be able to collect. We worked on two different software programs. Both of them are explained in Chapter 3 and 4 of this thesis.

Looking back at attributes of our knowledge collector (knowledge engineer) we identified the following characteristics for our software:

1. The software must have an easily understandable interface in order to help all contributors to use it without the hassles of long time learning process.
2. The software must be accessible through the web, so everyone can be involved in any convenient time using different browsers.
3. The software should be able to reduce the mistakes and even eliminate the effect of cheaters. The cheaters are actually tainting the acquired knowledge and certain security strategies must be in place in order to cut the cheaters' access to inject inaccurate knowledge in the database.
4. The software must pave the way for contributors in order to help them to contribute quickly and reliably. Asking contributors to type long sentences, as we

have in Open Mind project [32], would kill the enthusiasm and slow down the process of knowledge donation.

5. The software must be configurable. The administrator would tune the software to make it ready for specific domain of knowledge.

2.4 Methods of Knowledge Acquisition

Acquiring knowledge would be done in three different methods: Manual, Semiautomatic and Automatic. In the manual method, the knowledge engineer (our software) through the help of administrator is responsible to design a scenario to elicit knowledge from contributors. The contributors are the main source of knowledge and the quality of knowledge base depends on their valuable contribution. In the Semiautomatic method, the knowledge engineer (our software) is responsible for adjusting the inputs in order to ask better questions while collecting knowledge. In this method, contributors would need to spend much less time to respond and the computer should find answers in a certain threshold time. For example if someone ask an address from 3 random persons in a street and all respond with a same route, then it would be possible to trust in them and there may not be any need to ask more. In the automatic method, the software automatically tries to extract knowledge from different sources of knowledge such as books, films, computer databases, maps, pictures, videos, voices and observed behaviors, keeping in mind that all of these were initially created or developed by humans. In this method, the need for contributors is eliminated and the quality of extracted knowledge depends on the quality of sources.

We have used the semiautomatic method to design our knowledge acquisition system. The administrator is responsible for tuning the software and the software acts like a knowledge engineer. The administrator needs to be aware of certain technical requirements and has to design the scenario of knowledge acquisition before running the software. The domain of knowledge and the number of contributors would be important to consider. It is also necessary to know more about the target group of contributors. If they are small children under 5 years of age, the administrator must know what type of knowledge would be common in this age range. Asking hard questions for collecting knowledge from little children would certainly fail to meet the goals of acquiring accurate knowledge. For example most of the children can comment

on the colors of fruits and even their taste and shape. But they may not know where these fruits are growing and what vitamins each fruit has.

2.4.1 Interviews

Eliciting the knowledge from humans using a semiautomatic method is usually done through three main routes: interviews, tracking the reasoning process and observing. We use interviews for eliciting knowledge since other two routes are much more complex and usually humans are involved as knowledge engineers. Interviews are just like a discussion between a knowledge engineer (our software) and the contributor. The collecting instrument is a kind of web accessible platform that contributors can use anytime anywhere while they have connection to the internet. A well-designed form of data entry (like questionnaires) is used to help contributors use their mouse and keyboard to enter their knowledge. For example they may use their mouse to locate the objects inside a digital image shown on the screen. Contributors may also need to type a label for the objects they find in a digital image or confirm/reject small pieces of knowledge (called facts) about those objects. These pieces of knowledge were automatically created by the system as a questionnaire. These forms of data entry are discussed in more detail when we talk about Collimator and SYGY games in Chapter 3 and 4.

2.4.2 Structured and Unstructured

As we mentioned earlier, the contributors are asked to respond to certain types of questions or locate objects inside photos and try to label them. This is like an interview (or conversation) between knowledge contributor and our knowledge engineer which is software. The interview must be designed in a way that both parties find a common ground for dialoguc. The contributor must share his valuable knowledge and the software must ask good questions and provide first-class tools to help the contributor to work faster and easier. There are two types of interviews between contributor and the software: Structured and Unstructured. In the unstructured method, little planning for preparing the interview is needed and the software has to try to identify attributes of the problem by asking several questions which are not focused. The data obtained from these interviews are often confusing, complex and mostly difficult for the software to

understand. The structured method which we also use, is more meaningful and is a more precise method of Knowledge Acquisition. It is known as a "systematic goal-oriented process" as it tries to approach the contributors in a systematic way. The system is defined to achieve explicit goals.

2.5 Bottleneck in Commonsense Knowledge Acquisition

In the Artificial Intelligence community, the problem of acquiring the body of knowledge is known as the knowledge acquisition bottleneck. Depending on the type of knowledge we want to acquire, different methods and tools should be used. We studied some of the bottlenecks which would hinder the knowledge acquisition process and make it unfeasible. Depending on two main types of knowledge we would want to elicit from human brains, bottlenecks are sorted out.

Hindrances for extracting Expert Knowledge are:

- Designing intelligent software which would be able to extract this kind of knowledge is challenging. Machines do not know what complexities they encounter and even they can not sort the relevancies and significances of those collected pieces of knowledge easily. Many techniques of AI should be used to design such software.
- Experts are not always free. This is an important fact. We cannot expect experts to spend their valuable time to respond to machines.
- Experts do not always volunteer to share. Sometimes there are a few experts in a specific domain of knowledge and they hold intellectual property for what they know. They may even spend most of their lifetime to learn and earn that knowledge for money and fame. They do not usually volunteer for knowledge sharing.
- Experts can even provide incomplete and incorrect knowledge. This is because they even may not know all dimensions of a problem. They may be able to answer a few questions but they may not know all the answers for knowledge collectors. Sometimes they think they know the answers, but the answer may not be correct.
- Some Experts cannot articulate their knowledge easily. The structure of knowledge in their minds is complex and they do not know how to express

what they have learned. For example, capturing knowledge from an old man who has spent 50 years of his life opening different door locks, is not easy.

Some obstacles for eliciting commonsense knowledge are:

- It is wrong to believe that all common knowledge is known to all humans living on the earth. Depending on geographical location and communities of people, the common understandings would be different. There are many facts that all people in Singapore would know but they may be ambiguous and unknown to people living in Nordic countries.
- Since common knowledge is mainly collected from ordinary contributors, the risk for lexical ambiguity is higher. Some words have different senses and some facts would be understood differently by distinct groups of people. Lexical ambiguity has been discussed in section 2.7.2.
- People believe in many different things which would be in contrast to each other. They grow up in different locations. They live in different families and they have different cultures. Religion is also another important factor. Some people believe that God exists and some believe the opposite. We have to be careful about thoughts and beliefs. These thoughts and beliefs are in common among certain groups of people but they are not necessarily universal.
- There should be some incentives for ordinary people in order to contribute. We need them to share their common knowledge but what we do in return for them? Usually ordinary people are impatient. If they feel that they do not earn or enjoy, they will leave our knowledge collector system.

2.6 Validating Commonsense Knowledge

There are several methods for validating commonsense knowledge. Most of the time, validation is based on comparison processes. The knowledge which was collected (for example through the web) must be validated against other similar assertions. In this case, each piece of knowledge created by each contributor should be evaluated to see if it has enough support or not. Those pieces of knowledge which receive a lower support than specified limits are not reliable and could be ignored for analyzing or reasoning stages. By saying support, we mean a measure that clarifies how often the collection of

items (pieces of knowledge) in an association occurs together as a percentage of all the available items. As an example if we ask 100 contributors to comment on the fact: “Boiling water is Hot” and 98 out of them say that this fact is true and two of them say it is not true, having 10% threshold for minimum acceptable support, we can conclude that the support for “Boiling water is NOT hot” is below the threshold and this fact is ignored while the fact “Boiling water is Hot” is valid. We have used minimum acceptable support for our calculations in chapter 5.

2.7 Ambiguity in Commonsense knowledge

It is not always easy to conclude whether a fact is totally true or false. Many of commonsense facts are relative. They cannot be absolutely “true” or absolutely “false”. People usually comment on facts based on perception and this perception comes from their own experiences in their life and even the effect of public comment in their mind. For example, if we ask a tourist coming from Tokyo to Singapore who would travel by MRT (City transport train in Singapore) to comment on: “MRT is FAST”, he would say this fact is not true. But if we ask from a Singaporean, he would say that it is true. Local trains in Tokyo are quite speedy and that is why Japanese people do not have the same sense of high speed with MRT in Singapore.

In contrast, there are several facts that almost all people have the same belief in them, even if they live in different locations or whether they were educated or not. Most of these facts are realities of our life on the earth which come from the physical world around us. Some of these real facts are:

- Releasing an object held by the hand will cause it fall down (Newton's law)
- Cars which use gas are polluting the air
- People need to sleep to survive
- One can see himself through the mirror

2.7.1 Number of Contributors

As we mentioned, some commonsense facts are relative and some are absolute. If we decide to use an online knowledge collector system to ask about a few commonsense facts, we should be also aware of the trigger points for reaching

consensus. We have to be careful who our dominant contributors are. We may classify contributors by using factors such as: Age, Geographical Location, Level of literacy and expertise. In addition we should not forget that the number of contributors is also an important factor. If we are keen to ask comments of people on some absolute facts, we may not need so many contributors since almost all contributors should comment the same as each other. Asking 100 or 1000 or 1 million people that “Animals eat food to survive” would provide us with the same result. But if the fact was relative then the number of contributors would be an important element. More contributors would give us a better statistical view and thus help us in having richer knowledge about each fact.

Humans are best at solving ambiguities. When they judge, they usually use hundreds or thousands of small pieces of knowledge. They acquire these small pieces of knowledge through their life by learning, experiencing and using their five basic senses. We believe using the brain power of humans to solve ambiguities for computers is important. In this way, while we design any knowledge collector to elicit commonsense knowledge, we must do our best to put up ambiguities in front of contributors to solve for us.

2.7.2 Lexical Ambiguity & Word Senses

Most of the time, collecting commonsense knowledge depends on the processing of natural languages. In our project, especially for labeling images (discussed in chapter 3 and 4), contributors need to enter their own describing labels while they see an image. The labels that users are providing, are sources of ambiguity for us. Lexical ambiguity arises when the context is insufficient to determine the sense of a single word that has more than one meaning. For example, the word “bank” has several meanings, including “financial institution” and “edge of a river”, but if someone says “I went to the bank to deposit \$100”, the intended meaning is clear. However, phrases and sentences can be ambiguous even if all of their constituents are explicit. For example, “The police shot the rioters with guns” is ambiguous since it is not clear that if police was using guns to shoot the rioters or rioters have the guns when police was shooting them. Ambiguity can have both aspects of lexical and a structural basis. In this sentence: “He saw her duck”, it is not clear that duck means a bird or duck means bending over. So if it is a bird, then we know that the lady had the duck. But if it means bending over, then he saw her when she bent over.

Although ambiguity is fundamentally a property of linguistic expressions, people are also said to be ambiguous on occasion in how they use language. Even when some people believe that they are expressing themselves clearly, it may be still hard to understand them. In our knowledge collector system, we thought about lexical and structural ambiguity. Since we are collecting knowledge as labels for annotating digital images, we have fewer problems with structural ambiguity but we still have challenges for the lexical part.

2.7.3 Models of Data Entry

In our Image annotation tool (Collimator or SYGY^{TAG}) People must assign one or two words for each object they find in an image. Through our tests, we found different models of inputs that we usually need to deal with. In those cases where resolving ambiguity needs high level techniques of language processing, we have left the problem open to other researchers. Some models of inputs are as below:

- Some words do not have any meaning to us at all. These words are usually abbreviations or even wrong entries due to misspelling. We check our own online dictionary to see if the word is valid and within the English dictionary or not. If we find no match, we mark it as an incorrect label in our database. They are like noises which should be filtered.
- Some labels are just adjectives or Adverbs. For example Red, Big, Long and Heavy. In these cases, we usually mark the word as an adjective or adverb in our database. We usually encourage contributors not to use adjectives alone since adjectives are modifiers and the noun after the adjective is more meaningful to us while recording a label.
- Some labels are in form of adjective + noun. In these cases, we use an advanced word processing tool called POSTagger [33] to drag out the noun part. POSTagger is a natural language processing tool capable of analyzing text for lexical categories of words. Using POSTagger nominal labels can be distinguished from other forms.

Chapter 3 Developed Tools – Collimator

3.1 Motivations

Collimator is a web application which was designed and implemented mainly for manual image annotation. This application demonstrates a new environment for collaborative image tagging which helps users to create, confirm or reject any inferential concept or visual object which exists inside a selected image. We studied knowledge representation methods and several conceptual theories before designing this tool. Collimator is based on ontological thinking. Our main purposes to design Collimator were:

- Getting experience on how to design an online image annotation application.
- Getting ordinary volunteers on board to participate in image annotation process.
- Being a handy and tiny fast online application for collecting commonsense knowledge.
- Preparing and having access to a database of annotated images

The tool provides a simple interface for user inputs and also assists in building up an enhanced collaborative tagging environment. The tool will allow users to select visual areas (called regions) in an image and then generate a label (tag) for that region. These labels are basically made for Visual Objects (A kind of object which is observable within a digital image and the user can point it out without any ambiguity). The second tagging method is inferential tagging which comes from the inferring capability of the human mind. A sample inferential tag is the concept of autumn in a digital image which portrays an autumn scene. No one can select any object inside an image and say “*This is autumn*”. This comes from inferential capability of the humans. When some sees bended trees and yellow leaves on the ground, he/she can infer that this is an autumn scene. There are two main scenarios developed in Collimator: Tagging and Mapping. The Tagging scenario allows the user to tag the digital images (locating and labeling the objects in the images) and the Mapping scenarios is meant for making facts (propositions) about two objects in one image. We use the terminology of “Mapping” since the user will use a linking phrase to map two visual objects to each other in order to create a new fact (proposition) about those two objects.

The core of the application gets ordinary users involved in semantically annotating digital images. According to our studies, most annotating applications are developed based on academic concepts, and ordinary users are unable to understand the main purpose of those applications. Even for some experts, it takes time to find out how to use the tool. For example, ordinary users are not familiar with Ontologies. They are usually unfamiliar with the Semantic Web and the meaning of Resource description. How can we expect them to start importing Ontologies and make instances or assign properties? Even some of the academic annotating tools are difficult to use and users need to review a long tutorial to understand how the application works and why it works like that. We also analyzed stand alone applications and compared them with Web-based applications. Most of stand alone applications are written in Java and usually they need installed JVM on the machines and sometimes user's experience for running Java applications. There are cases where even administrator permission is a must.

Collimator is a web-based image annotation tool which has been designed based on the SVG standard and the AJAX technology. These two are explained at 3.4.1 and 3.4.2. Collimator makes no barrier for any user as long as the user is using a SVG/Java Script enabled browser. In this case, the user does not need to download any application to install it on his/her machine. This is good especially for students at universities or people who have limits on hard disk usage and writing permissions.

Our main focus was on developing an enhanced collaborative environment which did not have complexities of other annotating tools. Through such an environment, all categories of users can enjoy tagging images in their community. Even the tagging job would turn into a learning experience. Sometimes users will learn from other tags which have been generated by other groups of users. It also has benefits for children and young people who can work on sample school images and even run up a voting system over several concepts available in a single image. This shows great potential in Collimator.

We have used the Collimator database to implement a useful web service for Partial Image Extraction which is explained at section 3.8. In 3.4.1 and 3.4.2 we will review two basic technologies which have been used for implementing Collimator: SVG and AJAX. We will also discuss some of the selected features of Collimator in more detail.

3.2 Versioning of Collimator

Development of Collimator was a long journey. Since we earned valuable experiences during the evolution of Collimator, we decided to include them in this section of the report.

The first version of Collimator was created in January 2006, when we had the idea of building visual maps. These maps were actually exposing the combination of a few photos and their relations. We searched for different information resources to see if we could find any publicly accessible image databases. This public database had to provide us with some semantic metadata for searching images based on concepts available within each image, since a region inside an image can depict a concept. We found that there exists no such database which could give us access to meaningful information for different regions within an image. So we decided to design our own laboratory which would help us to have a database of image annotations. Later, we found that we could follow up on semantic web techniques while we build our laboratory for generating visual maps. When we started to think about functions of Collimator, we found that it could be turned into a powerful online image annotator in the future. So we started to learn Semantic Web technologies and compare different image annotation tools for the semantic web. In our survey, we found that most of these tools are big in size and hard to understand for ordinary users. The creators of such tools were just trying to show how current W3C standards like RDF and OWL could be used to annotate an image, but our aim was a bit different. We wanted large groups of users involved in applications to collaborate, so we had to make the user interface as easy as possible. Users' contributions were critical for us. Meanwhile we had the opportunity to verify different semantic web standards to be aware of the latest progress in the research sector and include those standards in our newer versions.

Three versions of Collimator were designed since January 2006. The first version was very simple and its focus was on the way we can get users involved in annotating regions. In the first version, we had two main concepts: Visual Concepts and Virtual Concepts. Later we changed Visual Concept to Visual Object and Virtual Concept to Inferential Tag in the second version.

The primitive design of Collimator in version 1.0 is depicted in Figure 3-1.

URL: http://rojina.com/images/fruits.jpg

Your Name: Alireza Kashian **ID:** ALIR0001

Visual Concept	Virtual Concept
Available Concept: <input type="text"/> Bowl : 1003 <input type="button" value="Confirm"/> <input type="button" value="Edit"/> Creator: ALIR0001 Citation: 18 Certainty level: sure Reference: http://wordnet.princeton.edu/perl/webwn?s=bowl Description: A round vessel that is open at the top	
New Concept: Table Reference: http://wordnet.princeton.edu/perl/webwn?s=table Description: Furniture supported by one or more vertical legs Certainty Level: Sure <input type="button" value="Submit"/>	

Visual Tools:

Select Drawing Tool: ☐ ☒ ☐ ☐

Visual Search: apple

Note 1: Each Pixel in an image could represent a concept.
Note 2: Many concepts in an image could have same names.

Concept Mapping

Concept 1: Grape Relation: is inside Concept 2: Bowl

Generated Fact: {Grape : 1001} ==> is inside ==> {Bowl : 1003}

Figure 3-1: Primitive design of Collimator (Version 1)

For the second version of Collimator we studied different techniques to implement the application and chose AJAX and SVG as our base technologies. SVG is quite accepted in academic communities since it is an open xml based standard and has become a W3C recommendation in 2003. AJAX is also new and in recent years more applications are migrating to use AJAX. Figure 3-2 is the screenshot of the 2nd version of Collimator.

We totally changed our implementation technology in the 3rd version of Collimator. Several features such as importing ontologies and even exporting RDF documents were added. The user interface was completely changed. According to our experience with the 2nd version, heavy SVG components had made Collimator slow and there were so many bugs within the SVG libraries which we had used to generate windows and text boxes. In the 2nd version, everything was SVG. We found if we were to continue on a full SVG interface, the project progress would be delayed due to the large amount of tasks for identifying bugs. Finally we decided to put up a new design and use Google Web Toolkit (GWT) [34] for AJAX messaging in the 3rd version.

Our 2nd version was not fully compatible with the latest Adobe SVG Viewer for Internet Explorer 6 and AJAX was also not able to exchange messages with the server. In the 3rd version, we could solve these bugs. We changed the interface to SVG and Non-SVG areas. The application got faster and text handling was not in graphical mode anymore. A sample screenshot of the 3rd version is shown in Figure 3-3.

Chapter 3 Developed Tools – Collimator

Figure 3-2: Screenshot of the 2nd version of Collimator

Figure 3-3: Screenshot of the 3rd version of Collimator

3.3 Collimator Terminology and Access Point

The primary name of Collimator was Collaborative Concept Tagging (CCT) [35], which was changed to Collimator later in June 2006. Collimator is the combination of **Collaborative Image Annotator** words. We believe that Collimator is analogous to a photonic X-ray imaging device which is called Collimator. This device filters a stream of rays so that only those traveling parallel to a specified direction are allowed through [36] In the same way, our application sieves and filters the tags generated by a group of users based on users' consensus. The product of our Collimator is clean and contains concentrated tags which can be used in other semantic web applications. We have not limited the users with templates and schemas. The free format input, will simulate the different directed photons as we have in X-ray imaging. We must focus on the methods of converging diverse number of tags.

We registered a new domain name for promoting the Collimator project [24] and hosted it on a Linux server. The domain is: www.xmlweb.info, and the latest version of the project is located at Collimator project [37].

3.4 Implementation and Architecture

The main two technologies used for developing Collimator are SVG and AJAX. We will shortly review them in the following section.

3.4.1 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a kind of two dimensional graphic file format and at the same time, a web development language based on XML. Developers and designers can use SVG to build dynamic and high quality graphics using real-time data with precise visual control.

SVG has two parts: an XML-based file format and a programming API for graphical applications. One can add shapes, texts and embedded raster graphics into a SVG file and use different painting styles. SVG supports animation which means it is possible to move, resize and change colors in an animation on the web page.

SVG 1.1 and SVG Mobile Profiles are Web standards (W3C Recommendations). Some groups in W3C are still working on SVG 1.2 and future profiles for Mobile and Printing. SVG 1.1 became a W3C recommendation in January 2003.

SVG Applications: SVG is widely used in Web graphic design, animation, user interfaces, graphics interchange, print output and mobile applications. (In our case, we have used SVG to design our user interface to manage graphical interactions with the users)

Since SVG is an XML vocabulary, it can be linked to other applications and can use different sources of information to process and generate new graphics. Since SVG is an open source standard, everyone can edit SVG files.

SVG Advantages:

- SVG images are scalable
- SVG is an open standard, works with JAVA technology
- SVG files are pure XML and can be read and edited by a large range of editors (like notepad)
- SVG files are smaller in size and compressible comparing to JPEG, GIF
- SVG Images can be printed in any resolution

To use SVG, SVG Viewer should be installed on the Internet browser. Internet Explorer 7 and Firefox 1.5+ have built-in support for SVG, but for other browsers, a plug-in should be installed for the browser. Adobe is one of the companies who have developed the plug-in to be used with different versions of browsers.

Below is the sample code to draw an ellipse and a rectangle:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1" xmlns="http://www.w3.org/2000/svg">
<ellipse cx="300" cy="150" rx="200" ry="80" style="fill:rgb(200,100,50); stroke:rgb(0,0,100);
stroke-width:2"/>
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:1; stroke:rgb(0,0,0)"/>
</svg>
```

The result on screen is like Figure 3-4.

Figure 3-4: Output of an example SVG Code on the screen

3.4.2 AJAX for Data Communication

AJAX stands for Asynchronous JavaScript and XML. AJAX is an outstanding approach that helps us transform heavy Web interfaces into interactive AJAX applications. One can use AJAX to build interactive Web Applications. AJAX will be the next step towards the Service Oriented Architecture revolution. AJAX is based on Open Standards which are JavaScript, XML, HTML, CSS, and DOM.

Some people are talking about AJAX as a new technology, but AJAX is not a new technology. It uses different standards and technologies to build a new method to exchange messages between users' browsers and web applications on the server.

Google Maps was a successful AJAX-based application compared to traditional competitors such as MapQuest. It shows that providing a better user experience will result in success.

Benefits of AJAX: Some benefits of AJAX are measurable and some are difficult to measure precisely.

Measurable benefits include:

1. Time spent for data transmission: Usually web applications need to transfer some bytes to the client side. This will take time according to available bandwidth and user internet connection type. AJAX will help to decrease the amount of information to be transmitted. This will lead to less time needed for a task to be completed.
2. Time spent for completing a task: A good AJAX interface will increase users' efficiency to manage their web application. Sometimes users need several iterative steps while working with a web application so they need to submit information and wait to see results and make decisions again.
3. Bandwidth consumed for entire task: AJAX needs less byte to transfer, so logically it needs less bandwidth. It will become important when a server receives many web requests and it has limited bandwidth.

Some other benefits which are not measurable:

1. Less steps to complete a task (in client side)
2. Efficient and Interactive user interface

3. Improved application responsiveness

Comparing classic web applications with AJAX web applications, one can identify a broad range of differences. Instead of loading a web page as in the classic model (at the start of a session), an AJAX engine which is written in JavaScript, will be loaded. This engine allows the user's interaction with the application to happen asynchronously. In short, AJAX applications will give an end to the start-stop-start-stop nature of classical web applications. To clarify how AJAX works, a comparison between the AJAX architecture and the traditional architecture is depicted in Figure 3-5. [38]

Figure 3-5: Traditional web application model compared to AJAX model

3.4.3 System Architecture

Collimator has been designed using AJAX, SVG and Java Servlets. The user interface (in the 3rd version) is a combination of SVG graphics and JavaScript menus. SVG graphics could be shown on any browser which supports SVG.

The application is event driven. For each event we have defined several tasks to be done. For example, when a user loads an image, we record that Image's properties (URL and resolution) in our database. The URL of the image acts as a unique identifier for our application. Each time a user creates a new region, we send all needed information to the server side. This is done through AJAX messaging so the browser

does not need to reload each time a user submits a new entry. AJAX technology plus SVG is an excellent combination for developing online applications.

The application has a very simple layering system (compared to the OSI model in networking). In both client-side and server-side we have similar peers which are aimed to talk to each other at both sides. The XMLHttpRequest layer on the client side talks to the RPC handler on the server side. Both data object handlers are working together and user interface controller at the client side gets service from session managers at server side. The lowest layer is responsible for transferring messages from server to client and vice versa. Figure 3-6 depicts this layering system for more clarification. The servlet package is connected to hibernate and the MySQL database handler at the server side.

The core logic of the application, including procedures and repositories, is shown in Figure 3-7. We use the Jena library to handle Ontology and parsing RDF documents. We have designed an engine in the heart of the application which handles most of the requests for database queries. This engine serves other applications such as concept map generator and RDF generator. There is also a web service for retrieving images based on provided keywords. Different applications can use this web service to search for images and even find some regions within a collection of images which are related to their searched keyword.

Figure 3-6: AJAX Technology and the Layering System of Collimator

Figure 3-7: Core of Collimator

As seen in Figure 3-7, the Working Image module handles all tagging issues, including inserting tags in Tags database or mapping two annotated regions to each other. This module includes all interactions with the users since it handles the user interface. Other modules like the Web services module or the concept map generator are external modules which Collimator does not depend on. Using the Jena library, we can import any ontology to our application. An Ontology Model is the module which handles all ontology related tasks like making instances and viewing class hierarchy of imported Ontology.

Google Web Toolkit

In our third version of Collimator, we started to use Google Web Toolkit (GWT) [34] which is provided by Google and helps to manage AJAX applications easier. GWT is a Java development framework that helps eliminate the difficulties of writing AJAX applications. Usually AJAX coding involves a matrix of technologies, so developers may mix up and make several mistakes while coding. Some important features of GWT are as below [34]:

Dynamic, reusable UI components: We can create a Widget by compositing other Widgets.

Really simple RPC: It is used to communicate from Collimator client side to our server side.

Browser history management: It lets us make Collimator more usable by easily adding state to the browser's back button history.

Real debugging: In production, the code is compiled to JavaScript, but at development time it runs in the Java virtual machine.

Browser compatible: GWT applications automatically support IE, Firefox, Mozilla, Safari, and Opera.

Internationalization: It supports different languages and all character encodings.

3.5 Collimator Features

According to our experience with other annotation tools and based on our idea for developing an interactive collaborative environment, we prepared a list of significant features of Collimator which are separated into three categories:

- 1- User interface features
- 2- Semantic Web features
- 3- General Capabilities

3.5.1 User Interface Features

1. Easy to understand environment with minimal need of learning.

Usually users need to read tutorials and guide documents to understand how an application works. Since ordinary users are unable to comprehend complicated concepts such as importing ontologies, we decided to take the easiness of Collimator into consideration. We need users' contribution, so we have to prepare a simple interface for them.

2. Easy to use drawing tools such as ellipse, rectangle and arrow

We have used very simple drawing tools like ellipse and rectangle. There are 4 groups of drawing tools:

1- *Rectangles*: to draw regions in rectangular mode. If the chosen object is best fitted in a rectangle, it would be better to use this tool. 2- *Ellipse*: is used to draw a region which is best fitted in an ellipse. 3- *Arrow*: is used for those objects which are mixed with

other objects and one cannot usually separate them by choosing a region, like water in glass. If a user selects the water with a rectangle, another user may think that the glass is the selected object. So an arrow will point to the water in the glass and the other arrow will point to the glass itself. 4- *Point*: this tool is used for small objects like stars in the sky. It even can point to one pixel within an image.

3. Loads multiple images in a single framework and session

User can load multiple images in a single session. We implemented this feature since we had the idea of cross-image mapping. In cross-image mapping, users can select a region in one image as the subject and another region in the 2nd image as the object. This way, two regions in two images will be connected to each other. Loading multiple images also has other benefits. For example, it is possible to jump from one image to another image without closing them.

4. Distributes the job of annotation among users in a fashionable web-based interface.

The application is designed in Web-based mode. So anyone having access to the internet with any kind of browser can start using Collimator without any installation or knowledge of compilation. Most of the annotation tools are stand alone JAVA applications where users download packages, unzip and run them. These tools are usually slow and need JVM installed. Since Collimator is web based, we decided to make the interface familiar for them and let them use it whenever and wherever they want.

5. The user does not need to save any document. The application will record each event in the database with AJAX real-time messaging.

In standalone applications, when a user makes changes or tags any region, he has to save his job each time. The tags will be saved on his machine. But in Collimator, each tag will be recorded on the server as soon as user creates it. So in real-time mode, when a user creates a tag, other users can see that tag right after elapsing a few seconds. This is interesting since users will realize how they are collaborating for annotating images. We use AJAX messaging techniques to send tag information to the server and load information to the client side.

3.5.2 Semantic Web Features

1. Exports the annotated data into RDF

RDF stands for Resource Description Framework and is a language for describing resources. RDF is W3C recommendation to be used for the semantic web. RDF files could be read and parsed through other semantic applications. Collimator can export annotated information to RDF files which can be used later in other semantic web applications. This feature will make Collimator interoperable with other semantic web applications.

2. Imports web accessible ontology

Ontologies are widely used for semantic web applications. They are accessible in different formats such as OWL (Web Ontology Language). If the user imports an ontology, he can make instances of classes and assign regions to those instances. By using ontologies, we have paved the way for ontological reasoning. For example one may have an ontology for mammals and in that ontology; some attributes of mammals are described. If there is an image which shows an elephant, it would be a good idea to use the mammal ontology and create an instance of mammals. So later if someone search for an animal which has warm blood, he/she may find this image since there exists a type of mammal in the image. This feature of Collimator is not developed since it has been out of our project scope. We are using JENA library for exporting RDF and importing ontologies.

3. Has the potential for development of Semantic web services for image retrieval

Semantic search for images is among research trends which search engine providers are interested in. Since images are annotated manually in Collimator, we have the potential to search for images not only based on keywords but also through ontological reasoning. In Collimator, regions are also mapped to each other which help us to search for relations. Collimator has not been widely used and we still need to have more annotated images in our database before we can work on different methods of semantic image retrieval.

3.5.3 General Capabilities

1. Enables ranking of each tag based on community consensus

Since users are collaborating on annotating the images, they can confirm/reject any tag or even report abuses. This democratic way of annotating images will let us assign a weight to each tag (each region which has been annotated). If more users confirm a tag,

it gains more weight (or rank). As it is seen in Figure 3-2 (in the first section of this chapter), 100 users have collaborated to annotate a region which depicts a monkey doll. 48 out of 100 have chosen Monkey as the label of that region, while 21 have chosen “Animal” which is more general than Monkey. It is also possible to see what other labels have been proposed. (Ronny, Pygmy and Simian are names of different kinds of monkeys). As seen, “Monkey” has got the highest weight since we have community consensus on this label.

2. Scans and filters out dirty data produced by abusers

Collimator has some preliminary methods to scan and filter out dirty data. Some abusers would make nonsense labels for regions. According to user’s abuse report, we usually decide to remove nonsense labels. We also find all the tags which have been created by abusers in our blacklist and we remove all the tags they have created.

3. Bans abusers based on their IP addresses and recorded cookies using PC feature extraction technology

We use cookies to record users’ latest activities with Collimator. We also record their IP addresses and try to map their login IP addresses to their user name. So if they try to make different user names and login to our system, we can find it out by checking their IP address. We also record their PC features like their Operating System type (Windows or Linux, etc), screen resolution and their browser’s version. We use this information to track abusers since we know what kind of computer they have when they use Collimator.

4. Ranks each user based on his activities

We try to assign a rank to each user based on his activities and loyalty. More useful tags will bring higher rank. At the moment the Trust Level for all users is 100% when they sign up. But in the future we plan to assign the Trust Levels after tracking a few activities like tagging some regions.

5. Users can find out if their image is already annotated or not (URL of images are unique identifiers in our system)

We do not collect images and we do not host images on our server. Users can not upload any images, and the only thing we keep in our database is the URL of the image. This means each image has a unique identifier and that is its URL. Since we have the URLs of all annotated images in our database, we can find out if the new URL request is among our annotated images or if this is a new one.

6. Capable of recognizing overlapping regions, preventing the creation of too many overlapped regions.

Sometimes a region which has been created by a user may be loaded by other users who try to create a very similar region. The application has a built-in procedure to check if the regions are overlapping or not. At the moment we have set the threshold on 70%. It means that if the new region covers 70% of the old region and vice versa, then we give an alert to the user that a very similar region is available. We ask the user if he really wants to create a new region or if he wants to choose an old region. The user can also review the tag information assigned to the old region and if he thinks that the region is almost the one which he wants to create, he selects that region and no new region will be created. We designed this overlapping detection method based on our experience with the older version which was not equipped with overlapping detection feature. Sometimes two or three overlapping regions would make users confused and the image would become so noisy and full of unnecessary drawings that they are not of much help. This overlapping feature is depicted in Figure 3-8. As it is shown, after reselecting the bottle with a new rectangle, an alert window appears at the right side, which shows all proposed Names (labels) and Descriptions.

Figure 3-8: Overlapping message for two regions

7. Creates a unique 128-Bit Address for identifying each region in each image

We assign a 128-Bit Address to each region first created. This 128-Bit address is a unique identifier for those regions which are similar to IPv6 system. This way, we can identify billions of regions and retrieve any region in any image. Partial Image Extraction is one of the applications of such an address.

8. Produces statistics about each image and each user.

Collimator has a section for statistics. It generates statistics for Images and Users. For example it is possible to find out how many users have collaborated to annotate Image X or how many tags have been created for Image Y. User’s statistics will show us how many Images the user has tagged and some other useful information about the user’s activities.

9. Searching concepts and relations among annotated images

One can search among all Tags and Relations based on keywords. The search will go through labels and descriptions of the regions in all images. It also searches the relations between two regions. For example if there is an image showing a child playing football, then we have “PLAYS” as the relation between “Child” region and “Football” tag (which is an inferential tag). The fact is like: Child -> Plays -> Football (“Child” is the subject and “Football” is the object).

Note: Each Multimedia annotation tool would have its own characteristics regarding its type of content, type of metadata and annotation level (free text input).

Table 3-1 demonstrates attributes of Collimator through measurements of the W3C working draft for the image annotation on the semantic web [24].

Table 3-1: Characteristics of Collimator	
Type of content	Images
Type of metadata	Descriptive
Format of metadata	RDF
Annotation level	Free
Operation mode	Web Based
License conditions	Open source
Collaborative or Individual	Collaborative
Granularity	Segment based
Threaded or unthreaded	Threaded
Access control	Yes

3.6 How to Use Collimator

Ordinary users can work with Collimator easily. Here we try to cover a very simple tagging scenario.

As we discussed earlier, Collimator is a web-based application which can be loaded in most well-known browsers. It is better to use the latest version of browsers which have built-in SVG viewers like Internet Explorer 7 or Firefox 1.5. In the case that the browser does not support SVG, the user can download SVG Viewer plug-in from plug-in providers such as Adobe SVG viewer [39].

Each user needs to have a User ID to enter into the working area. This User ID is easily acquired by simply registering at the index page located at <http://xmlweb.info/Collimator>. We do not have any complicated registration process. The email address will be used as the identifier of users.

Since the SVG is a scalable graphic, depending on browsers, the user can change the size of the drawing template and zoom on any part of the image (in Internet Explorer, the zooming function is performed through right click of mouse).

A simple startup scenario has the following steps (assuming that the image user loads is not annotated before and this is the first time it is loaded to Collimator):

1. Load application into the browser and use user ID to login.
2. Click on the open image button and enter the Image URL (the URL should be a public web accessible digital image in JPEG, GIF or PNG format)
3. Check all parts of the image and try to find out all interesting objects which can be annotated.
4. Select a drawing tool from the toolbar and start drawing a region (for example a rectangle)
5. A new window will pop up and ask for three text entries which are: Name (we may also call it label), Description (some more details about the selected region) and Reference (links to other web resources related to this region). User should fill all these boxes and click on the save button.
6. Continue creating other regions as described in steps 4 and 5
7. Click on sign out to close the application.

Figure 3-9 illustrates the annotation procedure explained before.

map two regions, he should select Mapping option in the menu. A pop up window will appear automatically which guides him to choose the Subject and the Object, and finally asks him to enter his proposed relation.

3.7 Form of Data Entry

Collimator is collecting data in two basic methods: A- Locating objects in digital images using mouse B- Entering free text inputs using keyboard. Both methods are based on interviews with the user as we discussed in section 2.4.1. The user will locate an object by drawing a bounding shape around the object. There are three key information elements which the user should provide when he/she wants to tag an object. These three elements are Name (or Label), Description and Reference. We do not have any predefined template for user data entry. There is no limit for a text entry. The only thing we expect is the validity and accuracy of the data which is entered by the user. The Name is quite important since it should identify the Concept in two or three words. Description would contain several sentences and should describe the Tag. Usually users will mix up the description with actions. For example we have a picture of a boy who is watching a movie at home. The user will select the boy's face and name it "Boy" or if he knows his name he would say "John". In Description he starts to write such a sentence: "He is watching a movie at home while his parents are out." This is not the description for the Boy. This is a kind of Fact which a user should make through mapping different tags. In the description field he has to provide such a sentence: "A young male of human kind". The Reference is much clearer. The reference should be any URI which links to some resource that explains more about the label, for example an article explaining what a "Boy" is or a definition in an online dictionary.

3.8 Applications

During development of Collimator we received several feedbacks from users about adding extensions. We decided to add new modules to our base application to make it clear that Collimator is not just an annotation tool. We are still receiving ideas and we

have planned to improve the application in several other dimensions. Two main applications are Concept Map Generator and a web service for Partial Image Extraction.

Concept Map Generator

This application has been one of our main incentives to design and implement Collimator. We needed to have access to a database of annotated images and Collimator made it possible for us. So the first intention was to have something which could annotate images and provide an easy and understandable interface. The Concept Map Generator converts a verbal map to a new visual map showing how different objects inside different photos could be related to each other. To do so, the Concept Map Generator searches in the tags' database and uses especial algorithms to find the most relevant images (or regions) to verbal topics in the verbal map. When we replace each node of the verbal map with its visual depiction, what we get is a kind of visual map which exposes wide range of photos and the relations between them. Visual concept maps are explained in chapter 5.

Web service for Partial Image Extraction:

Since each drawn region in each image, has a unique 128-Bit address (as discussed in section 3.2), we designed a very simple web service which is able to receive that 128 Bit address as a request and in response, sends back that region's content. For example if someone select his/her friend's face with an ellipse tool (drawing tool), he/she will receive a unique 128-Bit address which could be passed to others. The receivers can use this web service to retrieve the face without knowing exactly where the photo is located on the web (they do not know the URL of the image). The web service is supporting REST requests. (Like: <http://xmlweb.info/PI?FFFF:AB09:AAC0:45B2:33D4:0202:BBC8:99AF>).

Partial Image extraction has many advantages including Privacy improvement for image sharing over the web. Sometimes it is not intended to share the photo URL with the family or classmates, but one can use Collimator web service to share a selected region with other users (like the face in a photo of a party night).

3.9 Possible Extension Areas

A few other enhancements could be done over Collimator in future:

- ▶ Making Collimator interoperable with some other annotation tools like Amaya[40]
- ▶ Building automatic connection with WordNet [41] Ontology for lexicon processing.
- ▶ Performing some Naive Content Based Image Retrieval methods to see if two images located in two different URLs are the same or not. For example they could be exactly the same or have scaled proportions. This feature will prevent users from annotating the same image which has been annotated by a different group of users in another URL. (This happens since some famous images are uploaded to different servers and we may find them in various websites).
- ▶ Developing the reasoning part further. The reasoning part could be used in searching semantic concepts or even for learning purposes. A sample question which Collimator will be able to respond in future is: Show me something which is animal, has four legs and a human can ride it. The answer would be probably a collection of horse or elephant regions in various photos. Since “horse region” or “elephant region” in those images may be mapped to other regions, we can infer some more knowledge related to horse and elephant. Sometimes these relations and mappings are much more helpful than the image itself.
- ▶ Adding extensions to make cross-image mappings possible. At the time of writing this thesis, Collimator can just map two regions within a single image. Mapping two regions in two images would be much more interesting since gradually we would be able to find relationships between images. This is much like a network of concepts spread over the web and depicted in digital images.

3.10 Related Work

One of the applications which have given us the inspiration for designing the Collimator is Photostuff [42]. This Java based application is the work of researchers at MIND Lab at the University of Maryland.

PhotoStuff [42] is a platform independent (written in Java), image annotation tool which uses an ontology to provide the expressiveness required to assert the contents of an image, as well information about the image (date created, etc.). PhotoStuff allows users to annotate regions of an image with respect to concepts in any ontology specified in RDFS or OWL. It provides the functionality to import images (and their embedded metadata), ontologies, instance-bases, perform markup, and export the resulting annotations to disk or a Semantic Web portal. PhotoStuff is designed to load multiple ontologies at once, enabling a user to markup images with concepts distributed across any of the loaded ontologies. Using a variety of region drawing tools, users are able to highlight regions around portions of images (from Web and/or local disk), loaded in PhotoStuff. Classes from loaded ontologies can be dragged into any region, or into the image itself, creating a new instance of the selected class.

PhotoStuff also takes advantage of existing metadata embedded in image files by extracting and encoding such information in RDF/XML. This allows embedded metadata to be directly incorporated into the framework presented here and the Semantic Web in general.

Chapter 4 Games for Collecting Commonsense Knowledge

As we discussed earlier in Chapter 2, there are several techniques for collecting commonsense knowledge. Collimator was our first approach to achieve this goal but as we were expecting more online volunteers to use Collimator, we realized that intricate interfaces and complex regulations discourage contributors. We started to think how we could make the job fun for people and we finally concluded that having collimator in a form of an online game could be our best direction. Thousands of online games are created and played each day. The estimated number of human-hours spent playing solitaire around the world in one year: *billions*. We claim that games are one of the best methods to pave the way for solving open computational problems. Games are using the human's brain power in which its extraordinary power is proven. We need to be careful about designing a game. The game should be enjoyable while at the same time it must guarantee that playing the game appropriately solves an instance of the problem. Instead of using a silicon processor, games run on a processor comprised of ordinary humans interacting with computers over the Internet.

Since we are aimed at generating visual concept maps, we were concerned about two main elements in the maps which are Nodes and Links. According to our proposal (discussed in chapter 5), Nodes have to be visual depictions of concepts while Links are some facts about each pair of Nodes. Two main types of information which we require to collect through our games are: Visual depiction of concepts (we call them Visual Objects) and Facts in the form of verbal statements about those concepts. We have designed two counterpart online games to reach our goal. They are SYGY^{TAG} and SYGY^{Map}. SYGY is short for “Syncretic Synergy”. SYGY^{TAG} in nutshell is an interactive system for collaborative image labeling while SYGY^{Map} is used for collecting small pieces of knowledge about each label which was created in SYGY^{TAG}. In fact SYGY^{Map} is a complementary game for SYGY^{TAG}.

There are two concepts which are needed to be kept in mind. Knowing these two concepts is critical when we review both games in coming sections. These concepts are *Game Conductor* and *Game Player*. The game player is a person who is supposed to play the game. In our project, we assume that players are learners (like little children in

a kindergarten or students at school). The game Conductor is the person who is in charge of administrating a game. This person could be a teacher or an advisor in a class who knows how to configure SYGY to be played during the class time. A game consists of several digital images which the game conductor has to collect from his/her own resources. For example Game Conductor may use photos of animals. Game conductor needs to upload these photos to the server and then set a few configurations such as number of players, maximum time needed to finish a game round and etc. Game configurations and responsibilities of game conductor are discussed in more detail in Section 5.5.

The games were implemented in two phases and we used our broad experience with Collimator to define user interface strategies. Everyone having access to internet can load the game interface into his/her browser and it does not need any special broadband connection or browser configuration. The games are easy to play and all group of ages who can work with computers and are familiar with online games, will easily learn to play in less than a few minutes. General descriptions of games and related substances such as scoring and data evaluation are discussed in succeeding sections.

It is worthwhile to mention that some parts of this chapter and appendix II are adopted from our publications at 3rd World Congress of Computer Science [43] and ITZone at Nanyang Technological University [44].

4.1 Design of a Useful Game

Over the last 30 years, human computation researchers have documented the significance of placing pleasure and fun in a user interface [45], [46], [40]. Systems like StyleCam attempt to use game-like interfaces to amplify enjoyment and engagement with the software. Many other researchers have also stressed the power of game-like interfaces to increase user commitment with the systems and make the job fun for them [46], [40]. The idea of turning work tasks into games is often applied in educational domains for children's learning activities. There must be a tight interplay between the game interaction and the work to be accomplished.

From Wikipedia [47], we may define a game as “a recreational activity involving one or more players. This can be defined by (1) a goal that the players try to reach, and

(2) some set of rules that determine what the players can or can not do. Games are played for entertainment or enjoyment.”

According to Luis von Ahn [23], games for collecting knowledge are like human algorithms. A human algorithm game is thus composed of a computational problem requiring certain input and producing an output. The type of computational problems which we are interested in are those which are easy for humans to solve but hard for computers to understand and analyze. For example in SYGY^{TAG}, the player can easily look at a digital image and locate an object inside it even if the object is a bit dark and blurred. But computers are far away from being able to differentiate thousands of objects in digital images where each object would have different shapes, colors and visual qualities. As it is observable, human cycles are typically much slower and more expensive than computer cycles and that is why when we are determined to use them, we have to be sure that we are effectively using it to solve nontrivial problems which computers cannot unravel easily.

To design a valuable game which utilizes the human’s brain power we have to be sure about several factors which define effectiveness of a human algorithm game. These elements are described below:

- **Correctness:** It is important to note that in a standard algorithm the relation between input and output should be verifiably correct. Even if the players wish to provide wrong inputs, we have to have certain mechanisms to guarantee the correctness of output with a strong probability.
- **Problem identification:** One of the primary steps for developing a human algorithm game is identifying whether solving the problem through humans is valuable or not. A good problem has two attributes: it must be resolvable by humans and it should be quite hard for computers to unravel.
- **Human Interaction:** This is one of the most important aspects of designing a game. The interaction between humans and the game should provide us a guaranteed path from input to output. There are not always one route for defining interactions between humans and computers, but designers should be aware that an enjoyable engagement which brings challenge to a game is quite valuable.

4.2 Fairness Analysis

One important aspect of games is being fair to players. All have to be treated the same with equal chances for winning. A simple logic says that a strong player earns more points compared to others. But if we include the factor of chance, a player may not earn as many points as he deserves. For example in pairing games such as Google image labeler [48], players may complain they are not satisfied with their partner since he/she is poor in English or is not a fast typist or is too young to understand the image, etc.

Sample elements which could define the chance to get a better score are:

- a. Internet connection speed (broadband or dialup)
- b. Partner's skill for paired player games.
- c. The complexity of the input (for example a vague image)

That is not to say that playing games teaches us to play fair (although in many cases, of course, it does do that), but that through games - because of their abstract, "trivial," non-political, non-threatening, non-permanent status- we can come closer than through any other human activity to grasping the meaning of "fairness". Simply stated, games help us understand what our society means by fairness [49].

4.3 SYGY^{TAG}

As we mentioned before, our goal is to have a visual depiction for concepts in order to construct our visual concept maps. There are several ways for finding a visual depiction for a specific topic. For example using a search engine would help us find thousands of images about an object like a Tree. But many of the found images are not exactly related to Tree since search engines are using specific algorithms for indexing and sorting images which may not produce absolutely correct results. One of the most accurate and sophisticated methods for annotating digital images is manual labeling. It is both time consuming and extremely costly. An alternative solution is to collect labels (tags) for images collaboratively through online games.

As we were thinking about educational aspects of visual concept maps, we decided to use learners (students) to manually describe photos collaboratively. In this way, we solve two problems at once. The learner will try to locate and identify objects in digital

images and thus we obtain credible descriptions about objects in each image. This helps a learner to strengthen his visual understanding about a specific domain of knowledge. More academically it is called manual annotation of images in order to create useful and plausible descriptions for the images. This manual annotation is done through an online game which we developed in our research work and we named it SYGY^{TAG}.

As we said earlier, SYGY is short for **Syncretctic Synergy**. The philosophy of such naming leans on the aggregation of the meanings for two words of Syncretctic and Synergy. Syncretctic means union of two different things. These two things are humans and computers. One has extraordinary discovery ability with feeble computational power while the other has limited unearthing skill with dominant computational strength. The other word (Synergy) stresses combined action of two or more agents which produces a result stronger than their individual efforts and we were thinking that computers and humans can work together to build fascinating results.

SYGY^{TAG} was developed during spring 2007 and the beta version was officially launched on 1st of April 2007. A specific domain name is registered to cover both SYGY^{TAG} and SYGY^{Map} games in a single website. The domain is SYGY.org and reachable in browsers at <http://www.sygy.org>. The game is developed using PHP, Adobe Flash and MySQL database. PHP is used for server side scripts while Flash was used for the user interface. We chose Flash, since it was handy and compatible with almost any browser in any operating system. Developing the GUI and building animations up were also speedy in Flash. The game is using simple XML messages to perform dialogs with the server.

Unlike some other games in which players need to be paired online, SYGY^{TAG} is designed in single player style. Each player would play alone at any arbitrary time. The game policy is to compare each new player with all earlier players who have played with that same image. In this way, there is no need to wait for a long time for the second player to arrive. Every player will start his own game right after he presses the Start button.

Sample Game sessions from both SYGY^{TAG} and SYGY^{Map} and their database structures are attached to Appendix I for better understanding of the game environment.

Figure 3-9: Illustration of the annotation procedure in Collimator

More advanced user is able to do following tasks as well:

1. Login into the voting system in order to review created tags for the image and vote for each tag. The voting system is designed to let users confirm or reject a tag which is proposed by other users. There are four choices for each tag: Confirm, Reject, Neutral and Abuse. If he chooses Abuse, it means that he wants to report this tag as a nonsense proposal. By default, all tags are in Neutral mode, which means that users do not have any comment about it.
2. Create Inferential Tags by selecting inferential tag button. Inferential tags are those which are not assignable to any visual region within an image. For example nobody can point out an object in an image and say this is autumn.
3. Export RDF documents in order to use it in other semantic web applications.
4. Import Ontologies and make instances. For example if the user imports an ontology about Pizza, when he sees a pizza in an image, he can make an instance of the Pizza ontology and assign it to that region.
5. Search for tags by providing the keywords. It also searches in available relations between regions.
6. Use Collimator to Map two regions. In the case that the user is interested in mapping two regions, he can choose one region as the subject and the other one as the object. He can also define the relation between the subject and the object. To

4.3.1 How to Play SYGY^{TAG}

SYGY^{TAG} is an online game for annotating digital images over the web. The game can be launched in almost any updated browser which has Macromedia Flash installed. The game is of the Single player type. During the game the player is shown a digital image for a short period of time and during this period, he/she quickly has to choose a proper drawing tool and locate an object inside the image. After selecting the object, the player needs to provide an appropriate label for this object in the highlighted textbox. After submitting the label, the score is calculated and displayed on the screen with a message. The score is computed based on matching the provided label and region with previous players. If the label was assigned by previous players, the system will try to compute the common area between the current drawing and the average drawing of earlier players. A bigger common area means a higher score. The player can repeat this process as much as allotted time allows. The player can also choose to pass an image if he/she thinks that the image is complex or enough labels are already assigned to it. Basically, the player has a few options before starting a game session. The player can choose one to four images to play. For example for one image, he/she has 90 seconds time to play and for four images, 220 seconds is allowed.

Figure 4-1: Screenshot from SYGY^{TAG} session. The Ball is selected.

As seen in Figure 4-1, the player has selected the ball and typed “Ball” as the label. There are three images in the queue to be played at the left hand side and the remaining time (seconds) is shown in big font at the bottom of game window. A list of four latest labels provided by the player is shown below the entry textbox and scores are shown for each label. Total score and current game score are shown at bottom left corner and drawing tools are accessible on the right hand side.

There are a few policies and regulations to which the player must adhere to them. Some important regulations are:

1. Labels must be typed in English language.
2. It is recommended to use one or two words as the label. The entry box is limited to maximum 25 characters. There is no need to type verbs or even adjectives.
3. It is better to use sensible and common understanding labels to earn better scores. Vague and complex wordings have a very low chance to get matched with former players.
4. The player is encouraged to find visual objects in the images instead of inferential observations.
5. The player should use a proper drawing tool to draw quickly and efficiently.
6. The player cannot cease the game time. But he/she can quit whenever needed.

At the end of allotted time, the total earned score is shown and a list of high scorers is displayed. The user can select a nickname and repeat the game as much as he likes. It is important to mention the system does not offer a same image to a player who has seen it in earlier game sessions. The images are selected randomly from a list of not-played images in the database.

4.3.2 Scoring algorithm in SYGY^{TAG}

As it is seen in Figure 4-1, we insert a grid on the image. The player has to identify an object and select those cells in the grid which cover the object. In this way, we easily find which cells are selected and where the object in the image is located. We use a matrix to record all selected cells for each label in our database. To calculate the score, we have very special measurements. We have digested different possibilities to make the game fun and fair. For this reason, we performed a unique study to identify the main

elements of being fair when we score the players. Our exclusive method for calculating score is attached to this thesis in Appendix II.

Our scoring system is based on following three assumptions:

1. The score is provided to any user who is matched with earlier players. For example the n th player of an image, will be compared with all $n-1$ players who had proposed labels for the same image.
2. The score must be relative to proximity of the player to all other players who have played the same image. It means if the current player is same as others and his selected region is very similar to an earlier player, he should earn a high score.
3. Since there is no chance to compare a fresh new label with earlier label(s), we decided to make a policy for this special case. As we could not gauge the validity of the new label against other players and we were also supposed to generate a primary score for the sake of game, we made a fair assumption that each new label is half by half correct or incorrect. We were also sure that the primary score would be adjusted once more labels appear and they get matched with earlier ones.

We proposed two types of scores: Dynamic and Static. For any label which is new and is not matched with any earlier or future player, we provide 500 Static score. If we assign 0 points to new labels, it means we meant that the label is not totally matched and if we provide 1000 (which is maximum point) it means that the user is completely matched with earlier players which in both cases, our interpretation is wrong. So we decided to provide 500 static score. If future players' labels are matched with the current player, we calculate the new score based on selected regions of both players and we deduct these 500 static score from total static score and add the real score to his dynamic score. When the third player arrives and he/she provides the same label, we again try to calculate the dynamic score for all three matched players and we update this score for these three players. These calculations are done each time a new player is matched with other groups of preceding players and the score of all related players is updated. Score changes in a period of time (e.g. 30 days) for two sample players (good and bad) are discussed in section 4.3.9. Since we wanted to encourage players to play more, we decided to provide a gift for each 10000 Dynamic points. For this reason we renamed Dynamic score to *Cash* points and Static score to *Stock* points in the game display. So for the stock points, the player has to wait to see if the future players would

be matched with him or not and if it happened, a new cash point is calculated and will be added to his total cash points.

4.3.3 Tools for Players in SYGY^{TAG}:

There are several drawing tools for the player when he/she wants to locate an object in the image. We tried to provide a range of tools which would help players to choose objects in more detail and even faster. For example, using Spray tool for selecting “Face” in Figure 4-2 is the best choice since the process of selecting cells would take less than a few seconds and the selection exactly covers the Face region.

Figure 4-2: Using Spray tools to select face region.

The drawing tools are:

- Select Tool: To select single cells. For each click one cell is selected by clicking the cell again, the selection is removed. Dragging the mouse to select multiple cells is active.
- Rectangular: To draw a rectangle to select a bigger area, which would be covered inside a rectangle. The rectangle also works reversly for selecting and unselecting the cell.
- Spray: Much freedom and faster painting is possible by using Spray. Spray is a perfect tool for choosing large objects inside images which have curves. Using rectangle tool for such cases is not a good choice since the player needs to cover exactly those cells which the object is located in.
- Select All: some objects are as big as an image. So by clicking on this tool, all cells will be selected.

- New Select: This is like a reset button. If the player decides to clean and start a new drawing, he/she should click on this tool.

4.3.4 Center of Attention

Here we discuss exactly what information is collected through SYGY^{TAG} and how we are using this information to find Center of Attention (CoA). As we said earlier, a cellular grid is overlaid on the image. The player has to identify an object inside the image and use the proper drawing tool to fill up those cells which are covering that object. After submission of a label, the client application will dispatch the following information to the server:

- Cell ID of covering cells
- Number of Cells
- Label assigned to this group of Cell

As seen in Figure 4-3, a few parts of an airplane are labeled. For example “Windows” is the smallest drawing with 12 covering cells, while “Engine” is the biggest region with 68 Cells.

Figure 4-3: Created Labels and Regions in SYGY^{TAG}

The Center of Attention is actually a dynamic region which is formed based on the commonsense of players. It shows an average segment of interest inside an image for a specific label among all players who have played with the same image. This CoA is calculated and updated each time a new player uses the same Label for a similar region. Assume three players have played with the image in Figure 4-3. All these players have labeled the cockpit. The first player would choose 28 Cells, the second would choose 40 cells and the third would choose 35 cells. In this case, some of the cells are in common in all three selections. These cells have the weight equal to 3. It is obvious that more players selecting a cell, the more weight it gains. We call it *Cell's Weight* since it shows how many players are insisting on it and thus we sense it is weighty. The cell's weight and Center of Attention are our groundwork for locating an object in an image. For calculating the score, we defined CoA as the combination of all those cells which have weight bigger than the average. Our score calculation method is discussed in more detail in Appendix II.

A player may not take action on an image and try to choose the second or third image from the left tab menu. This would indicate that the player found it impossibly hard because of poor picture quality or there is a dubious relation between the object and its label.

Having a deportation policy for those images we found “bad”, we can improve the quality of the collected data and rectify the fun level of the game.

It is worthy to mention again that these small pieces of information which are showing the location of objects inside the images are highly useful to computer vision researchers.

4.3.5 Application of Labels and Regions

SYGY^{TAG} provides us a high quality list of Labels and Regions which are essential for our second game called SYGY^{Map}. We will review how we are going to use these labels and regions for constructing Visual Concept Maps while we also talk about their practice in SYGY^{Map}. In addition to our own defined application of labels and regions, they are perfect for several other purposes which we discuss here:

a. Image Understanding: Machine learning, image processing and pattern recognition techniques have been improved over the past two decades but computers are still unable to understand images easily. Finding objects inside images would be one

part of the problem but understanding an image is still an unresolved problem. For example if there is an image which gives a picture of two fast trains about to crash, even the most sophisticated pattern recognition algorithms cannot understand if both trains will crash or not because they do not know that fast trains cannot stop in a short distance of each other.

b. Image Searching: Having proper labels associated to each image on the Internet would allow for highly accurate image search thus we have improved the image search results.

c. Accessibility: Properly labeled images would improve the accessibility to the Web for the blind and visually impaired individuals. Browsers could read the description of the images as well as webpage contents.

d. Filtering: labeled images could help filter unwanted images, like violence and nudity scenes.

e. Studying local influence: SYGY^{TAG} and similar games could be used for social studies on how a player's local environmental conditions might influence the labeling habits.

f. Practicing English: Players can practice English while they play SYGY^{TAG}. We may publish segments of photos with corresponding labels. In this way, English learners would learn English words by looking at related photos. This is one of the best methods to learn English visually. We may also translate English words to several other languages as well and let language learners choose their intended language to learn.

g. Self assessment: Players may find an opportunity to assess themselves on multiple skills such as typing speed, language skills, general knowledge, brain efficiency on different occasions based on their game scores.

4.3.6 Cheating in SYGY^{TAG}

SYGY^{TAG} is a collaborative game. Each player needs to be matched with a group of other players to maximize his/her scores. If some of the players collude to cheat on the game, collected data could be noisy and invalid. The cheaters would aim at earning high scores or even deliberately want to pollute the database. If players can not communicate outside the game environment then there is a guarantee that we obtain correct information. SYGY^{TAG} has several anti-cheating mechanisms. Some of the cheats are detected at the time of play while some others are treated offline. Before detailing our

techniques for avoiding fake entries, we point out that cheating efforts are quite uncommon. According to our experience, only a small number of players would like to test the system to see if they can earn high scores, while the majority of players prefer to play honestly. Some measures to prevent and detect collusion are:

- *Random Images:* We have a large image corpus in SYGY^{TAG}. Each time a new player starts the game, we randomly select a few images to display in the game window. The players have no option to select their own images. This will reduce the probability of launching same images into cheaters' windows. If for specific gaming purpose (like a class of students) the image corpus was small, there is a big chance that an image would be repeated for most of the players. The other anti-cheat techniques will cover such occasions.
- *IP Addresses:* We will record and keep track of all players' IP addresses. Through such development, we can identify if a single player has tried to login into different game sessions repeatedly. This player may want to repeat games with different user names in order to match with himself/herself and thus bring up his/her score. Two IP addresses from two geographical locations for two players are much preferable to having two IP addresses from a same IP class which could be two machines in a computer lab.
- *Bots:* Since our game is a web-based application, there is a high chance of getting polluted through bots (i.e. automated players). The bots would try assigning as many labels as possible during the game sessions. They are much faster than humans, so they may submit hundreds of random regions and labels in a short time. To detect this bulk submission of labels, we limit the speed of submission to 5 seconds. It means the fastest player cannot submit two labels in less than 5 seconds. Even in a game round with 220 seconds time, the bot is still able to submit 44 labels (each 5 seconds). Since we are recording the time of submission for each label, we can identify such malicious actions and we discard all current and future game play data associated with that bot.
- *Abnormal matches:* We know that two players or a group of players would match with each other if all of them provide the same labels and very similar regions for an image. In the cases where the image corpus is not big enough, some of the players would try to collude in order to obtain high scores. This is where we do random offline checking to see if any two or more players are matched with

each other for the considerable number of times. If we find such cases, we remove all labels and regions which are created by those players and suspend their accounts to stop them playing anymore.

- *Repeated labels:* The game policy is not to let any player use one label more than once in each image. Some players would repeat drawing different regions and assign the same label to that region. If we allow them to provide the same label again and again, they can obtain points by providing the same label each time. Even the smart player would try to enhance his/her score by drawing better regions over and over. In this case, we reject any repeated label for each image so the player is pushed to generate new labels in each try.
- *Aggregating data from multiple players:* In addition to the above strategies, we aggregate data from multiple players for a given image-word pair. By doing this, we can simply eliminate any outliers.

4.3.7 Quality of Labels and Selections

As we know, the labels and regions are provided by the players. If we want to use these labels in other applications, we need to have certain measures to identify if the labels and regions are valid or not. The SYGY^{TAG} itself provides such validation mechanism intrinsically. If two or more players provide a same label and similar regions (based on location and area), this implies that the label and the region are probably valid. More players having consensus on a label and region means we have stronger validity. In fact, since these labels and regions are coming from different players, they are potentially more robust and descriptive than labels and regions that an individual indexer would have assigned [9]. We can define a good label threshold as the number of people who have agreed on the same label for the same image. We define a good region threshold to a location where that at least $x\%$ of players have consensus over. If most of users draw a very similar region, it means that all of them are interested in that region so we may call that region, Center of Attention. At the moment, the threshold for the Center of Attention is 50% meaning that CoA is the region where at least 50% of players are pointing to that region in the image. Figure 4-4 illustrates a sample CoA. Four players have proposed a same label but the selected regions are not exactly the same. As can be seen, selections are overlapped. Each selection is shown in different shading. The CoA is also marked as the region where at least 3 players have

common selection. It means CoA is the region which more than 50% of players have selected.

Figure 4-4: Sample for Center of Attention – higher than 50% agreement

4.3.8 Experiment 1: Analyzing Database of SYGY^{TAG}

As the number of players was growing, we had more and more labels and regions in our database. Some tracking information such as IP addresses, login dates and times, number of images played was also recorded in our database. We realized that making queries to our database for analyzing and tracking the results is quite important. We were using simple queries to extract information from our database, but as we moved forward, we realized that we needed to design a very specific application which helps us to illustrate the results on the images as well. For example, we needed to know where the Center of Attention is located for each label in each image, or how many users have contributed to build such a center and what the exact delineated shape for each player was. In this way, we had more opportunity to analyze the performance of game players. Having the analyzer, we could also enhance our user interface in order to solve some of the bugs we would find through our analysis. A good tool was developed using the Delphi programming language. The tool was providing numerous possibilities for making fast and reliable queries and it was also very helpful in displaying drawn regions on the images. A sample screenshot from the tool is shown in Figure 4-5. The analyzer has different sections and each section is used to explore a certain type of

information. There is a preview window on the right corner to show the current selected image. The user can click on one sample image from the image table and then view all related labels to that image. The user is also allowed to see the frequency for each label. By selecting a label, the user can ask the analyzer to highlight the CoA region over the image. There is a gauge bar for changing the threshold for Center of Attention.

One major section is designed for analyzing the players. Using this section, the user can review some useful information such as number of times a player has logged in, number of labels a player has assigned, total scores for each player (including static and dynamic scores) and drawn regions for each player based on a chosen label.

Using this tool, we could successfully evaluate the quality of the Center of Attention for a random set of labels. The test was proving when a satisfactory number of players, label the objects inside an image, the Center of Attention is representing the object delicately even if some of those labels are not selected precisely. The test even proved that most of users provided valid and valuable labels in good faith.

Figure 4-5: Screenshot from SYGY^{TAG} Analyzer

4.3.9 Experiment 2: Commonsense Measurement through Time

Since each player is supposed to be compared to all previous players to earn a score, we designed a mechanism to generate points based on consensus of all those

players who have agreed on a same label and they all have similar regions. To do this, we construct a Center of Attention which was discussed in section 4.3.4. The Center of Attention is dynamically changed due to growing number of players. Each new player causes a new change to CoA. As explained our scoring formula is based on the CoA for each label and since this CoA is changed based on each new player, we need to update the score for the current and all previous players in our database. Total score is the sum of all generated scores for each label. So even if the score for one label changed, the total score has to be updated as well. It would be valuable to track these changes since we could understand how the commonsense among a group of players is shaping up. If all players draw very similar regions for a specific label, all will earn high scores for that region. If some of them draw a bigger or smaller region comparing to the average (average would be the Center of Attention with 50% threshold), they may obtain lower scores.

In our experiment, we tried to analyze this scoring oscillation for each player. We did server side coding to make queries from our database. We also tried to record the historical scoring background for each player in our database. At the first shot, it was interesting to see what happens in a space of time to one sample player's score. Before doing the test, we knew that the scores would increase or decrease for a group of players who were associated with the updated region and label. As we said, each time a new player assigns a new label to an image, the system tries to update the score for all those players who have proposed the same label earlier. When the player's drawing is very similar to CoA, he/she will obtain a high score. This implies that an inaccurate drawing will result in a low score. We designed a test case to evaluate our concept. To simulate two players, we manually created two accounts. One account represented a strong player and one account represented a weak player. For the strong player, we tried to label and choose objects in the images accurately but for the weak player, we used the same labels with inaccurate regions like much bigger or smaller selections. After we finished the labeling procedure, we promoted our game in different web societies and invited them to play it. They were supposed to label those images which we had labeled them before. We waited for 30 days to receive as many labels as possible. Figure 4-6 and Figure 4-7 are showing the results for our simulated players. As seen in Figure 4-6, the total score for the strong player increased during the period since his labels and regions were matched perfectly with the group of players who played after him.

But the scores for the weak player will descend since his inaccurate drawings

makes him isolated and drew him off from the average center of commonsense which is based on aggregation of all players.

Figure 4-6: Change of total score for a meticulous player over 30 days

Figure 4-7: Change of total score for an inaccurate player over 30 days

4.3.10 Experiment 3: Change of Threshold for Center of Attention

Studying Center of Attention was among one of our priorities to explore. As we stated before, the Center of Attention is the region in the image where at least $x\%$ of players have stressed on through their drawings. For our scoring purpose, we apply 50%

threshold for Center of Attention, but changing this threshold would reveal interesting information to us. To explore the effect of threshold on the shape of CoA, we picked up a sample image and tested different thresholds to see different results. Figure 4-8 illustrates 8 different regions based on 8 different thresholds. Ten players had labeled this image with ‘stairs’. Each player had a different drawing. The test reveals that when we put threshold low, the CoA is not focused on the object accurately but when we increase this threshold, we received a better focused region. This is because when we increase the threshold, we include more and more players in our CoA judgment and that beautifully presents a successful collaboration of ten players to select one object in an image. Some players would draw imprecise regions while some others will draw pretty accurate regions. We tested several other labels and increased the threshold from 0 to 90% and the results were prominent as we were expecting.

Figure 4-8: Exploring different thresholds for Center of Attention for “Stairs”

4.3.11 Possible Improvements

While we were developing SYGY^{TAG} we started to ask volunteers to give it a test. We understood that several improvements could be planned for SYGY^{TAG} in the future. We touch on some of these improvements here.

- *Checking Spelling*: The spelling is important for our matching purpose. Some of the players would have typos since they type fast to save the time. This way they may produce incorrect labels which we cannot use. It is good to have our own intelligent word parser, so in case we find a wrong entry, we would warn the player to do the correction. According to our primary database queries, almost 10 labels out of 100 entries are incorrect. This means 10% of labels have no chance of getting matched with other players and will not earn scores for the respective players.
- *Taboo Words*: One of the major improvements which were also proposed in ESPGame [50] is having Taboo Words. We identified that many players are trying to propose generic labels for any object they see in the image. That is because they want to earn points in lesser time so the most probable label would be the most generic one which they guess to type. For example, labels such as “Animal”, “Man”, “Woman”, “Lady”, “Boy” are frequently used. In order to upgrade our label’s quality and encourage players to produce more specific labels for the objects, we may use Taboo words. To incorporate the Taboo words in our game, we have to add a label instance counter for each object in each image. In this way, we know exactly how many times a label is used to describe an object. We may choose a very specific threshold to define what label is frequent so we can add it to our Taboo word list. Players are not allowed to type an image’s taboo words, nor can they type singulars, plurals or phrases containing the taboo words. Using this policy would bring more challenge to the game since players have to think harder to find better labels.
- *Word Sensing*: One of the issues that we were expecting to encounter was the word sensing challenge. Each player has specific preferences while picking up a label in his/her mind. We recognized that for most of the labels, we could use resources such as a thesaurus to identify word senses and insert this sense into our game database. Players would choose Verbs, Gerunds, Adjectives, Nouns or Adverbs as labels. Knowing the role of each label in our database would help us to choose proper labels while we are making queries to our database. For example, if we are searching for objects in our image corpus, we should probably search for type of Noun in our database. We also can use a part of speech detector to analyze those labels which are two or three words. For example a detector could tell us that Red in “Red Wine” is an adjective and the noun is probably an object.

4.4 SYGY^{Map}

Over the past two decades, several projects were proposed and developed for collecting large database of commonsense knowledge [26], [51], [52]. This knowledge consists of basic facts that the majority of humans accept as truth. A good review of commonsense knowledge and collection methods is sketched in chapter 2 of this thesis.

The motivation for collecting a large database of true statements is the belief that such knowledge is necessary to create truly intelligent and supportive systems. For example, we use the collected facts to build the concept maps. These maps systemically depict the organization of knowledge in a comprehensible illustration. There are many applications for such valuable databases such as improving semantic searches in the search engines or using them in machine learning projects to train robots.

In our research, building a large database of commonsense facts is not the primary goal but we are interested in collecting enough facts about ordinary objects in the particular digital image corpus (for example images of mammals). We introduce SYGY^{Map} as a collection tool for basic commonsense knowledge which would be visually observable in the digital images. SYGY^{Map} is a single player online game which tries to capture these small pieces of commonsense knowledge in a form of hobby. SYGY^{Map} is a sister game for SYGY^{TAG}. We are launching all those validated labels and regions, which were created in SYGY^{TAG}, into SYGY^{Map}. The player has to select one object in the image shown on the screen, and try to generate a fact about that object. The following section will discuss the basic game mechanism in more detail. A screenshot from the SYGY^{Map} interface is shown in Figure 4-9. In chapter 5, we will clarify how we use outputs from SYGY^{Map} to build visual concept maps.

SYGY^{TAG} was developed during spring 2007 and the beta version was officially launched on 1st of April 2007. The development was in parallel with SYGY^{TAG} game. Both games are accessible through <http://www.sygy.org>. The game is developed using PHP, Adobe Flash and MySQL database. The game is using simple XML messages to perform dialogs with the server.

Figure 4-9: Screenshot from SYGY^{Map} graphical user interface

4.4.1 Basic Mechanism of SYGY^{Map}

SYGY^{Map} is a single player game. The game purpose is to generate as many facts as possible in each game round. A game round would last for 220 seconds. During this period, an image will be shown to the user in the game window. This image is selected randomly from the image corpus which the game conductor is responsible for preparing. The player has to choose one object from a drop down menu. By choosing an object, the system will provide a fact-generation sheet which allows the player to choose the correct facts. The facts are simple statements which the player has to review and use his/her mouse to confirm or reject. These facts are about attributes and characteristics of the object. If the player feels the statement is wrong, he/she says NO to that statement. Saying YES to one statement means the player is sure that the statement is correct. It is vital to point out that the two choices for each statement are mutually exclusive. If choice one is correct, then the other choice is absolutely incorrect. Both choices are not correct or incorrect at the same time. In this way, we are sure that choosing the correct option would not be a big deal for players. After the player finished answering the questionnaire, he/she needs to submit it. The choices for all statements in the

questionnaire will be sent to the game server. All these statements will be compared with all earlier players. For any matched statement, the player earns 100 points. The total calculated score is displayed on the screen. The player can again choose another object in the same image to receive the fact-generation sheet or click on the Pass button to go to next image. Users must answer as many questions as possible, though there is no obligation to complete the fact-generation sheet for every image. For better clarification, questionnaire for the Frog in Figure 4-9 is shown in Figure 4-10.

The frog is Green.	☒ YES	☐ NO
The frog is Animal.	☒ YES	☐ NO
The frog has 4 legs.	☒ YES	☐ NO
The frog can jump.	☒ YES	☐ NO
The frog can fly.	☐ YES	☒ NO
The frog is insectivor.	☒ YES	☐ NO
The frog is carnivor.	☐ YES	☒ NO
The frog is calm.	☒ YES	☐ NO
The frog lays eggs.	☒ YES	☐ NO

Figure 4-10: Fact-generation sheet in SYGYMap game for sample object “Frog”

As we said before, all the images and labels in SYGY^{Map} are coming from the SYGY^{TAG} game. So those images which have no labels assigned to them in SYGY^{TAG} cannot appear in SYGY^{Map}. Right now, we are checking for at least two valid labels for each image to let it enter to SYGY^{Map} image corpus.

4.4.2 Game Conductor and Templates for the Facts

The game could be customized for a specific group of players thus the game conductor has a critical role for controlling all aspects of a game. For example a group of 30 students want to learn more about animals. The teacher has the role of game conductor in the class. A game would consist of a few images of mammals. The images are provided by the teacher and should be uploaded to the server. The teacher has to configure the game timing according to the number of students in the class. The teacher also has to prepare the students and inform them about the game procedures. The game could take a few minutes or hours to complete. The number of satisfactorily generated facts for each object in each image depends on the game configuration.

The game conductor is also responsible for preparing the statement templates. These statements are shown on the screen for every object in every image. These statements could be about characteristics of objects or their functionalities. For example in Figure 4-10, the game conductor has focused on characteristics of the objects. Some of these characteristics are: The Color, the behavior, food habits, body parts and the reproduction methods. For better clarification:

The fact “*The frog is Green*” - is about - the color attribute

The fact “*The frog lays eggs*” – is about – the reproduction method.

As it is seen, the game conductor tries to evaluate if all players know about some attributes of animals or not. The statements are general and easy to answer. These questions would be repeated for all other objects in all images. When the game time is over, we have a good database of facts about attributes of animals and we may compare those animals using similarity measurement algorithms. We have explained how we are using such a database to generate visual concept maps in chapter 5.

4.4.3 Using Templates or Natural Language

As we clarified in the previous section, the game conductor is responsible for preparing the templates for the statements. The labels for the objects will be replaced in these templates. For example the following template would be good when the image corpus is about animals:

The _____ is a Carnivore.

The game will replace the label for the objects (animals) in the black part of this statement. We could also let players type their own statements in free text entry boxes. In this case, they are using natural language. There are multiple reasons for using sentence templates instead of natural language:

- *Disambiguation.* Some of the words would have several meanings. By using templates, we will not have any problem with multiple meanings.
- *Categorization.* Different types of information could be entered if we use natural languages. We may not have control on the types of information but using templates; we can categorize the pieces of information and receive a variety of information for our special purposes.

- *Parsing*. If we use natural languages, we need to be careful about parsing the statements. By using templates, we avoid the complexities of parsing. Some of the statements have poor grammar and parsing them is sometime impossible.
- *United View*. By using templates, all players would have same vision while they are looking at images. They will all think about same facts about each object in each image and that would help us to measure the commonsense among them.

Once again, we mention that the templates have to be prepared by the game conductor and he/she needs to do primary studies to choose valuable templates which would have stronger educational response.

4.4.4 Scoring and Handling the Cheaters

The players in SYGY^{Map} are scored based on their consensus with other players. The game is more or less like a voting system. A definite number of players would play SYGY^{Map} during a period of time and they need to vote on facts. They all vote for the same series of objects in different images. In this way, we know exactly how many facts are created for each object at any given time. For example a sample game would consist of: 100 players, 20 images, 60 objects (3 objects in each image by average), and 20 facts in the questionnaire. Assume all players will play with all images and they vote on all facts for all objects. In such a condition, we have 2000 facts for each object. As it is clear, each fact has got 100 votes. Since each player chooses facts to be correct or incorrect, we exactly know how many players are voting for correct and how many are voting for incorrect. Normally, a majority of players will vote for the same option and a few others would be wrong on the other side. Actually there are two main groups of voters for each fact. We give a fraction of 100 points to each group of voters. If $x\%$ of them are saying a fact is correct then $(100 - x)\%$ are saying the fact is incorrect. The first group will receive x points while the second group will earn $100 - x$ points. Cheating in SYGY^{Map} is quite rare. Some players would try to pollute the entry data by providing wrong answers in questionnaires. We have certain mechanisms to avoid and remove some wrong facts for consistency of our database and better analysis results. Two major methods for capturing the cheaters are:

- *IP Address*: We detect the IP address of each player. If the player tries to login into the game many times with different accounts to vote on facts, we may identify

him by looking at his IP address. If the IP address is repeated many times for different players, we would suspend and remove all provided facts by all those fake players.

- **Wrong voters:** Some of the players will try to vote incorrectly for all statements in the questionnaire on purpose. We do checking on players' scores at random times. We identify if the player has got all the answers wrong in almost any questionnaire. We know that the probability of wrong voting to all simple facts in every questionnaire is very low. Even those weak players with very limited knowledge would be able to answer a few facts correctly in the questionnaire. Since wrong voters are suspected of trying to pollute our database, we take action and suspend their facts in our database.
- **Cahoot:** some players would decide to play together on different machines and they would all agree to vote same on the set of facts. It is very hard to say that a group of players are cheaters since their votes are almost very similar. This is because it is easy to vote for big number of facts almost all of players would choose correct choices hence they are similar to each other. For this purpose, we launch the images randomly to each player so we reduce the probability of having same images on the screen of a group of players at the same time.

4.4.5 Related Works

We have shown that SYGY^{Map} is a game for collecting commonsense knowledge. It uses a kind of voting system to identify true and false statements. There have been other approaches for collecting commonsense knowledge on the Web. We review a few of these approaches in order to specify the purposes and distinctions.

Cyc: Cyc (e.g., [26]) is one of the first attempts for constructing a large commonsense database. At the first step, Cyc was using paid experts to enter commonsense facts into the seed database. Experts were supposed to use CycL to enter the facts into the database. The CycL is a very precise language to avoid ambiguity problems. More data could be collected once this seed database would be mature enough. For example by combining those first expert entries, more facts could be generated.

One problem with Cyc is that it is limited by the power of the experts. The number of needed commonsense facts is much bigger than what experts can provide in a few years. Over the course of a decade, Cyc has been able to populate its main database with

around a million pieces of information [52]. Using games such as SYGY^{Map}, we have the possibility to collect millions of facts in a much shorter range of time. At the time of writing this thesis, SYGY^{Map} has a few thousands facts in the database.

Open Mind and Mindpixel: Few years back, the Open Mind [53] project was launched to use ordinary internet users to enter commonsense facts. Its database is depended on the volunteer contributors. Open Mind consists of several activities each designed to collect specific types of knowledge: spatial, hierarchical, implications, etc. Open Mind has gathered several hundred thousand pieces of knowledge in a few years.

On the other side, Mindpixel [51], like Open Mind, relies on ordinary Internet users in order to build up a commonsense database. Mindpixel is similar to SYGY^{Map}. The users will create and classify a statement as true or false collaboratively. In this way, a large database of true/false facts is built up. To authenticate the facts, the system rewards those users who consistently validate a fact inline with other Mindpixel users. The SYGY^{Map} scoring strategy is also very similar to this concept. One major difference between the SYGY^{Map} and Mindpixel is that the process of collecting facts is twisted as a challenging game environment.

Verbosity: Verbosity is one of the latest efforts in order to collect commonsense knowledge in a form of a challenging online game. The game will pair two online players and assigns one as guesser and the other one as a narrator. The roles are exchanged in each game round. The system will provide a random word to the narrator. The narrator should send some hints to the guesser in order to describe that word. The hints must follow specific templates so the narrator is bound to these sentence templates. If the guesser finds that word, both players will earn points and they repeat the game with a new word. SYGY^{Map} and Verbosity have the similar concept of turning job into fun to bring in more contributors in a short space of time. The major difference between them is that SYGY^{Map} is customizable based on the purpose of the game conductor to collect some facts (even a few) for specified study while Verbosity is aimed at building up a large database of commonsense facts. In SYGY^{Map} the facts are generated by the system and the user has to evaluate them. The consensus among players is our key to validate the facts while in the Verbosity, if the guesser finds the word, the generated fact by the narrator is assumed to be valid.

Chapter 5 Generating Visual Concept Maps

5.1 What is a Visual Concept Map?

As we discussed earlier in Chapter 1, we believe that concept maps are a kind of template or platform which helps us in organizing knowledge and representing it in a systematic way. Concept Maps are useful to categorize and identify main elements of our knowledge and at the same time as a wonderful tool, they can help humans to learn and distribute knowledge easier and faster. They are actually stimulating our brains to improve our visual learning ability.

As we know, concept maps have a long history in education and academia and they are not new gadgets to us. But the hidden potential of using concept maps and methods of creating constructive maps are still there to be investigated.

A few commercial and academic applications for drawing concept concepts are produced and made accessible through the web. Among free applications, CMapTools that is developed at the Institute for Human and Machine Cognition, [54] is considered to be the most comprehensive tool for drawing concept maps. The users are allowed to create their own concept maps and they can also share them online to construct a map collaboratively. Today, hundreds of universities and academic institutes are using CMapTools to construct and share their organizational knowledge.

In this thesis, we proposed visual concept maps as an alternative to verbal concept maps (a map which all concepts are written verbally in confined boxes). These visual concept maps (VCMs) are created semi-automatically through collaboration of learners and professionals. We believe that humans are good visual learners and that is why we studied the methods of integrating visual depiction of main concepts into existing ordinary verbal maps. It has been proved that for many concepts, people may not be familiar with verbal identification of the concepts but they can easily recognize and even memorize the concepts by seeing them. For example if we talk about an animal like Horse, a 3 year old child would not have any clear imagination about what a Horse is. Is it a food or a toy or an animal? A verbal concept map consisting of names of animals would be ambiguous to new learners but visual portrayal of each animal would help the learner to identify and match them with his own previous experience.

According to scientific studies, the human brain is powerful for memorizing and comparing visual elements. This is how one may remember the face of his friend after 20 years but he may not remember the name or the hometown.

To start with visual concept maps, we illustrate a sample one in Figure 5-1 which is about red fruits. This map represents the concepts (fruits) visually. As can be seen, the photos of the fruits are replaced in nodes of the map and the attributes of each fruit are placed around. The map is basically talking about Red fruits and their attributes.

In our research, we have focused on generating VCMs semi-automatically using two main information resources:

- Collection of annotated digital images
- The database of facts about annotated objects in the digital images.

For the first resource, we have developed two online applications for preparing a collection of annotated digital images which are Collimator and SYGY^{TAG}. For the second resource, we implemented an online game called SYGY^{Map} which allows users to evaluate groups of facts (propositions) about annotated objects.

Figure 5-1: Sample visual concept map about red fruits

As we discussed in chapters 4 and 5, Collimator was our first endeavor and mainly designed to use ontologies and semantic web technologies for annotating images while SYGY^{TAG} is an online game to label digital images and it is much simpler and more fun for users to work with. SYGY^{Map} will basically use database of annotated images (created by SYGY^{TAG}) to propose some facts about those objects which were labeled before. The players are supposed to evaluate the facts and specify if the fact is true or false. The administrator (instructor) is responsible for preparing the primary fact template to be launched in the game. Both games are in one package called SYGY and they were developed synchronously.

5.1.1 Focus Question for Visual Concept Maps

A good way to define the context for a concept map is to construct a *Focus Question* which clearly specifies the problem that needs to be resolved. Every concept map responds to a focus question, and a good focus question can lead to a much richer concept map. This is also same for our visual concept maps. The starting point for constructing a VCM can consist of only the focus question. The type of focus question makes a difference in the type of concept maps that would be built. A question like “What are green plants?” will lead to a declarative, more classificatory concept map than the question “Why do we need green plants?” Experiments show that not only the focus question, but also the root concept of a concept map have a strong influence on the quality of the resulting concept map [55].

5.2 Frequency and Support of the Facts in SYGY^{Map}

Each reply to a question in SYGY^{Map} is considered to be a true or false Fact about an object in an image. The players are actually evaluating the facts for each object in an image during a game session. By having an adequate number of players, we gradually get to consensus for the evaluation of each Fact. The Facts are generally among human’s commonsense knowledge so players usually do not need to have expertise to judge them. For example for the selected object (elephant) in the image shown in Figure 5-2, the following sample bundles of facts would be shown on the screen:

Bundle 1: Elephant ☐ has trunk ☐ does not have trunk

Bundle 2: Elephant	○ is big	○ is not big
Bundle 3: Elephant	○ has large ears	○ does not have large ears
Bundle 4: Elephant	○ has tusk	○ does not have tusk

The player needs to evaluate each bundle individually. It is observable that two complement facts are shown in each bundle and the player should choose and confirm one out of two for each bundle.

Figure 5-2: Elephant as a object selected in SYGY^{TAG}

Once each player evaluates the facts in each bundle for the label “Elephant”, we may perform simple statistical analysis and find out how many players have consensus on each fact in each bundle. Assuming that fact A and fact A' are both in one bundle, we know they are both complementary to each other. We define frequency of fact A as the total number of players who would confirm that fact A is correct in the bundle. We also define support for fact A as:

$$Support(A) = \frac{F(A)}{F(A + A')} \quad \text{And} \quad Support(A') = \frac{F(A')}{F(A + A')} \quad (5-1)$$

In above equation $F(X)$ is the frequency function which returns total number of players who have confirmed that fact X is existed. For example if 50 players say: “Elephant has trunk” (fact A) and 10 player say: “Elephant does not have trunk” (fact A') then $Support(A) = \frac{50}{50 + 10} = 83\%$ while $Support(A') = 17\%$. We express

$Support(X)$ in a percentage. Support and frequency of each fact are our base numbers for doing our statistical analysis.

5.3 Structure of Visual Concept Maps

In this thesis, we propose methods for generating two types of visual concept maps, namely Center-based and Similarity Visual Maps (or Similarity Maps). After evaluating the databases of Collimator and SYGY games, we portrayed a few possible sample visual concept maps and finally we selected these two models to work on. The models are actually two samples from a list of ideas we had for creating visual maps. The field is still open for interested researchers to evaluate our databases (annotated images and facts) in order to propose rich and useful types of visual maps for different kinds of applications.

Looking at Figure 5-1, one could identify that nodes of VCMs are a combination of verbal concepts and visual illustration of objects which are annotated through SYGY^{TAG} game. We defined verbal concepts to be attributes of the objects. These attributes are divided into two main categories: Characteristics and Adjectives. In our VCMs links are mainly carrying two types of relations: *Possessions* and *Existence*. By saying possession, we mean the relation between the two nodes is that Node A possesses (or has) Node B. By saying existence, we meant that Node A exists for Node B.

For better clarification of types of Nodes and Links, we provide an example below.

between each pair of nodes in the concept map. The following table shows samples of possibilities and similarities:

Table 5-1 Sample Facts for a VCM

FACT	MEANING
An Elephant <i>"has"</i> Large ears	It is possible for an elephant to have large ears.
A Banana <i>"is"</i> Yellow	It is possible for a Banana to be yellow
A Horse <i>"is extremely similar"</i> to a Zebra	Similarity between Horse and Zebra is high
A Bus <i>"is similar to"</i> a Minibus	Both objects are similar

VCMs are generated collaboratively. A good example of collaboration is a class of students. For the rest of our discussion in this chapter, we chose this class of students as a sample community of volunteers who are working together to generate a VCM. Students are basically asked to play with both SYGY^{Tag} and SYGY^{Map} during the class time. The instructor (administrator) prepares a collection of images and configures the games for students in order to teach them a specific topic. If the instructor is aiming to teach students that different animals are similar to each other and what type of attributes these animals have, he/she must prepare a collection of photos related to those animals and also define those specific attributes of animals which would make them similar or dissimilar. Choosing a collection of related photos and selecting attributes (which will be used in SYGY^{Map}) requires expertise in that specific field of knowledge. It is assumed that the instructor has this expertise and he/she is able to conduct configuring the game independently. It is also assumed that each student in a class has access to PCs connected to the internet since the student needs to play SYGY^{TAG} and SYGY^{Map} online using a web browser.

In the next two sections, we will explain two different perspectives for configuring and designing a game session in order to generate a VCM. Both perspectives are describing the issues related to instructors (administrators). The first view is from the point of technical requirements and the second one is about responsibilities of the administrator. In our case, the administrator is the instructor (teacher) who conducts the game and prepares students to play the game during the class. We believe that the instructor should be aware of all technical definitions and procedural steps in order to set up the games properly.

5.4 Technical Perspective

As we discussed in chapter 4, SYGY^{TAG} and SYGY^{Map} are online collaborative games which capture small pieces of knowledge from volunteers. These small pieces of knowledge are divided into two main categories:

1. Labels and locations of the objects in digital images.(Through SYGY^{TAG})
2. Facts about annotated objects.(Through SYGY^{Map})

For generating VCMs, we have designed an application which makes queries to the database of SYGY^{Tag} and SYGY^{Map} to propose the nodes and links of a VCM. We define our VCM generation system as the integration of SYGY games, Visualization Manager, administrator and contributors. Figure 5-4 clarifies the organization of the VCM generation system.

Figure 5-4: Flowchart of VCM Generation system

In our designed system, there are two key role players:

- **Administrator** (instructor) who is responsible for preparing the game. He has to configure and design the game. For example, if the administrator is a teacher, he must be aware of complexities of the photos, facts and time restrictions during class time. He also has to know what topics students need to learn and which concepts are challenging for students to comprehend. The role of administrator is quite critical in our system as he/she is involved in the logical and technical administration of the game.

- A group of **volunteers** who need to participate in the games. For example, volunteers could be students in a class who play SYGY games during their class time. Students are asked to locate objects inside a digital image and try to assign a label to those objects. They are also prompted to evaluate some pre-ready facts (statements) about objects in the digital images. The system allows students to learn a topic of knowledge in the form of a game. Students can find out more about the relations of concepts in a specific domain of knowledge by reviewing the generated VCM. The VCM is actually the product of teamwork of students so they would enjoy seeing the result of a collaborative job. This way they realize the topics intuitively as the learning process is mixed with the enjoyment of discovery.

The VCM generation system consists of few major input components:

Image corpus: A set of images are collected and incorporated into the system according to the purpose of the administrator. The images are mainly chosen from a specific field of knowledge and they clearly show the objects without any visual ambiguity. For example they may all be showing pictures of garments. The target object of the images (an article of garment in our example) must clearly stand out as the protagonist so that the players can distinguish it from other objects easily. If the object is too small or unrecognizable then the image is not a good candidate to include in the image corpus. The size (number of Photos) and other features of the corpus all depend on parameters such as the purpose of study; the way players are assumed to approach the images, the number of players, available time and space, etc.

Fact Templates: In SYGY^{Map}, each image is shown to the players alongside a fact-generation sheet that contains double choice incomplete statements or yes/no statements. The sheet is uniform across all images and the statements inside the sheet are adjusted by the game administrator. It is important to mention that the administrator is able to propose statements after completion of SYGY^{TAG} game. The sheet is actually used to prompt the user to describe the objects inside the images he/she is observing. For example if the image shows some garments and the user would select the “T-Shirt”, he/she would see following statements on the screen:

“The T-Shirt is long-sleeved” ---- ☐yes ☐no

“The T-Shirt is dirty” ---- ☐yes ☐no

“The T-Shirt is ugly” ---- ☐yes ☐no

In this example, the T-Shirt had been tagged before in SYGY^{TAG} game and the administrator was aware of all tagged objects in SYGY^{TAG}.

We refer to the act of completing or answering these statements as *fact generation* and each completed statement is considered as a *validated fact*. The list of facts is called a *fact sheet*. The administrator should provide fact templates as the input to SYGY^{Map}. Each template has an empty place which the game fills it with the chosen object name automatically. The second part of the template is a fixed word which is usually an attribute or characteristic of those objects which the administrator is interested in. For example fact templates for our previous example are:

<The Object> is (Short-sleeved)

<The Object> is (Dirty)

<The Object> is (Ugly)

In this example, the administrator was interested in attributes of garments. It is important to mention that the administrator is responsible for evaluating and choosing the most informative attributes or characteristics to be inserted in the fact sheets. The administrator is also responsible to select all those related objects from the pool of tagged objects (from SYGY^{TAG}) which could be selected by the user during SYGY^{Map}. For example, many objects would be tagged in the images which are not garments or related to garments and the administrator need to filter them before starting SYGY^{Map}.

Game settings: Students should answer as many questions as possible, though there is no obligation to complete the fact sheet for every image. The administrator sets a parameter that tells the system how many players should generate a particular fact for a particular image before the fact is considered satisfactorily strong; and when a fact exceeds that level, it is omitted from the fact sheet of that particular image. This parameter is set based of administrator experiences. Each image keeps circulating on the players' platforms until sufficient number of facts is generated for all objects inside that image. At this time, the image is removed from the game automatically. A few other parameters are also important to take into consideration. Some of these parameters are: time allocated for each game session in SYGY^{TAG} and SYGY^{Map}, the total number of images to be played by each player, number of players and the total time needed for reaching a reliable consensus point based on the number of generated facts in SYGY^{Map}.

Other major components of VCM generation system are:

Fact database: The generated facts about each image are stored in a database for further analysis. The following information is stored in each record of the database:

- **ImgID:** The ID of the image that the fact is created about
- **FtrID:** The ID of the feature that is attributed to the image by the user
- **FtrType:** The type of the feature (i.e. either a property or an attribute)
- **FctFrq1:** The percentile of users that have chosen option 1 for the corresponding statement.
- **FctFrq2:** The percentile of users that have chosen option 2 for the corresponding statement.
- **QstnType:** The type of statement that has been tackled by the user (i.e. double choice or yes/no).

Visualization Manager: This is an application which makes queries to databases of SYGY^{TAG} and SYGY^{Map} in order to visualize the information in the form of a Visual Concept Map. The manager chooses the related nodes and suggests proper linking phrases for the VCM. As we mentioned earlier, we proposed two types of VCMs in this chapter: The Center-Based VCM and Similarity Visual Map (Similarity Map).

- **The Center-Based VCM:** The Center-Based VCM allows the user to navigate through various images and facts created about the images. The process must start with the user choosing a specific object in one image. The object will be located on the centre of the map with various related features being scattered around it. Each feature will be linked to the object using a labeled arrow that indicates how strongly the feature has been attributed to the object. The user can also specify a minimum support that the features on the map must have or they will not be displayed on the map. For example if an attribute is rarely assigned to one object, this attribute will not be displayed. The user then moves on to choose one of the features and the map toggles to scatter all of the objects that have been linked to that feature. Labels again indicate the strength of the attribution which can again be controlled by the administrator. Saying strength of attribution we mean how much consensus exists on assigning one attribute to one object. In next steps, the user can choose an object which makes features being scattered around the object, and the process of toggling between objects and features could be repeated again and again. Figure 5-5 shows the scheme of a centre-based

VCM in three stages of the toggling process. For the purpose of clarity and size of print in this report, the nodes are shown verbally.

Figure 5-5: Sample centre-based diagram showing toggling process

The procedure of creating this diagram is discussed in more details in section 5.7.

- **The Similarity Map:** The Similarity Map allows the user to find out about the most similar objects within the image corpus. Based on the features that have been attributed to various objects in the system, the map shows a hierarchical network of the most similar objects. Objects (visual representation of them) will again be connected to each other using labeled arrows that indicate how similar the objects are. The user must specify a particular object as the root of the map. Other objects will be scattered under the root. As the levels of the map expand the vertical distance of each lower level is indicative of its difference from the nodes in the upper level of the map. Ultimately, all features of each node that have a specified support (for example 50% of users have used it) will be sketched around the corresponding node. The user can specify a subset of features according to which similarity is calculated. He/She can also specify a minimum similarity under which no two objects will be connected on the map. Figure 5-6 shows the scheme of a similarity map for a database of garment objects. The user has chosen the feature *cloth season* and *material* for calculating similarity and as can be seen from the picture, clothes belonging to similar seasons are vertically close.

Figure 5-6: Sample Similarity Map, visualizing the classification of garment based on the season they're made for

The procedure of building this diagram is discussed in more detail in section 5.8. The similarity between each of the two objects is measured using the algorithm explained in section 5.6.

5.5 Administrator Perspective

The person who is involved in the logical and technical administration of the SYGY games is called *Administrator*. As discussed earlier, the administrator is in charge of modifying and manipulating game settings. The administrator is also responsible for preparing the image corpus and constructing an appropriate fact sheet. The administrator would be a teacher in the class or even a researcher conducting an online research about knowledge acquisition techniques. The administrator should perform the following steps in order to set up and run the game and analyze the results properly:

- **Step 1: Defining the purpose of study:** The administrator must decide (or be notified) about the purpose(s) of the study he/she is going to conduct. For instance, suppose an elementary school teacher wants to acquaint his/her students with different implications of specific features of animals, like their color, size, skin texture (e.g. bare skin, feather, fur, scales, etc.), moving style (e.g. two-legged, four-legged, crawler, etc.), predation class (e.g. herbivore, carnivore, omnivore, etc.). He/she must know the implications he/she is going to introduce to the students after getting them to specify the physical features of animals. For example, he/she may be interested in showing the students how these physical features would reveal scientific information about an animal (e.g. the class it belongs to, its place of origination, etc.). The teacher must choose these features according to the purpose he/she is attempting to fulfill. For example, if he/she is interested in showing the students how the physical features of an animal would reveal useful information about its place of origin, then color and skin texture would be better choices than moving style and predation class, since these features are more suggestive of the geographical characteristics of the animal's place of origin.

- **Step 2: Specifying target objects:** At the next step, the administrator is supposed to specify the set of objects that he/she is going to discuss. This set of objects must deliberately be chosen according to the diversity of features they exhibit. For example, in the example above, the teacher must choose a set of animals that are easily distinguishable according to their size, color, skin texture, moving style and basic food.

- **Step 3: Customization of fact sheet:** After the set of distinguishing features is specified, a fact sheet must be designed to let the players generate appropriate facts about each object. This sheet will be universal and uniform across all images that are going to be displayed to users, and thus must provide the possibility of generating all available facts about an object. The administrator must take the following steps to create this sheet:

- First of all, various categories of features need to be distinguished. Common features of a graphically displayable object include:

- i. **Characteristics:** *Physical characteristics* of the object (e.g. its body organs) and its *capabilities* (e.g. its capability to think, run, crawl, swim, manage, etc.). Both types can be attributed to an object using the term “has (the ability of)” (e.g. “Elephant has four legs”, “Man has the ability of thinking.”). Physical

characteristics are usually countable (e.g. “Horse has four legs”, “Fly has a hundred eyes.”), while capabilities are non-epistemological processes (e.g. “thinking”, “running”, “flying”). In our example skin texture would be a physical characteristic (“has fur”) while the style of moving would fall into the category of capabilities (“has the ability of crawling”).

ii. Attributes: *Visual attributes* that the human eye can process (e.g. its color, size, shape), *taxonomical attributes* that humans can deduct about an object (gender, occupation, scientific class) or *attitudinal assertions* that they attribute to the object (e.g. its appearance, the feeling it conveys). These features would usually be attributed to objects using the term “is” (“Blood is red”, “Wolf is carnivore”, “Frog is ugly”). In our example size and color would be considered the visual attributes of animals (“Bear is brown”, “Elephant is big”) while basic food that they eat would reveal their class of predation and hence fall into the category of taxonomical attributes (“Horse is herbivore”).

- Secondly, for each feature, different values that the feature might exhibit should be specified. For example in the set of animals collected for the study, the color values that the animals in the collection might have should be specified (e.g. white, black, brown, gray, etc.). The same procedure should be performed for values of size (e.g. small, big), skin texture (e.g. bare skin, feather, fur, scales), moving style (e.g. walking on two legs, trotting, crawling, jumping, flying) and predation class (e.g. carnivore, herbivore, omnivore).
- Thirdly, values of each feature are integrated into an incomplete, double choice statement describing the target objects (e.g. for the visual attribute of size, an appropriate statement would be “The <target animal> is big ☐Yes ☐No”). For the sake of convenience, speed and ease of analysis, no statement must have more than two choice values and the values should be mutually exclusive, hence negating each other and disambiguating the problem of choice for the user. For example the <target animal> is big or not. There is no third option in the middle. For features such as color where several values are on the list, each value has to be integrated in a yes/no declarations, so that the number of statements of this type grows to equal the number of values. For example we have 3 values for walking ability of an animal:

- “The <target animal> has the ability of walking on two legs” ☐yes ☐no

- “The <target animal> has the ability of trotting” ☐yes ☐no
- “The <target animal> has the ability of crawling” ☐yes ☐no

In both type of statements the connector of the subject to the target property or attribute must be chosen according to the type of the feature. For example the appropriate connector for properties of objects would be “has (the ability of)”, and for the attributes “is”.

- **Step 4:** Time estimation: The next step for the administrator is to estimate the total game time needed to create enough facts for the system in order to come up with a reliable results which should be based on consensus of players. Since each player is just allowed to fill up one fact sheet per object in SYGY^{Map}, we should consider the satisfactory number of players who are supposed to generate facts for each specific object in the database of SYGY^{Map}. In order to estimate the time it takes to reach it, the administrator must calculate the following parameters:

- *P*: the number of players
- *I*: the number of images that the administrator is going to use for visually representing objects to the players (which can be different from the number of objects)
- *S*: the number of incomplete statements and yes/no questions on the fact sheet which he/she has customized in the previous step
- *FperM*: average speed of a player in generating facts using the statements on the fact sheet. This speed is estimated in terms of fact per minute.

The maximum number of facts (*MaxF*) which is possible to generate is computed from the following formula:

$$MaxF = S \times I \times P$$

And the average playing time needed to generate this maximum limit is calculated according to the following formula:

$$PT_{average} = \frac{MaxF}{(FperM \times P)} \quad (5-1)$$

For example if the game has 40 players, the number of images is 20, there are 60 available statements about each object and the average speed of each player is about 20 facts per minute, then the maximum number of possible facts will be equal to:

$$MaxF = 60 \times 20 \times 40 = 48000$$

And the maximum time needed to accomplish this limit will be about:

$$PT_{average} = \frac{48000}{20 \times 60} = 60 \text{ min} \quad (5-2)$$

This measure will allow the administrator to estimate the time feasibility of the study and thus adjust *I* and *S* if the estimated time seems unreasonable.

- **Step 5:** Preparing the image corpus: The administrator is in charge of providing a set of images for the players. The images must visually represent various target objects with various target features vividly so that the detracting ambiguity which keeps the player from generating a reliable fact is minimized.
- **Step 6:** Configuring the game: After all required data is collected, the administrator must incorporate the image corpus as well as the fact template into the system. He/she must also set the level of *satisfactory consensus* for the facts which expresses the sufficient number of facts for each object. This is actually a parameter which specifies the percentage of players that must create a specific fact so that the fact is considered highly reliable and ready to be processed. In our example the teacher may be interested in setting this parameter to 90%, indicating that at least 90% of students must generate a specific fact for a specific object so that the fact is reliable to be taken into consideration. Facts that reach this level are omitted from the fact-generation sheet of images while the resting facts circulate among users until they reach this level of satisfaction.
- **Step 7:** Running the game: After all of the options are configured, the administrator runs the game on multiple computers and controls the playing procedure until all facts for all images reach the satisfaction level and the game is terminated.
- **Step 8:** Generating VCM: The last stage of the procedure is to generate a VCM according to the pre-specified purpose. As previously discussed, we have proposed two types of VCMs which are explained in section 5.7 and 5.8.

5.6 Similarity Measurement of Objects

In this section, we discuss the computations and algorithms for finding the similarity between each pair of objects in our annotated database, based on the set of

features assigned to each object in our fact collecting game called SYGY^{Map} which was reviewed in section 4.4.

Research on human judgments of semantic similarity between objects has existed in the past four decades. Several approaches for measuring the similarity are proposed which we try to cover shortly before we describe our own measurement approach. Supplementary topics like linguistics and world phenomena beside human bias for the selection of features that delineate similarity, could be covered in a longer context. We focus on human bias for similarity judgments and there would be no absolute solution giving 100% satisfactory results for all different applications. It is good to note that the theorem of the Ugly Duckling [56] states that “in the absolute absence of bias any two categories of objects are equally similar”.

Typically, the idea of similarity is discussed through the idea of distance. It actually involves finding a method to measure distance between any two objects and it means that most similar objects are those which are nearest to each other. The notion of metric is used to express the distance. We define distance measure D as a metric to gauge the distance between each two objects in a set of Objects O . Following conditions must hold for any three objects o_1, o_2, o_3 in O :

Non-negativity: $D(o_1, o_2) \geq 0$;

Reflexivity: $D(o_1, o_2) = 0$ if and only if $o_1 = o_2$;

Symmetry: $D(o_1, o_2) = D(o_2, o_1)$;

Triangle inequality: $D(o_1, o_2) + D(o_2, o_3) \geq D(o_1, o_3)$;

We can assume that objects are points in a N dimensional space, where N is the number of distinct attributes. In feature-based approaches to similarity, each object has a feature vector and a distance metric between objects could be defined as Minkowski metric [57] with a specific parameter $r (r > 0)$:

$$L_r(o, o') = \left(\sum_{k=1}^N |o_k - o'_k|^r \right)^{\frac{1}{r}} \quad (5-3)$$

Where r is the order for Minkowski distance, o_k is the value of the k^{th} feature of O (o is Object) [58]. L_1 (i.e., the above expression for $r = 1$) is known as Manhattan or city-block distance. L_2 is the familiar Euclidean distance. The Minkowski metric

clarifies that distance is inversely related to similarity strength. According to [59] L_1 and L_2 are the most commonly used in feature-based similarity measures.

The Minkowski metric is a kind of geometric model to calculate the distance between two objects. Most of geometric/spatial models assume that humans, in their judgments of similarity and/or dissimilarity of stimuli, pay equal attention to the various dimensions.

Several studies of human similarity judgments [60] indicate that when humans estimate the similarity of concepts, they thoroughly violate all of the above properties of distance metrics we discussed earlier. For example judgments by humans violate triangle inequality in the following way: “ball” (o) and “moon” (o') are both round, and “moon” (o') and “lamp” (o'') both can give off light, but “ball” and “lamp” are less similar than either of the mentioned pairs (ball-moon) and (moon-lamp) [61].

5.6.1 Tversky's Contrast Model

Tversky (1977)[61] formulated an alternative set theory-based model called the contrast model similarity. Unlike geometric/spatial models, the contrast model does not represent stimuli as points in multidimensional space. Rather, it defines stimuli as sets of features and the similarity of any two stimuli as a linear function of a measure of their common and unique/distinctive features. According to the contrast model two objects are more similar if they have more common features and fewer distinctive features. Researchers in cognitive science state that "humans will place more weight on common features when judging similarity and more weight on distinctive features when judging dissimilarity" [62].

According to Aggarwal et al. “In most high dimensional applications, the choice of the distance metric is not obvious; and the notion for the calculation of similarity is heuristic” [63].

We decided to use Tversky's contrast model instead of the metric distance model for similarity measurement. As we mentioned earlier, Tversky's model is based on common/distinctive features and formulated as:

$$S(o_1, o_2) = \theta f(o_1 \cap o_2) - \alpha f(o_1 - o_2) - \beta(o_2 - o_1) \quad (5-4)$$

Where o_1 and o_2 represent the features of the two objects and $S(o_1, o_2)$ is the similarity between two objects. For the rest we have:

- $(o_1 \cap o_2)$ represents the common features of o_1 and o_2
- $(o_1 - o_2)$ represents those features of o_1 which o_2 does not have
- $(o_2 - o_1)$ represent those features of o_2 which o_1 does not have
- f is additive function ($f(A \cup B) = f(A) + f(B)$ when A and B are disjoint.)
- θ, α, β Reflect weight and are non-negative.

Figure 5-7 illustrates the relations between two sets as a means of clarification.

Since SYGY^{Map} is a game based on human perception and judgment, Tversky's model seems to suit the best. But modifications are still crucial for customizing the formula so that it can be used to measure the similarity of two objects as defined in our system. A distinctive characteristic of our system is that features have a specific validity and this validity is measured in terms of frequency (number of people who have proposed a same fact) In other words when a specific feature has been linked to a specific object by more players that feature is considered a more valid characteristic for that object and hence it has more support. We decided to integrate this validity system into Tversky's model. The first step is to remove those features which have less than 5% frequency support (i.e. frequency percentile) and equalize them with zero. In fact we round all features with supports less than 5% to 0. This is because we consider these features to be completely invalid and insignificant to be accounted in statistical analysis. Using 5% is based on our experience with total number of facts which we were collecting in different test. 5% is a just primary cutting parameter and could be evaluated in a different research context if needed. An object would be bereft of a specific feature if the support for that feature is zero. For example if we ask 100 contributors to confirm or reject the statement: "Cucumber is Green" and we find that 4 out of 100 say this fact is false, we assume that assertion by these people is invalid and convert the support from 4% to 0. When support for a feature is equal to 0, it means that the object does not have this feature at all (Means: this feature does not exist for this object)

Now suppose that this process has been done for two objects A and B. The remaining features have their own support values and some of them may belong to both objects (common features) while others may not.

Figure 5-7 shows a sample state with objects A and B and their corresponding significant features.

Figure 5-7: Sample object pair and their common and distinct features

We have adopted the Tversky's measure and created the following formula based on it:

Suppose F is the total number of features with non-zero support for both objects (after we performed the check for 5% strength), F_A is the number of features that object A has and object B does not have, F_B is the number of features that object B has and object A does not have and F_C is the number of common features. If $Sup_A(x), Sup_B(x)$ are the supports of feature x in objects A and B, then the average distance between the frequencies of common features can be calculated from the following formula:

$$Dist(A \cap B) = \frac{\sum_{x \in A \cap B} (|Sup_A(x) - Sup_B(x)|)}{F_C} \quad (5-5)$$

This measure tells us about the level to which A and B's common features vary in terms of their validity (frequency). Now the absolute similarity of object A to object B can be calculated according to the following formula ($0 < Dist(A \cap B) < 100$):

$$AbsSim(A, B) = (100 - Dist(A \cap B)) \quad (5-6)$$

By saying absolute similarity we mean a kind of similarity which is based just on common features of two objects. But this absolute similarity measure does not tell us anything about the differences of A and B since it does not account for the distinctive

features of A or B. In order to take those features into consideration, we calculate a weight for features in $A - B$ and $B - A$:

$$W(A - B) = \frac{F_A}{F} \times \frac{\sum_{x \in A-B} Sup_A(x)}{F_A} \quad (5-7)$$

is the average support weight of all features that object A has and object B does not have. And

$$W(B - A) = \frac{F_B}{F} \times \frac{\sum_{x \in B-A} Sup_B(x)}{F_B} \quad (5-8)$$

is the average support weight of all features that object B has and object A does not have. Now the overall similarity of the two objects can be computed according to the following formula:

$$Sim(A, B) = \frac{\frac{F_C}{F} AbsSim(A, B)}{\frac{F_C}{F} + \alpha W(A - B) + \beta W(B - A)} \quad (5-9)$$

The coefficient $\frac{F_C}{F}$ tells us how much the two objects have in common and it balances the absolute similarity measure according to its extent. α and β specify how much the administrator is willing to certify each object. For instance if α is bigger than β , then the similarity is said to be measured “more from the viewpoint of A”. The fractional form of formula conducts a direct relation between the overall similarity and the absolute similarity and an inverse relation between the overall similarity and the difference of objects. The coefficient $\frac{F_C}{F}$ is repeated in the denominator to avoid zero division in cases where all features are in common between A and B and hence both $W(A - B)$ and $W(B - A)$ are equal to zero. In this case, the overall similarity will be equal to the absolute similarity of A and B.

It should be noted that $Sim(A, B)$ is bounded between 0% and 100% and all the values of $AbsSim(A, B)$, $W(A - B)$ and $W(B - A)$ are used in percentile.

5.7 Center-Based Visual Concept Maps

The Center-Based VCM reveals the most prominent, most recognized features of an object. Having chosen a specific object in one image, this map shows the facts that have been frequently generated for that object. Choosing each fact, the map moves to show other objects for which the same fact has been frequently generated. Again, by choosing each object, the diagram toggles back to showing the common facts about that object. Edge labels will show the strength of the fact which means how much consensus exists on that fact (i.e. the support for the fact). It is important to mention that the lower threshold value for the level of support for each fact is specified by the administrator. For example the administrator may decide that facts that have been generated by less than 10% of players must not appear on the map which means facts with less than 10% support are filtered.

In order to draw such a map, the following steps are performed in the system:

1. **Analyzing the query:** First of all the query submitted to the system will be analyzed for consistency by the computer. As a strategic decision that we made, the logic of the system does not allow the process to start with a feature so the initial input must necessarily be an Object (By saying object we mean a portion of an image). If the requested object is found, then the first node of our map is coined. Otherwise, the user receives an error, prompting him/her to try again with another object. The user's also had the choice to choose minimum support threshold for the facts which should be appeared in the map.
2. **Feature specification and labeling:** After the central node is specified, all of the statements in the fact database will be analyzed for their frequency regarding the central node. This will be carried out by scanning the fact database and searching for the features that have been linked to the object at the central node. If the support for the found features is equal to or higher than the minimum specified by the user, a single node will be assigned to each selected feature. The node will be labeled same as the value for the feature (e.g. "warm cloth").
3. **Edge labeling:** After all proper features are selected and the corresponding nodes are created and labeled, an edge will link the central node to each feature. The edges will be labeled according to the type and frequency of the feature in the fact database. These edge labels consist of two parts:

- a. *The relational process*: This part of the label is specified according to the type of feature. If the feature is a type of *property*, then this part will be equal to *said to have* while for the type of *attribute*, this part will be equal to *said to be*. These two expressions are our proposal and they could be redefined in future if needed. Finally the chosen expression will precede the adverb of frequency.
- b. *The adverb of frequency*: This part of the label is specified according to the frequency of the fact that is linking the feature to the central node. The higher the frequency, the more intensified the corresponding adverb will be. We defined our own customized quantization scheme for differentiating the adverbs. The presented scheme could be defined later based on requirements and a different list of adverbs could be also used. The following table shows the frequency percentiles and their corresponding adverbs. As seen, we opted to choose a high resolution division of adverbs. Lower resolutions are also acceptable (like 3 levels or 4 levels instead of 11 levels)

Table 5-2: Fact frequency percentiles and their corresponding adverbs for edge labels in the centre-based VCM

Always	100%
Almost always	90-99%
Usually	80-89%
Generally	70-79%
Regularly	60-69%
Often	50-59%
Sometimes	40-49%
Rarely	30-39%
Seldom	20-29%
Scarcely ever	1-19%
Never	0%

For example, the edge label for a property with the frequency percentile of 78% will be equal to *generally said to have (feature X)*, while the edge label for an attribute with the frequency percentile of 34% will be *rarely said to be*. As it is seen, the expression pattern is: *Adverb of Frequency + (Said to be) OR (Said to have)*

4. Graph sketching: After the nodes and edges are all specified, then the map is ready to be sketched. The central node will be represented using a thumbnail of the corresponding image. All other nodes will be scattered around the central node as ellipses containing labels that were specified in the second stage. They will be linked to the central node using labeled edges. The edges will be labeled according to the modifications performed in the previous stage. Figure 5-8 shows the initial stage of constructing a center-based VCM. Edges are labeled according to Adverb of Frequency.

Figure 5-8: Sample centre-based VCM at the initial stage

5. Toggling between the two modes: After the map is displayed on the screen, the user can explore other aspects of the system by clicking on any of the features that surround the central node and a new center-based map will be generated which the central node is the feature. The following steps are taken in the system to re-modify the map accordingly:

- a. **Image specification:** Upon clicking any of the feature ellipses, the name of feature (its ID) is passed on to the system that allows it to look up in the fact database for all of the objects that have been linked to that specific feature more strongly than specified by the minimum support threshold. The query performance is based on number of total facts created during the game. So if the game is played for a limited time or limited number of players, the number of facts is fair enough to have a

fast response from the database. The corresponding objects will then be retrieved and a node will be assigned to each of them in the map.

- b. Edge labeling: Since the type of the new central node is already specified, the relational process that needs to be included in the edge label does not need to be changed. The adverb of frequency will again be specified according to Table 5-2.
- c. Graph sketching: The map will be sketched using thumbnails of each image. Objects will be linked to the central node (feature) using labeled edges that indicate the strength of their connection (i.e. fact frequency). Figure 5-9 shows the Center-Based VCM after the user has selected feature 6 among the features of the main object. All objects (object1, object2 ...) are linked to feature 6 and the links are labeled according to their frequency strength.

Figure 5-9: Center-Based VCM after selecting a feature to explore (second stage)

The user can again click on any of the images to change back to the initial mode and the toggling process can continue again and again. This will allow the user to browse through the system thoroughly and examine various relationships between objects and their properties and attributes. The minimum support threshold can also be justified at each toggle, allowing the user to prune the map for more important links. Figure 5-10 illustrates the toggling from object to feature and from feature to object.

Figure 5-10: Toggling between Object and Feature in sample Center-Based VCM

5.8 Similarity Map

The Similarity Map shows the most similar objects of the system in a hierarchical structure. This hierarchical map is supposed to represent the network of the most similar objects of the system. This similarity is estimated based on different features that objects have and the commonality and diversity of those features among multiple objects. The user must specify an object as the root of the map. All other objects will be scattered under the root. The vertical distance of each level of the map indicates its difference (in terms of properties and attributes) from the upper level. The edge labels indicate the level of similarity between each two connected objects and the user can specify a similarity threshold which no two objects will be connected on the diagram below this threshold. The user can also specify the features upon which he/she likes the similarity to be estimated. For example the primary school teacher would be interested in choosing *size*, *moving style* and *predation class* for generating a map that shows the classification of animals based on their wildness.

The following steps describe the process of creating a similarity map:

1. **Choosing the root object:** First of all, the user must choose a specific object as the root of the similarity map. He/she can limit the set of features that similarity will be calculated according to, based upon his/her own purpose.

2. Calculating the similarity matrix: At the second step, the system will calculate joint similarities between pairs of objects and create a table accordingly. The similarity of pairs of objects will be calculated according to the formula introduced in section 5.6 and it will be stored in a matrix. Since this matrix can be visualized using a complete graph, it can also be titled the similarity graph of the objects.

3. Extracting most similar objects: Since the administrator is looking for the map of the most similar objects (regarding their features), this graph will be a sub graph of the global similarity matrix that is created in step 2 with the greatest similarity links among its nodes. This sub graph is the maximum spanning tree of the similarity matrix. So the map can be sketched by applying Prim's algorithm (for finding maximum spanning tree in a weighted graph) to the existing matrix provided at the previous stage. The algorithm will carry out the following process in order to create the map:

- a. The initial tree-node set is set to empty.
- b. The root object is added to the initial node set.
- c. Among all other objects, the most similar one to the root is selected and added to the node set, along with the edge that links it to the root.
- d. Among all other objects, the most similar ones to the objects in the node set are added to the node set along with the edge that links them to a node in the node set, unless this edge creates a loop in the tree. If there are more than two objects satisfying the condition, the one with the greater similarity to the root will be selected. If all have a equal similarity to the root, the one that stands at a higher level is selected.
- c. Step d is repeated until all nodes are added to the tree.

4. Replacing nodes vertically: The resulting sub-graph is like a tree which starts from the root node and is expanded to all other nodes according to the maximum weighted tree explained in the previous section. The root node is placed on top of the page while other nodes of the graph are placed vertically below the root node. The vertical distance of each level of the tree will indicate the difference of each object from the object(s) in the upper level.

5. Labeling the edges: In addition to distance, the edges linking each two objects are also labeled according to the similarity measure of the nodes they are linking. The format of labels will be like: "*is <adverb of degree> similar*" to and the corresponding adverb of degree will be decided according to the similarity measure

between objects that the edge is connecting. Table 5-3 Shows various similarity measures and their corresponding adverb of degree.

Table 5-3: Adverbs of degree for Similarity measures between two objects to be used for the links

Exactly	100%
Perfectly	90%-99%
Extremely	80%-89%
Strongly	70%-79%
Quite	60%-69%
Almost	50%-59%
Partially	40%-49%
Moderately	30%-39%
Hardly	20%-29%
Scarcely	1%-19%

For example if the similarity of two objects is estimated as 35%, then the edge linking those objects will be labeled *<is moderately similar to>* which “moderately is chosen from Table 5-3.

Again, the user can specify a minimum similarity threshold under which no two images will be linked on the tree.

6. Expanding to features: At the last step, we can add features around each node of our similarity tree. Each feature of the object will be placed around the object as a new node and will be linked to the object. The link is labeled according to fact frequency of the object extracted from the database. For example if the support for a fact is more than 50%, then it will be picked up to be placed around the object.

Figure 5-11 illustrates a sample similarity map for various animals. As it is seen, by replacing the nodes vertically, we find that animals are categorized based on their similarities. This is an interesting result since this final similarity map is constructed through collaboration of people and it is entirely based on similarity of the objects and the effect of using Prim's algorithm to extract the maximum spanning tree out of the similarity matrix of the objects.

Figure 5-11: Sample similarity tree showing classes of animals

This Similarity Map is very useful in depicting similarity clusters among the objects in the system. The user can control clusters by choosing specific subsets of features upon which similarity is calculated and the minimum similarity threshold.

5.9 Maximum Spanning Tree and Prim's Algorithm

A maximum spanning tree (MST) of a weighted graph is a tree-like connected sub graph that contains all of the nodes of the graph and has the maximum weight compared to all other possible sub graphs of the same graph. As we know, the similarity matrix which shows the similarity between each pair of objects is a complete graph. Each node is connected to all other nodes (vertices) and links (edges) have weights. This weight is the similarity of two nodes. Since in the SYGY^{Map} game the administrator or users may be looking for the sub graph that shows the most similar set of objects, we used the concept of MST to draw the maximum similarity sub graph.

The algorithm that we have used for creating this tree is Prim's algorithm since the administrator specifies a root for the tree and Prim's algorithm is also node-driven rather than edge-driven (like Kruskal's [64]). The algorithm is demonstrated in the following steps:

1. The initial node set is set to empty.
2. An arbitrary node is chosen as the root of the tree and added to the node set (the user chooses this node in our case).
3. Among all of the nodes that are linked to the nodes of the node set, the one which has the lightest link to the node set is added to it unless the edge creates a loop in the tree. (In our case, the node with the heaviest link to the node set is chosen). If there is more than one node satisfying the condition, an arbitrary node is chosen among qualified nodes. (In our case the node which either is most similar to the root or stands at the highest level is chosen).
4. Step 3 is repeated until all of the nodes are added to the node set.

As a resource, we have included the pseudo code for extracting Maximum spanning tree using Prim's algorithm.

Table 5-4: Pseudo code for Maximum spanning tree using Prim's Algorithm

```

Given a graph, G, with edges E of the form (v1, v2) and vertices V
dist : array of distances from the source to each vertex
edges: array indicating, for a given vertex, which vertex in the tree it
       is closest to
i : loop index
F: list of finished vertices
U : list or heap unfinished vertices

/* Initialization: set every distance to INFINITY until we discover a way to
link a vertex to the spanning tree */
for i = 0 to |V| - 1
    dist[i] = INFINITY
    edge[i] = NULL
end

pick a vertex, s, to be the seed for the minimum spanning tree

/* Since no edge is needed to add s to the minimum spanning tree, its distance
from the tree is 0 */

```

```

dist[s] = 0

while(F is missing a vertex)
    pick the vertex, v, in U with the shortest edge to the group of vertices in
    the spanning tree add v to F

    /* this loop looks through every neighbor of v and checks to see if that
    * neighbor could reach the minimum spanning tree more cheaply through v
    * than by linking through a previous vertex */
    for each edge of v, (v1, v2)
        if(length(v1, v2) < dist[v2])
            dist[v2] = length(v1, v2)
            edges[v2] = v1
            possibly update U, depending on implementation
        end if
    end for
end while

```

5.10 Applications of Visual Concept Maps

Numerous applications can be described for Visual Concept Maps. We highlight some of these practices here:

- Educational applications:** Perhaps the most outstanding application of the visual maps is its educational potential. As we mentioned earlier, the human brain is much more powerful in interpreting and understanding images comparing to verbal contexts. The VCMs would encourage learners to focus on the illustration of concepts and their connections. They are also handy resources for instructors in order to teach new concepts to small children. Children are amused while they create their own maps or even when they collaborate with each other to create one single map together. It occurs that deep learning happens when humans read VCMs since the visual architecture of the map is backed up in the long term portion of humans' brain.
- Knowledge representation:** Visual Maps are one of the best methods for representing knowledge to humans. We recognized that much of the research in the knowledge representation area is focused on the techniques and algorithms of

representing knowledge in computers and less attention is paid to humans. Organizing knowledge and putting it into well-formed structures improves human learning considerably. In fact, VCMs are representing the mixture of Visual illustrations and verbal contexts and this new mixture will reinforce human's perception and imagination about different concepts and their relations.

- **Knowledge transfer and sharing:** There have been always efforts for investigation of methods for transferring knowledge between two entities. For example transferring knowledge between humans and computers or computers and computers. Visual maps are excellent instruments for transferring knowledge between humans. They facilitate sharing what we have learned with each other. They are also powerful tools because they are not only representing a single mind but also consensus among tens or hundreds of volunteers.
- **Analytical and Scholarly applications:** Another important application of the VCMs is its usefulness in academic studies and scientific surveys. Researchers would benefit by reading those VCMs which are generated collaboratively to gauge the public understanding of subjects.
- **Visual Search Engines.** VCMs could be used to find relationships between a set of images for special purposes. There are also possibilities of developing visual search engines based on VCMs. The visual search engine enables users to click on images to find more about them. In this engine, the query itself is an image and the answers are those similar or dissimilar images to source query.

5.11 Enhancing the Learning Process

We believe that Visual Concept Maps can improve the learning process in several ways.

Acquisition of knowledge: Pictures can usually convey much more information than text, in the same amount of time and space available. For instance a child who has never seen an elephant would have difficulty in visualizing it no matter how good the textual description is. A picture would be far more effective.

Meaningfulness: The concept/similarity map is created through the participation of students/learners. This gives each student a sense of participation and ownership in the process of creation of the associations between pictures and the objects they represent.

The sense of peer participation is also likely to improve the depth of interest and attention span of younger children.

Modular learning: Since content can be divided into nodes and propositions, individuals can see the concepts faster and acquire specific knowledge more effectively by just looking at specific parts of the map and not having to cope with the whole map at one time.

There is still a lot of work required in investigating the effectiveness of VCMs in the learning process. Field experiments involving control groups would be required. However, such work is beyond the scope of this project which is primarily to provide the necessary tools and ideas for doing this.

5.12 Saving Visual Concept Maps

After successfully generating a visual concept map, we would need to save it in order to be used at suitable times or even we may need to export it to other applications. We studied different saving formats and we concluded that our format should be simple, consistence and extendable. It should be simple, since all other developers should understand it without spending too much time learning the complexities of our syntaxes. Consistency is also an important feature since it would help us to synchronize our application with the world of gadgets. Web 2.0 and emerging web technologies are advising us to be compatible with the global software development. In addition, extendibility is a priority. More and more features would be added to future versions so we may need to save different data types in addition to the original structure of concept map. This urges us to use extensible formats.

Looking at our requirements, we decided to use XML. It has all three features for us. XML is extendable, compatible and easy to read and write. There are a few ready syntaxes for exporting concept maps to XML files. For example, CXL is a publicly available XML-based language for describing the content of Cmaps [65]. CXL has a complex structure and keeps all details such as formatting and graphical representation syntaxes like fonts, colors, background images, borders and location of nodes on the screen (coordinates of x,y). In our project, we did not need all these details so we defined our own XML structure which is somehow similar to CXL. For the sake of simplicity we focused on the content of the map and left the graphical and formatting

information open to be defined by external developers. Everybody would use his own coloring and graphic design to present the visual concept map on the screen.

We are using our own namespace for defining elements of our generated XML. Main elements are:

- **VCM:** Starting and ending of concept map
- **Res-data:** includes primary information about the concept map. This information is about title of concept map, focus question of concept map, creators and date of creation.
- **Map:** this element is binding all structural elements for creating the concept map.
- **Concept-list:** This element includes all concepts in our concept map. Each concept has an ID, Label and resource. The ID is used for the connection-list. Resource is mainly an image.
- **Linking-phrase-list:** All the present links in the concept map are listed here. Some links would be used a few times between different pair of nodes. Each link has a label and ID.
- **Connection-list:** This element includes connections between each pair of nodes. Each connection has To, From and linking phrase ID. From-ID and To-ID are the IDs of concepts and they are also clarifying the direction of each link.
- **Resource-list:** this element contains a list of all resources used in the concept map. Almost all resources in our visual map are images. But resources could be texts or even sound and videos in future versions of visual maps.

For better understating of our XML file format, we have included the xml file content in Table 5-5 which is about a simple similarity map shown at Figure 5-12. The figure expressed the similarity of a few animals to a horse.

Figure 5-12: Similarity Map for similar animals to horse

Table 5-5: XML file Generated for sample Visual Concept Map of Animals

```

<?xml version="1.0" encoding="UTF-8"?>
<vcm xmlns="http://sygy.org/xml/vcm/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <!-- Primary information about concept map -->
  <res-meta>
    <!-- Map Title -->
    <dc:title>Similarity Map</dc:title>
    <!-- Focus Question -->
    <dc:description>How much animals are similar to Horse?</dc:description>
  </res-meta>
  <!-- Concept Map structure is defined by element: Map -->
  <map>
    <!-- List of Concepts (Nodes) in Visual Concept Map -->
    <concept-list>
      <!-- Each Concept has an ID and Label and Resource -->
      <concept id="001" label="Horse" resource-id="1001"/>
      <concept id="002" label="Zebra" resource-id="1002"/>
      <concept id="003" label="Goat" resource-id="1003"/>
      <concept id="004" label="Lion" resource-id="1004"/>
    </concept-list>
    <!-- Links between nodes have Label and ID -->
    <linking-phrase-list>
      <linking-phrase id="100" label="is Extremely Similar to"/>
      <linking-phrase id="101" label="is Partially Similar to"/>
      <linking-phrase id="102" label="is Scarcely Similar to"/>
    </linking-phrase-list>
    <!-- Connection list shows which nodes are connected to each other-->
    <connection-list>
      <connection from-id="002" to-id="001" Phrase-id="100"/>
      <connection from-id="003" to-id="001" Phrase-id="101"/>
      <connection from-id="004" to-id="001" Phrase-id="102"/>
    </connection-list>
    <!-- List of all resources such as Images, uri is an identifier to resource -->
    <resource-list>
      <resource id="1001" type="img" uri="http://sygy.org/photo/horse.jpg"/>
      <resource id="1002" type="img" uri="http://sygy.org/photo/zebra.jpg"/>
      <resource id="1003" type="img" uri="http://sygy.org/photo/goat.jpg"/>
      <resource id="1004" type="img" uri="http://sygy.org/photo/lion.jpg"/>
    </resource-list>
  </map>
</vcm>

```

Chapter 6 Case Studies for Evaluation

In this chapter, we will use sample case studies to examine the algorithms and procedures that we proposed in chapter 5 for generating visual concept maps. To build our cases, we assume that a teacher would like to use our system in order to teach sample topics of knowledge to young students in a primary school. In this case, the teacher would act as an administrator and the students have the role of volunteers in our system. We also assume that both parties (teacher and students) are familiar with the game mechanism. Since we do not have access to students at a primary school, we emulate the class of students with those volunteers who accepted our invitation to play SYGY^{TAG} and SYGY^{Map} during their free time. We play the role of the teacher to pick up the sample topics of study. We also configure both games. The topics for Similarity Maps are “*Vehicles*” and “*Mammals*” and the topic for Center-Based Visual Map is “*Fruits*”.

6.1 Center-Based Visual Concept Map for Fruits

Scenario: A teacher would like to assimilate knowledge about a wide range of fruits with students. The students are supposed to collaborate and share their own knowledge about the features of fruits through playing at SYGY games. At the final stage, a Center-Based visual map is generated. The students can explore the map to learn more about features of each fruit in the map. They may also discover which fruits are similar considering a specific feature.

Game statistics: A list of 35 fruits is prepared and at least one photo for each fruit is added to the image corpus. For some fruits which would have different possible colors in the universe, we tried to find and add the same fruits with verity of colors. For example three photos for red apple, green apple and yellow apple. 16 features were selected to make fact templates in SYGYMap. The list of fruits and features is shown in Table 6-1 and Table 6-2. The games were active for 12 days and a community of 100 volunteers played SYGY^{TAG} and SYGY^{Map} during this period. In average, each player spent almost 10 minutes playing SYGY^{TAG} and 25 minutes playing SYGY^{Map}.

Table 6-1: List of selected fruits

Apple	Apple	Apricot	Avocado	Banana	Blackberry	Cantaloup
-------	-------	---------	---------	--------	------------	-----------

Chapter 6 Case Studies for Evaluation

Cherry	Chestnut	Coconut	Cucumber	Orange	Grapefruit	Grape
Greengage	Hazelnut	Kiwi	Lemon	Lime	Mandarin	Mango
Medlar	Melon	Nectarine	Okra	Peach	Peanut	Pear
Persimmon	Pineapple	Plum	Pomegranate	Quince	Raspberry	Tomato

Table 6-2: List of selected 16 features for fruits

Taste/Shape/Touch	Sweet/ Sour/ Acrid/ Circular/ Hard/ Soft/ Having Peel
Color	Yellow/ Red/ Green/ Black/ Brown/ Orange/ Purple/ Violet/Pink

Fact Templates: Since the players should complete the fact sheets in SYGY^{Map} game, a fact template has to be configured by the teacher. A few sample facts which were used in SYGY^{Map} are:

<the Object> *is Sweet.* ☐YES ☐NO

<the Object> *is Circular.* ☐YES ☐NO

<the Object> *is Soft.* ☐YES ☐NO

<the Object> *is Red.* ☐YES ☐NO

Database of Facts: We have made queries to the fact database in order to extract the frequency of facts for each animal. A small portion of fact the frequency table is shown in Table 6-3. As it is seen in this table, each figure shows the number of players who have submitted a same fact for a specific fruit. For example, 37 of the players are saying that *Banana* is an *Acrid* fruit while remained 63 believe that *Banana* is not *Acrid*.

Table 6-3: Fact frequency for each fruit

Fruits ►	Apple	Apricot	Avocado	Banana	Blackberry	Cherry	Chestnut	Coconut	Cucumber
Features ▼									
Sweet	96	86	18	93	57	96	96	97	4
Not Sweet	4	14	82	7	43	4	4	3	96
Sour	4	15	6	1	62	17	0	0	2
Not Sour	96	85	94	99	38	83	100	100	98
Acrid	3	4	29	37	2	0	2	43	63
Not Acrid	97	96	71	63	98	100	98	57	37
Has Peel	96	97	96	99	1	21	100	100	97
Has no Peel	4	3	4	1	99	79	0	0	3
Circular	84	27	1	0	19	89	96	94	0
Not Circular	16	73	99	100	81	11	4	6	100
Hard	88	43	91	6	1	2	97	98	87
Not Hard	12	57	9	94	99	98	3	2	13
Soft	9	62	11	89	97	97	4	3	11
Not Soft	91	38	89	11	3	3	96	97	89
Yellow	2	17	97	98	1	1	91	0	0
Not Yellow	98	83	3	2	99	99	9	100	100

Generating Visual Map: To generate a center based visual Map, we will start from a central feature. In our case, we choose **Circular** as our central feature so all fruits which have this feature would be selected and placed around the center. Since we set the minimum support for fact frequency to 50%, those fruits with less than 50% support will be automatically removed from the map.

Figure 6-1 Center-based Visual Map for circular fruits

We draw the links between center and all fruits which are scattered around the center. For labeling the links, we refer to Table 5-2 to choose the proper adverb of frequency for each link. For example if support for the fact was 95%, we pickup “almost always” as our adverb. To complete the link text, we use “said to be” as a complementary expression since “circular” is a type of attribute for the fruits. In final stage, we insert the photos of each fruit in the map and put the labels on the links. The Center-Based Visual Map is ready. Figure 6-1 illustrates the results. As an example of the toggling feature, we chose “Greengage” as a new center. Three attributes of Greengage are expanded, and by choosing “Green”, the center is again a new feature and all fruits which are green will be selected. As we said in chapter 5, toggling could be done as many times as needed in order to explore the features of fruits. The generated map is actually reflecting the result of knowledge collaboration over the web.

6.2 Similarity Map for Mammals

Scenario: An elementary school teacher is trying to demonstrate the scientific classes of animals and the features upon which scientists carry out the classification. The teacher asks students to share their knowledge about a group of Mammals through playing SYGY Games. At the end, a Similarity Map would be generated which reveals the similarity of animals and their possible classification.

Game statistics: 12 Mammals were selected and a photo of each mammal was added to image corpus. As we need to measure the similarity of animals in order to classify them, we prepare a list of common and uncommon features of mammals. The list of animals and features are shown in Table 6-4 and Table 6-5. The games were run for almost 15 days and the total number of 100 contributors played the games in this period.

Table 6-4: List of Animals

Horse	Zebra	Donkey	Tiger	Lion	Leopard
Fox	Wolf	Dog	Sheep	Goat	Antelope

Table 6-5: List of Features

Attributes	Wild/Tame/Flocking/Carnivore/Herbivores/Warm-blooded/cold-blooded/Hunter/Swimmer/Nocturnal/Living on Land/Living in Water/Vertebrate/Invertebrate Climber/Runner
Characteristics	Having Fur/Having Hair/Having Tail/Having Mane Having Large Eyes/Having Small Eyes/Having Large Ears/Having Small Ears/Having Horns/Having Plain Skin/Having Patterned Skin

Figure 6-2: 12 Mammals selected for the games

Fact Templates: Each player is supposed to evaluate a list of facts while playing SYGY^{Map}. As we had taken the role of teacher, we prepared the template for the facts. Some of the facts on the template are:

- <the Object> *is Carnivore.* ☐YES ☐NO
- <the Object> *is Wild* ☐YES ☐NO
- <the Object> *is Tame* ☐YES ☐NO
- <the Object> *has tail* ☐YES ☐NO
- <the Object> *has mane* ☐YES ☐NO

Database of Facts: We have made queries to the fact database in order to extract the frequency of facts for each animal. A small portion of fact frequency table is shown in Table 6-6. As it is seen in this table, each figure shows the number of players who have submitted the same fact for a specific animal. For example, 27 of the players are saying that *Goat* is a *Wild* animal while remained 73 believe that *Goat* is not *Wild*.

Table 6-6: Fact frequency for animals												
Mammals ►	Horse	Zebra	Donkey	Tiger	Lion	Leopard	Fox	Wolf	Dog	Sheep	Goat	Antelope
Features ▼												
Wild	7	24	19	94	89	93	71	87	67	14	27	34
Not Wild	93	76	81	6	11	7	29	13	33	86	73	66
Flocking	66	55	23	12	39	19	28	32	15	87	82	74

Not Flocking	34	45	77	88	61	81	72	68	85	13	18	26
Carnivore	2	3	2	94	96	97	91	85	65	1	2	2
Not Carnivore	98	97	98	6	4	3	9	15	35	99	98	98
Herbivore	96	93	90	3	4	4	10	15	30	99	89	92
Not Herbivore	4	7	10	97	96	96	90	85	70	1	11	8
Nocturnal	10	18	7	85	89	92	84	95	64	21	24	27
Not Nocturnal	90	82	93	15	11	8	16	5	36	79	76	73
Has Mane	97	96	90	34	85	71	4	3	1	0	0	2
Has No Mane	3	4	10	66	15	29	96	97	99	100	100	98
Live on Land	100	100	100	98	100	100	100	100	100	100	100	100
Not Live on Land	0	0	0	2	0	0	0	0	0	0	0	0
Horns	1	0	1	1	1	2	1	0	0	0	92	95
No Horns	99	100	99	99	99	98	99	100	100	100	8	5

Table 6-7: Matrix of Similarity Map for Animals

	Horse	Zebra	Donkey	Tiger	Lion	Leopard	Fox	Wolf	Dog	Sheep	Goat	Antelope
Horse	100	89.1	90.7	49.7	55.7	49.1	58.1	58.6	60.3	67.8	68.6	70
Zebra	88.6	100	88.3	54.2	53	56.5	57.7	58.5	59.7	61.9	67	67.6
Donkey	90.1	88.2	100	51.6	57	51.7	59.8	59.4	63.1	65.3	67.4	67.6
Tiger	46.1	51.9	50.5	100	85.4	93	84	83.3	77.8	38.4	44.9	44.4
Lion	55.6	54.6	59.1	85.8	100	89.5	83	84.3	79.3	45	49.8	50.2
Leopard	47.3	56.2	52	93.1	89.1	100	80.3	81	76.1	37.6	44	43.9
Fox	53.6	54.9	58.1	83.9	83.5	81	100	93.9	87.8	49.7	57.4	56.7
Wolf	54.1	55.6	57.7	83.3	84.6	81.6	93.9	100	87.9	50.7	58.9	57.5
Dog	56.6	57.4	62.2	77.4	78.2	75.7	87.8	87.9	100	57.5	64.8	62.2
Sheep	66	61.9	65.6	39.7	44.5	39.1	53.7	54.8	61.1	100	85.5	83.6
Goat	66	65.1	66.9	45	47.8	43.4	58.8	60.2	65.4	82.7	100	94.4
Antelope	67.4	65.8	67.1	44.5	48.2	43.4	58.1	58.9	62.9	81.1	94.6	100

Generating Similarity Map: To generate the Similarity Map, we choose one of the objects as a root node. In this case we have chosen the “Horse”. In the next step, we will build the matrix of similarity using the similarity measurement formula which was proposed in section 5.6. The Similarity matrix shows the joint similarities between pairs of objects. The calculated matrix is shown in Table 6-7: Matrix of Similarity Map for Animals. This matrix is actually representing a complete graph between all animals. Figure 6-3 shows such graph. We will use Prim’s algorithm to extract the maximum spanning tree from our similarity graph. Each edge of our graph has a weight equal to similarity of both nodes which are connected through that edge. After finding the maximum spanning tree, we will start from the root node to construct the map. The root node (*Horse*) will be placed on the top of the map while other connected nodes to root

node will be placed below it. In our case, *Donkey*, *Zebra* and *Antelope* are those connected nodes to *Horse* node. We use the weight between each two nodes to decide about vertical distance between them. The nodes are actually sorted vertically in the map and we are free to choose the horizontal distance at our own preferences. Figure 6-4 shows the final Similarity Map. The distance between two objects is the reverse of the similarity. If the similarity is high, the distance will be short. As it is seen Figure 6-4 the similarity between *Horse* and *Donkey* is 90% which means the distance between them is 10 (100-90). The *Donkey* is placed 10 blocks (in grid page) below *Horse*. The *Antelope* is 69% similar to *Horse* so the distance would be 31 (100-69) and it is placed 31 blocks vertically below the horse. We emphasize again that horizontal distance is free to choose so one can place the nodes anywhere which fits best for clarity and better organization. After placing all the nodes on the map, we need to label the links between connected nodes. We use Table 5-3 to find the appropriate adverb of degree related to similarity between each two nodes. For example the adverb of degree for the link between “*Goat*” and “*Sheep*” would be “*Extremely*” since the similarity is 82%. The “*Similar to*” expression is also added to the adverb to make a meaningful label for the link between two animals. In final step, we replace the nodes with their visual depictions so we place the photos of animals in places of the nodes.

Figure 6-3: Complete graph for similarity of the objects

Figure 6-4: Similarity Map for animals

Categorization: Looking at the animals similarity map at Figure 6-4, we see that the map illustrates a hierarchy of animals in a vertical organization. By analyzing the map

one may identify that it is divided into 4 sections and each section consists of a group of animals which are members of a same class of animals. Constructing the matrix of similarities based on the features of animals and using Prims' algorithm to extract the maximum spanning tree are key techniques to obtain such an interesting map. We do not have to forget that placing the nodes vertically plays an important role for revealing the classification of objects based on their similarities. The art of Similarity Maps is their ability to expose the synergic output of a group of volunteers who each shared small pieces of knowledge while playing an online game.

6.3 Similarity Map for Vehicles

We performed a second study for generating the similarity map as we were interested to test our proposal to see the reliability of the results by repeating the procedure with a different topic. The results were prominent in both case studies.

Scenario: A teacher is willing to introduce some transportation vehicles to student at an elementary school. The teacher would like to use Similarity Maps to show the classification of vehicles based on the game results. The students are assigned to play SYGY games. At the end, a Similarity Map would be generated which reveals the similarity of vehicles.

Game statistics: 9 Vehicles were selected and a photo of each vehicle was added to the image corpus. We prepared the list of features which would help identify the common and distinct characteristics of the vehicles. List of Vehicles and the list of features are shown Table 6-8 and Table 6-9. The games were run on the internet for almost 15 days and total number of 100 contributors played the games.

Table 6-8: List of vehicles

Bicycle	Tricycle	Motorcycle	Automobile	Bus
Minibus	Van	Truck	Tanker	

Table 6-9: List of features for vehicles

Attributes	Gas Burner/ Transporting Loads/ Transporting Passengers/ Fast/ Slow/ Heavy/ Light
Characteristics	Having 2 Wheels/ Having 3 Wheels/ Having 4 Wheels/ Having More than 4 Wheels/ Having Engine/ Having Driver

Fact Templates: Each player is supposed to evaluate a list of facts while playing SYGY^{Map}. As we had taken the role of the teacher, we prepared the template for the facts. Some of the facts on the template are:

<the Object> *is Gas Burner*. ☐YES ☐NO

<the Object> *has Two Wheels* ☐YES ☐NO

<the Object> *has Driver* ☐YES ☐NO

<the Object> *has Engine* ☐YES ☐NO

<the Object> *is Fast* ☐YES ☐NO

Database of Facts: We made queries to the fact database in order to extract the frequency of facts for each vehicle. A small portion of fact frequency table is shown in Table 6-10. As it is seen in this table, each figure shows the number of players who have submitted a same fact for a specific vehicle. For example, 3 of the players are saying that *Bicycle Transports loads* while remained 97 believe that *Bicycle is not meant for transporting loads*.

Table 6-10: Fact frequency for vehicles

Vehicle ►	Bicycle	Tricycle	Motorcycle	Automobile	Bus	Minibus	Van	Truck	Tanker
Feature ▼									
Has 2 Wheels	98	3	97	3	2	3	2	1	0
Not 2 Wheels	2	97	3	97	98	97	98	99	100
Has 3 Wheels	3	96	1	2	1	1	0	0	0
Not 3 Wheels	97	4	99	98	99	99	100	100	100
Has 4 Wheels	1	1	1	98	4	30	93	3	2
Not 4 Wheels	99	99	99	2	96	70	7	97	98
More than 4 Wheels	0	0	0	1	97	84	8	99	100
Not More than 4 Wheels	100	100	100	99	3	16	92	1	0
Has Engine	1	1	97	99	100	100	99	100	100
No Engine	99	99	3	1	0	0	1	0	0
Is Gas Burner	3	2	96	98	99	100	99	100	100
Is Not Gas Burner	97	98	4	2	1	0	1	0	0
Transports Loads	3	4	2	12	23	18	67	94	97
Not for Load	97	96	98	88	77	82	33	6	3
Transports Passengers	9	7	39	90	96	97	35	7	4
Not for Passengers	91	93	41	10	4	3	65	93	96
Has Driver	98	97	98	97	99	98	98	97	99
No Driver	2	3	2	3	1	2	2	3	1
Is Heavy	9	14	23	50	78	61	54	97	96
Is Not Heavy	91	86	77	50	22	39	46	3	4
Is Light	90	87	76	43	24	34	38	3	2
Is Not Light	10	13	24	57	76	66	62	97	98
Is Fast	18	17	76	69	53	49	40	53	50
Is Not Fast	82	83	24	32	47	51	60	47	50

Generating Similarity Map: The procedure for generating the Similarity Map for vehicles is the same as the previous case which was about Animals. We skip over explaining the details and revert to Figure 6-5 which shows the final output. As it is seen in this figure, the Similarity Map is started by *Bicycle* as the root node. The Similarity graph is obtained using Prim's algorithm and the nodes are vertically sorted according to their distance with each other. As an example, *Automobile* is 68% similar to *Motorcycle* which means the distance between these two vehicles is 32 (100-68). Looking at Table 5-3, the related adverb of frequency is "***Moderately***". The Label for this link is "***Moderately similar to***" and the students can read the map following the link directions such as: "*The Motorcycle is Moderately Similar to the Automobile*".

Figure 6-5: Similarity Map for vehicles

Categorization: Looking at Vehicles similarity map at Figure 6-5, we identify that vehicles are classified into 3 classes. These three classes are shown by big numbers on

the map. The first class is all those vehicles which basically only transport one or two passengers and they have less than 4 wheels. We may call this class as Light Vehicles. The second class consists of those vehicles which mostly have engines and they are meant for transporting the passengers. The group of these vehicles could be called Semi-Heavy Vehicles. The last class is those Heavy Vehicles which are usually meant for transporting loads.

6.4 Expanded Similarity Maps

As a final proof of the efficiency of Visual concept maps, we have combined both Similarity Maps and Center-Based Maps in a single map to obtain a richer organization of knowledge. The final output consists of a similarity map in which each node acts as a center for a new center-based map. As it seen in Figure 6-6, a portion of Similarity Map for animals is selected to illustrate the main idea clearly. The selected nodes from the Similarity Map are *Horse*, *Donkey*, *Zebra* and *Antelope*. Using each of these nodes as a center for a new center based map, will result in a bigger map which reveals following information:

- Similarity percentile of each two Animals
- Specific features of each Animal according to consensus of contributors

In our sample case, we have just chosen a few features to be placed on the map. We also set the threshold for minimum fact frequency to 0% so all the facts have the equal chance to be placed on the map. Reading and exploring the concepts and links of this map would be interesting. For example, following statement is read through the map:

The Horse is extremely similar to the Zebra. Most people think that a Zebra is almost always a herbivore and no one thinks that a Zebra has horn.

In Figure 6-7, a portion of Similarity Map for Vehicles is expanded. As it is seen, the student can learn that Bicycle is extremely similar to Tricycle. If someone asks why Bicycle and Tricycle are so much similar, looking at features of both objects would reveal the answer:

- Both objects do not have Engine.
- Both objects are not Fast.
- Both objects do not burn Gas.
- Both objects have a driver.

- Both objects are not meant to Transport passengers.

Figure 6-6: Expanded Similarity Map for animals

Using the expanded similarity maps, would help students to learn about objects and related features in an efficient manner. The map lets the student compare the objects by reviewing the assigned features. They can judge on similarities based on the common and distinct characteristics of each object and this means that the maps are extremely useful for strengthening the ability of student to think logically based on evidences. The maps will also help to organize small pieces of knowledge in a network of nodes and

Chapter 6 Case Studies for Evaluation

links which would be beneficial in understanding the relation between concepts in much easier way.

Figure 6-7: Expanded Similarity Map for animals

Chapter 7 Conclusion and Future Work

The goal of this work was to define a clear vision “to acquire commonsense knowledge from contributors through online applications in order to generate visual concept maps.”

Our work was discussed in two main parts:

- Methods and procedures for collecting commonsense knowledge through online developed applications such as Collimator, SYGY^{TAG} and SYGY^{Map} (Chapters 2, 3 and 4 are covering the material)
- Processing and analyzing the collected knowledge in order to generate a collaborative visual concept map (Chapters 5 and 6)

By saying commonsense knowledge, we refer to the knowledge about location and name of the objects which are visible inside digital images. We believe that most people can easily identify and locate the objects while they are looking at a digital image and that is why this knowledge is said to be common among humans.

The first phase of our research was focused on the methods of annotating the digital images through the web. We designed and implemented a sophisticated online application called Collimator (Collaborative Image Annotator) in order to collect information about visible objects in the images. We were basically using a manual annotation method in which humans are involved in the process of annotating. Two main information elements were collected through Collimator: Label and Location for the objects. The annotator could use a drawing tool to draw a boundary around the identified object. Having the experience of launching and promoting Collimator, we understood that we needed more incentives for people to use our application. Looking at different research projects which were aimed at collecting knowledge through online application, we found that using Games would be useful to encourage people to share their knowledge while they enjoy playing an online game. We used our experience with Collimator to design a new game which was called SYGY (Syncretic Synergy). The game was divided into two counterpart games: SYGY^{TAG} and SYGY^{Map}. SYGY^{TAG} was aimed at collecting labels and locations of objects in the digital images while SYGY^{Map} was used to collect some facts about those objects which were labeled before in SYGY^{TAG}. Briefly, the output of SYGY^{TAG} was the input for SYGY^{Map}. The design and implementation of the games was a success in our research endeavor as the games received a lot of attention in the first few weeks of starting up.

In the second phase of our research, we focused on the methods and algorithms to use a database of SYGY games to generate sample Visual Concept Maps (VCM). We defined two models for visual maps which are: Similarity Maps and Center-Based Maps.

The similarity Map is actually a tree style concept map in which nodes are replaced by digital photos and links are labeled through processing the facts in database of SYGY^{Map}. The Similarity Map represents the similarity of a group of objects and tries to classify them according to their similarity measures. For classification, we used Prim's algorithm to pull out the maximum spanning tree out of a complete matrix of similarity. We also proposed a new formula for calculating the similarity between two objects based on their features in order to construct the matrix of similarity.

Center-Based Map as the name suggests, represents a concept map which has a center and all other concepts are scattered around and connected to the center. There are two types of nodes in the Center-Based Map: *Objects* and *Features*. If the Object is located in the center, then the features are around it which means the central object has those features. If the feature is in the center, then the objects would be placed around it which denotes all those objects have the same central feature. The links between the central node and each surrounding node is labeled according to results from the SYGY^{Map} game.

The visual maps represented in this work are actually demonstrating the synergic output of a group of volunteers who each shared small pieces of knowledge while playing an online game. We claim that Visual Maps have great potential to be used in teaching classes especially for young children as they get entertained playing games while they are sharing and learning new topics of knowledge. We also encourage teachers and instructors to use our developed methods to teach new topics in an interactive and collaborative environment. Using these methods enriches the learning experience during class times and makes all students involved in the learning process.

The main contribution of our research work would be summarized as:

- Developing Games and Online Applications as an innovative method for capturing human cycles over the web (SYGY and Collimator)
- Proposing new models for constructing collaborative concept maps (Similarity Maps and Center-Based Maps). The Similarity Map can be used for classifying the objects based on their attributes.

In our routine research schedule we also achieved following goals:

- 1- A new similarity measurement for gauging the resemblance of two objects based on their feature vector.
- 2- A new innovative technique for generating scores in an online game which is based on the common-consensus of the player.
- 3- Using Prim's algorithm to extract the maximum spanning tree out of the matrix of similarity and inserting nodes vertically to construct the similarity map which will expose the classification of the objects

We also explored the following research fields to furnish our literature reviews:

- Methods for Annotating Digital Images (Automatic, Semi-Automatic, Manual)
- Web 2.0 and collaborative tagging systems
- Using concept maps in a learning environment
- Bottlenecks in commonsense knowledge acquisition
- Games for harnessing human cycles (human computation)
- Methods and algorithms for measuring the similarity of objects based on the feature vector

We believe our approach to generate visual concept maps collaboratively is in line with efforts to enhance the learning experience. We also stress that humans tend to learn faster and easier while they work together and that is the key point to success for our collaborative tagging system. Using games to collect small pieces of knowledge is promising as they enhance the level and quality of contribution over time. Games are also extremely useful if the target community of users is young children. Converting the learning task into an entertaining job is proven to be efficient for children.

We believe there is an immense amount of work to be done in the continued development of our research project. Some of the work which could be done in the future are:

- Extending the SYGY game to collect different types of knowledge. Such as relation between objects in digital images or relevancy of images to different topics of knowledge.
- Discovering methods for using ontologies in order to label digital images semantically.
- Proposing new models for constructing visual concept maps
- Studying different classification methods and mixing the idea of Similarity maps with other algorithms of classification to earn new results

- A new search engine could be developed based on the idea of classifying objects using Similarity Maps. One may search for most related or similar objects to one base object. This base object is the query term for the search engine (the keyword).
- Using different algorithms to explore the matrix of similarity in order to extract new models of knowledge
- Improving the fact templates in SYGY^{Map}

One of the main areas for future work is the study of the quality and effectiveness of a visual concept map. There are several factors for measuring the quality of visual concept maps for enhancing the learning process. For example, is it easy to read and understand? These kinds of questions require field studies where different users are engaged to read a map and then do a comprehension test or write a brief essay about the map. Such studies allow students to exercise their imagination which is a component often missing in the learning process.

During our research, we have received several proposals and ideas which could be interesting to explore and develop. Some of them are:

- Since the location of the objects is identified by the player in SYGY^{TAG}, the database of objects could be used in the future as a training set for pattern recognition.
- The Visual Maps are useful for representing the voting results in different fields. For example a company would ask customers to vote on the quality of its products through a game. The visual map helps the customers to compare and realize the differences between different products much easier.
- The SYGY game could be used for learning new languages. Since the objects are labeled in English, we may translate it to other languages using different dictionaries. The learner can look at images and click on objects to see what the object is called in the target language.

We hope the presented methods and techniques in this thesis will bring forward new opportunities to researchers who believe in concept mapping as a tool for transferring knowledge. We also encourage the researchers to evaluate, expand and enhance our proposed methods to generate more prolific results in future.

Publications and Lectures

- [1] Alireza Kashian, Robert Kheng Leng Gay, Abdul Karim, “*Collaborative Concept Tagging for Images based on Ontological Thinking*”, in International Conference on Image and Video Retrieval. July 13-15, 2006: Arizona State University , USA. p. 521-524.
- [2] Alireza Kashian, Robert Kheng Leng Gay, Mehdi Milanifard, Hashem Tatari, “*Collimator – Collaborative Image Annotator & Visual Concept Generator*”, in The 5th International Semantic Web Conference, Challenge 06. November 2006: Athens, University of Georgia, USA.
- [3] Alireza Kashian, M.R., Habib Mir M. Hosseini, and Robert Kheng Leng Gay, “*Commonsense Measurement Through Gaussian Interpolation of Tags for an Image Labeling Game*”, in The 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing. June 2007: Las Vegas, Nevada, USA.
- [4] Lecture, Semantic Web & Intelligent Life, 2005, IT Management and National Development,
- [5] Map Annotation and Semantic GIS Services, 2006, Iran Telecom Research Center

References

1. Novak J. D., J.W., *Coeditors, special issue on concept mapping*. Journal of Research in Science Teaching, 1991. **28**(10).
2. Quinn J. J, H.J., Mintzes, R. A. Laws, *Successive concept mapping. Assessing understanding in college science classes*. Journal of College Science Teaching, December/January 2003. **33**(3): p. 12 – 16.
3. Novak, J.D., *Concept maps and vee diagrams: Two metacognitive tools for science and mathematics education*. Instructional Science, 1990. **19**: p. 29-52.
4. Ausubel, D.P., *A Cognitive View*. Educational Psychology, 1963.
5. Gahr, A.A., *Cooperative Chemistry. Concept mapping in the organic chemistry lab*. Journal of College Science Teaching, February 2003. **32**(5): p. 311 – 315.
6. Joseph D. Novak, A.J. Cañas., *The Theory Underlying Concept Maps and How to Construct and Use Them*, in *Technical Report IHMC CmapTools*. 2006, Florida Institute for Human and Machine Cognition.
7. Zeilik, M., *Concept Mapping tutorial*, in *Department of Physics & Astronomy*, University of New Mexico.
8. E. Bruillard, G.-L.B. *Computer-Based Concept Mapping: a Review of a Cognitive Tool for Students*. in *Proceedings of Conference on Educational Uses of Information and Communication Technologies (ICEUT 2000)*. 2000. Beijing: Publishing House of Electronics Industry (PHEI).
9. A. O'Donnell, D.D., R. H. Hall, *Knowledge maps as scaffolds for cognitive processing*. Educational Psychology Review, 2002. **14**: p. 71-86.
10. C.D. Holley, D.F.D., *The development of spatial learning strategies*, in *Spatial learning strategies. Techniques, applications, and related issues*. 1984, Academic Press: New York. p. 3-19.

References

11. J. D Novak, D.M., *A twelve-year longitudinal study of science concept learning*. American Educational Research Journal, , 1991. **28**(1): p. 117-153.
12. J. D. Novak, D.B.G., *Learning how to learn*. 1984, Cambridge University Press: New York, NY.
13. J. Bransford, A.L.B., R. R. Cocking, *How people learn: Brain, mind, experience, and school*. 1999, Washington, D.C.: National Academy Press.
14. *Flickr*. [cited; Available from: <http://www.flickr.com>.
15. *Delicious*. [cited; Available from: <http://del.icio.us>.
16. A Scott , G., A Bernardo, Huberman *The Structure of Collaborative Tagging Systems*, in *Information Dynamics Lab, HP Labs*. 2005.
17. LiuWenyin1, S.D., Yanfeng Sun, HongJiang Zhang, *Semi-Automatic Image Annotation*, Microsoft Research.
18. *Wikipedia. Definition: Automatic image annotation*. [cited; Available from: http://en.wikipedia.org/wiki/Automatic_image_annotation].
19. Wei Li, M.S. *Automatic Image Annotation Based on WordNet and Hierarchical Ensembles*, *Computational Linguistics and Intelligent Text Processing*. in *7th International Conference, CICLing 2006*.
20. Kosorukoff, A., *Human-based Genetic Algorithm*. *IEEE Transactions on Systems, Man, and Cybernetics*. 2001: SMC-2001. 3464-3469.
21. L. Brandt, J.E., J. Hellemans, L. Heerman, I. Couwenberg, L. Volckaert, H. J. Morisse, J. J. Quinn, R. A. Laws Mintzes, *The impact of concept mapping and visualization on the learning of secondaryschool chemistry students*. *International Journal of Science Education*, 2001. **23**(12): p. 303-1313.
22. L.H. Hill, M.M.R. *Using Visual Concept Mapping to Communicate Medication Information to Chronic Disease Patients with Low Health Literacy*. in *Proceedings of the first international conference on concept mapping*. 2004. , Pamplona, Spain.

References

23. Luis von Ahn, M.K.a.M.B., *Verbosity: A Game for Collecting Common-Sense Facts*, Computer Science Department, Carnegie Mellon University: Pittsburgh PA.
24. Alireza Kashian, R.K.L.G., Mehdi Milanifard, Hashem Tatari, *Collimator – Collaborative Image Annotator & Visual Concept Generator*, in *The 5th International Semantic Web Conference, Challenge 06*. November 2006: Athens, University of Georgia, USA.
25. H. Lieberman, H.L., P. Singh, and B. Barry, *Beating common sense into interactive applications*. AI Magazine, Winter 2007. **25**(4): p. 63-76.
26. D.B.Lenat, *CYC: A Large-Scale Investment in Knowledge Infrastructure*. Communications of the ACM, 1995. **38**(4): p. 28-32.
27. Sowa, J. *Ontology*. [cited; Available from: <http://www.jfsowa.com/ontology/>].
28. *Ontology and ArtificialIntelligence*. [cited; Available from: http://ontologyworks.com/what_is_ontology.php].
29. H.Lieberman, D.A.S., A. Teeters, (*liberman-20*) *Common Consensus: a web-based game for collecting commonsense goals*, in *IUI'07*. 2007: Hawaii, USA.
30. P.Singh, T.L., E. Mueller, G. Lim, T. Perkins, and W. L. Zhu. *Open mind common sense: Knowledge acquisition from the general public*. in *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. Lecture Notes in Computer Science*. 2002. Heidelberg: Springer-Verlag.
31. Turbane, E. *EXPERT SYSTEMS AND APPLIED ARTIFICIAL INTELLIGENCE*. September 1993 [cited; Available from: <http://www.scism.sbu.ac.uk/inmandw/review/knowacq/review/rev11656.html>].
32. *Visual Resources Association Data Standards Committee*. [cited; Available from: <http://www.vraweb.org/vracore3.htm>].
33. *A part-of-speech tagger for English*. [cited; Available from: <http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/>].

References

34. *Google Web Toolkit Product Overview*. [cited; Available from:
<http://code.google.com/webtoolkit/overview.html>.
35. Alireza Kashian, R.K.L.G., Abdul Karim, *Collaborative Concept Tagging for Images based on Ontological Thinking*, in *International Conference on Image and Video Retrieval*. July 13-15, 2006: Arizona State University, USA. p. 521-524.
36. *Wikipedia. Definition: Collimator*. [cited; Available from:
<http://en.wikipedia.org/wiki/Collimator>].
37. *Collimator project*. [cited; Available from: <http://www.xmlweb.info/collimator/>].
38. Cardwell, L. (2005) *AJAX – Bridging the Thin-Client Performance Gap*. **Volume**,
39. *Adobe SVG Viewer*. [cited; Available from:
<http://www.adobe.com/svg/viewer/install/main.html>.
40. Turing, A.M., *Computing machinery and intelligence*. 1936.
41. *Large lexical database of English*. [cited; Available from:
<http://wordnet.princeton.edu/>
42. Hendler, C.H.-W.a.J.G.a.A.S.a.M.G.a.B.P.a.J., "*Photostuff - an image annotation tool for the semantic web*", in *4th international semantic web conference*. 2005.
43. Alireza Kashian, M.R., Habib Mir M. Hosseini, and Robert Kheng Leng Gay, *Commonsense Measurement Through Gaussian Interpolation of Tags for an Image Labeling Game*, in *The 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing*. June 2007: Las Vegas, Nevada, USA.
44. Kashian, A., *SYGY: New Generation of Online Collaborative Games*, in *ITzone*. April 2007, Nanyang Technological University: Singapore.
45. M. M. Bongard, *Pattern Recognition*. 1970, Rochelle Park NJ: Spartan Books.
46. B.C.Russell, A.T., K.P Murphy, W.T. Freeman, *LabelMe: a database and web-based tool for image annotation*. September, 2005, MIT AI Lab Memo AIM-2005-025.

References

47. *Wikipedia. Definition: Game.* [cited; Available from:
<http://en.wikipedia.org/wiki/Game>.
48. *Google Image Labeler .* [cited; Available from:
<http://images.google.com/imagelabeler/>].
49. Sniderman, S. *Fair Game.* [cited; Available from:
<http://www.gamepuzzles.com/tlog/tlog25.htm>].
50. *ESP Game.* [cited; Available from: <http://www.espgamec.org/>].
51. *Mindpixel Project.* [cited; Available from: <http://www.mindpixel.org>.
52. [cited; Available from: <http://kurzweilai.net>.
53. *OpenMind project.* [cited; Available from: <http://www.openmind.org>.
54. *CMapTools: An application or constructing and sharing concept maps* [cited;
Available from: <http://cmap.ihmc.us/>.
55. Derbentseva, e.a. *Concept Maps: Theory, Methodology, Technology.* in *Proc. of the Second Int. Conference on Concept Mapping.* 2006. San José, Costa Rica.
56. *Wikipedia. Ugly duckling theorem.* [cited; Available from:
http://en.wikipedia.org/wiki/Ugly_duckling_theorem.
57. *Minkowski distance.* [cited; Available from:
<http://en.wikipedia.org/wiki/Distance>.
58. Richard O. Duda, P.E.H., David G. Stork, *Pattern Classification* 2nd ed. 2000,
New York: John Wiley & Sons, Inc.
59. Goldstone, R., *chapter Similarity*, in *The MIT Encyclopedia of the Cognitive Sciences (MITECS)*. 1999, MIT Press: Cambridge, MA. p. 763-765.
60. A. Tversky, I.G., *Similarity, separability, and the triangle inequality.*
Psychological Review, 1982. **89**(2): p. 123-154.
61. Tversky, A., *Features of similarity.* *Psychological Review*, 1977. **84**(4): p. 327-352.

References

62. Melara, R.D., *The concept of perceptual similarity: From psychophysics to cognitive psychology*. psychophysical Approaches to Cognition, 1992: p. 303-388.
63. C. C. Aggarwal, A.H., D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space* in *ICDT 2001: 8th International Conference on Database Theory*. p. 420-434.
64. *Kruskal's algorithm*. [cited; Available from:
http://en.wikipedia.org/wiki/Kruskal's_algorithm.
65. HJ Irvine, K.C., G Jones *Concept mapping to enhance student learning in a financial accounting subject*. in *the Accounting Educators Forum*. 24-25 November 2005. Sydney.

Bibliography

1. *Computer Vision Online Publications*. [cited; Available from: http://www-2.cs.cmu.edu/_cil/v-pubs.html].
2. *PhotoStuff, Image Annotation Tool for SW*. [cited; Available from: <http://www.mindswap.org/2003/PhotoStuff/>].
3. *Image Annotation on the Semantic Web*. [cited; Available from: <http://www.w3.org/TR/2006/WD-swbp-image-annotation-20060322/>].
4. [cited; Available from: <http://www.xmlweb.info/collimator/>].
5. A. Bashir, a.L.K. *A Framework for Image Annotation Using Semantic Web*. in *ACM SIGKDD First International Workshop on Mining for and from the Semantic Web* August 2004. Seattle, Washington.
6. Ahmed Id Oumohmed, M.M., Jian-Yun Nie. *Semantic-Based Cross-Media Image Retrieval*. in *Pattern Recognition and Image Analysis: Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Proceedings, Semantic-Based Cross-Media Image Retrieval*.
7. Alan Dix, S.L., Alessio Malizia, *Semantic Halo for Collaboration Tagging Systems*, in *at Workshop on the Social Navigation and Community-Based Adaptation Technologies In Conjunction with Adaptive Hypermedia and Adaptive Web-Based Systems 2006*.
8. Alberto J. Cañas, R.C., Greg Hill, Marco Carvalho, Marco Arguedas, Thomas C. Eskridge, James Lott, Rodrigo Carvajal, *Concept Maps: Integrating Knowledge and Information Visualization*, in *Lecture Notes in Computer Science*. p. 205.
9. Andreas D. Lattner, A.M., Otthein Herzog. *A Combination of Machine Learning and Image Processing Technologies for the Classification of Image Regions*. in *Adaptive Multimedia Retrieval: First International Workshop, AMR 2003, Proceedings, Lecture Notes in Computer Science*.
10. Antti Karvonen, E.R., Jorma Tarhio, Jari Turkia, *Versatile Concept Map Viewing on the Web*. ACM SIGCSE Bulletin, 2001. 33(3): p. 105-108.
11. Chen-Yu Lee, V.-W.S., Yi-Ting Fu. *An Image Annotation Guide Agent*. in *Intelligent Agents and Multi-Agent Systems: 7th Pacific Rim International Workshop on Multi-Agents, PRIMA 2004, Proceedings, Lecture Notes in Computer Science*.
12. Chiung-Hui Chiu, C.-C.H., Wen-Tsung Chang, Tsung-Ho Liang. *Interaction processes in network supported collaborative concept mapping*. in *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*. 1999: ACM SIGCSE Bulletin.
13. Chu-Hong Hoi, M.R.L. *Web image learning for searching semantic concepts in image databases*. in *13th international World Wide Web conference on Alternate track Proceedings*. 2004.
14. Dadong Wan, P.M.J. *Computer supported collaborative learning using CLARE: the approach and experimental findings*. in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. 1994.
15. Donald Metzler, R.M., *An Inference Network Approach to Image Retrieval*, in *Image and Video Retrieval: Third International Conference, CIVR 2004, Lecture Notes in Computer Science*. p. 42-50.

Bibliography

16. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., and Tomkins, A. *Visualizing tags over time*. in *In Proceedings of the 15th international Conference on World Wide Web*, (Edinburgh, Scotland, May 23 - 26, 2006). *WWW '06*. ACM Press, New York, NY. 2006.
17. Dymrna O'Sullivan, E.M., Michela Bertolotto, David Wilson. in *Modeling and Using Context: 5th International and Interdisciplinary Conference CONTEXT 2005, Proceedings, Lecture Notes in Computer Science*.
18. E. Hyvönen, S.S.a.K.V. *Intelligent Image Retrieval and Browsing Using Semantic Web Techniques- A Case Study*. in *International SEPIA Conference at the Finnish Museum of Photography*. September 2003. Helsinki.
19. Eero Hyvonen, S.S., and Kim Viljanen, *Intelligent Image Retrieval and Browsing Using*, in the *SEPIA Conference*. 2003: Helsinki.
20. Elisa Bertino , B.C., Gian Piero Zarri, *A conceptual annotation approach to indexing in a Web-based information system*. Vol. 2. 2000: NETNOMICS. p. 247-264.
21. Elizabeth Santhanam, C.L., Chris Dawson, *Concept mapping: How should it be introduced, and is there evidence for long term benefit?*, in *Higher Education*. 1998. p. 317-328.
22. F. C. Pereira, A.C.O.a.A.C., *Extracting Concept Maps with Clouds*, in *Procs. of Argentinean Symposium on Artificial Intelligence*. 2000.
23. Ferguson, E., *Object-oriented concept mapping using UML class diagrams*. *Journal of Computing Sciences in Colleges*, 2003. **18**(4): p. 344-354.
24. Francisco Câmara Pereira, A.C. *Clouds: A Module for Automatic Learning of Concept Maps*. in *Theory and Application of Diagrams: First International Conference, Diagrams 2000, Proceedings*.
25. Fu, C.-Y.L.V.-W.S.Y.-T. *An image annotation guide agent*. in *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*. 2004.
26. George W. Furnas, C.F., Luis von Ahn, Joshua Schachter, Scott Golder, Kevin Fox, Marc Davis, Cameron Marlow, Mor Naaman, *Why do tagging systems work?*, in *CHI '06 extended abstracts on Human factors in computing systems CHI 2006*.
27. Gerhard Weikum, J.G., Ralf Schenkel, Martin Theobald. *Towards a Statistically Semantic Web*. in *Conceptual Modeling – ER 2004: 23rd International Conference on Conceptual Modeling, Lecture Notes in Computer Science, Proceedings*.
28. Hieu T. Nguyen, A.S. *Everything Gets Better All the Time, Apart from the Amount of Data*. in *Image and Video Retrieval: Third International Conference, CIVR 2004, Proceedings, Lecture Notes in Computer Science*.
29. Hong-Gee Kim, C.F., Bob Smith, Dietmar Wikarski. *Visualizing a Dynamic Knowledge Map Using Semantic Web Technology*. in *Engineering and Deployment of Cooperative Information Systems: First International Conference, EDCIS 2002, Proceedings, Lecture Notes in Computer Science*.
30. Huberman, S.A.G.a.B.A., *The Structure of Collaborative Tagging Systems*, in *Information Dynamics Lab, HP Labs*. 2005.
31. Hugo Liu, H.L. *Robust photo retrieval using world semantics*. in *Proceedings of the 3rd International Conference on Language Resources And Evaluation Workshop: Creating and Using Semantics for Information Retrieval and Filtering: State-of-the-art and Future Research*. 2002. Canary Islands.
32. Hui Xu, H.X. *An Image Retrieval System Based on MPEG-7 and XMLDB Query for Digital Museum*. in *Technologies for E-Learning and Digital Entertainment*:

- First International Conference 2006, Proceedings, Lecture Notes in Computer Science.*
33. Ilkka Autio, T.E., *Flexible view recognition for indoor navigation based on Gabor filters and support vector machines*, in *Pattern Recognition 2003*. p. 2769-1779.
 34. JB, T., *Bayesian modeling of human concept learning*, in *Advances in Neural Information Processing Systems*. 1999, Cambridge, MA: MIT Press.
 35. Jianye Ye, X.Z., Jian Pei, Lian Chen, Liang Zhang. *A Stratification-Based Approach to Accurate and Fast Image Annotation*. in *Advances in Web-Age Information Management: 6th International Conference, Proceedings, WAIM 2005, Lecture Notes in Computer Science*.
 36. Jivoon Joen, R.M. *Using Maximum Entropy for Automatic Image Annotation*. in *Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, July 21-23, 2004. Proceedings, Lecture Notes in Computer Science*.
 37. Junho Choi, M.C., Se Hyun Park, Pankoo Kim. *Concept-Based Image Retrieval Using the New Semantic Similarity Measurement*. in *Computational Science and Its Applications – ICCSA 2003, Proceedings, Concept-Based Image Retrieval Using the New Semantic Similarity Measurement*.
 38. Kevin Hartman, K.P.B., *The Unless Switch: Adding Conditional Logic to Concept Mapping for Middle School Students*, in *CHI '05 extended abstracts on Human factors in computing systems. Late breaking results: short papers*. p. 1439 - 1442.
 39. Khan, L., *Translating Images to Keywords: Problems, Applications and Progress*, *Advances in Multimedia Information Systems*. 11th International Workshop, MIS 2005, Proceedings. Vol. 3665. p. 4.
 40. Knoware, G.R., *a concept map based collaborative virtual environment*. ACM SIGGROUP Bulletin, 2000. 21(1): p. 14-15.
 41. Kobus Barnard, P.D., David Forsyth, Nando de Freitas, David M. Blei, Michael I. Jordan, *Matching Words and Pictures*. In *Journal of Machine Learning Research*. 2003. p. 1107-1135.
 42. Kok F. Lai, H.Z., Syin Chan. *Query expansion by raw image features and text annotations in image retrieval*. in *ACCV'98: Third Asian Conference on Computer Vision, Proceedings, Lecture Notes in Computer Science*.
 43. L. Hollink, G.S., J. Wiclemaker, and B. Wielinga. *Semantic annotation of image collections*. in *Proceedings Knowledge Markup and Semantic Annotation Workshop*. October 2003.
 44. L. Hollink, S.L., J. Hunter. *Evaluating the Application of Semantic Inferencing Rules to Image in Proceedings of the 3rd international conference on Knowledge capture, Information retrieval*.
 45. Lambert Schomaker, E.d.L., Louis Vuurpiji. *Using Pen-Based Outlines for Object-Based Annotation and Image-Based Queries*. in *Visual Information and Information Systems: Third International Conference, VISUAL 1999, Proceedings, Lecture Notes in Computer Science*.
 46. Lambiotte, R., Ausloos, M, *Collaborative tagging as a tripartite network*. Arxiv. 2005.
 47. Lars Grunewaldt, K.M., Karsten Morisse. *Workflow for Integrated Object Detection in Collaborative Video Annotation Environments*. in *Computational Science – ICCS 2006: 6th International Conference, Proceedings, Lecture Notes in Computer Science*.

48. mann, T.H. *A hierarchical mixture model for word occurrence document in document databases*. in *workshop on learning from text and the web, CMU*.
49. Marcos, G.S., Tim; Jiménez, Iván; Posada, Jorge; Stork, André; Pianciamore, Massimiliano; Castro, Rui; Di Marca, Sergio; Mauri, Marco; Selvini, Paolo; Sevilmis, Neyir; Thelen, Bruno; Zecchino, Vincenzo. *A Semantic Web based approach to multimedia retrieval*. in *Fourth International Workshop on Content-Based Multimedia Indexing* 21-23 June 2005. Riga, Latvia.
50. Markus Koskela, J.L. *Semantic Annotation of Image Groups with Self-organizing Maps*. in *Image and Video Retrieval: 4th International Conference, CIVR 2005, Proceedings, Lecture Notes in Computer Science*.
51. Marshall, B.a.M. *Element Matching in Concept Maps*. in *Proceedings Joint Conference on Digital Libraries*. 2004. Tucson, AZ.
52. Mathes, A. *Folksonomies – Cooperative Classification and Communication Through Shared Metadata*. 2004 [cited; Available from: Adam Mathes.com.
53. Matt Southern, J.T. *Creating a collaborative learning environment for children*. in *Proceeding of the 2003 conference on Interaction design and children*.
54. McCulloch, G.M.a.E., *Collaborative Tagging as a Knowledge Organisation and Resource Discovery Tool*. *Library Review*. 2006: Library Review.
55. Michael S. Lew, N.S., John P. Eakins. *Challenges of Image and Video Retrieval*. in *Image and Video Retrieval : International Conference, CIVR 2002, Proceedings, Lecture Notes in Computer Science*.
56. Milićević, M.T.a.V. *The Semantics of Collaborative Tagging System*. in *2nd International Workshop on Scripting for the Semantic Web Proceedings*. 2006.
57. Milićević, M.T.a.V. *The Semantics of Collaborative Tagging System*. in *2nd International Workshop on Scripting for the Semantic Web proceeding*. 2006.
58. Ming Li, T.W., Bao-wei Zhang, Bi-Cheng Ye. *A Novel Method of Image Retrieval Based on Combination of Semantic and Visual Features*. in *Fuzzy Systems and Knowledge Discovery: Second International Conference, FSKD 2005, Proceedings, Lecture Notes in Computer Science*.
59. Mishne, G. *AutoTag: a collaborative approach to automated tag assignment for weblog posts*. in *15th International World Wide Web Conference (WWW2006), 2006 Proceedings of the 15th international conference on World Wide Web WWW '06*.
60. Nicolas Maillot, M.T. *A Weakly Supervised Approach for Semantic Image Indexing and Retrieval*. in *Image and Video Retrieval: 4th International Conference, CIVR 2005, Proceedings, Lecture Notes in Computer Science*.
61. Oge Marques, N.B., *Semi-automatic Semantic Annotation of Images Using Machine Learning Techniques*. *The SemanticWeb - ISWC 2003, Lecture Notes in Computer Science*. Vol. 2870. p. 550-565.
62. Oleg Starostenko, A.C.-A., J. Alfredo Sánchez, Yulia Ostróvskaya. *A Hybrid Approach for Image Retrieval with Ontological Content-Based Indexing*. in *Progress in Pattern Recognition, Image Analysis and Applications: 10th Iberoamerican Congress on Pattern Recognition, CIARP 2005, Proceedings*.
63. Pat Hayes, T.C.E., Raul Saavedra, Thomas Reichherzer, Mala Mehrotra, Dmitri Bobrovnikoff. *Collaborative knowledge capture in ontologies*. in *Proceedings of the 3rd international conference on Knowledge capture*. 2005.
64. Peter G.B. Enser, C.J.S., Paul H. Lewis. *Automatic Annotation of Images from the Practitioner Perspective*. in *Image and Video Retrieval: 4th International Conference, CIVR 2005, Proceedings, Lecture Notes in Computer Science*.

Bibliography

65. Philipp Cimiano, A.H., Steffen Staab, *Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis*. November 2004, Technical Report , Insitute AIFB, University of Karlsruhe.
66. Postel, R.D.C.a.G.C., *Image Annotation with Adobe Photoshop*. Journal of Digital Imaging. **15**: p. 197-202.
67. Pu-Jen Cheng, L.-F.C. *Effective image annotation for searches using multilevel semantics*. in *International Journal on Digital Libraries, Proceedings, International Journal on Digital Libraries*.
68. R. Huang, S.L.a.F.F., ImageSpace. *An Image Ontology Creation and Annotation Tool*. in *Proceedings of the 19th International Conference on Computers and Their Applications (CATA'2004)*. 2004. Seattle, WA, USA.
69. Rob Kremer, B.R.G. *concept mapping techniques*. in *Proceedings of the 12th annual international conference on Systems documentation: technical communications at the great divide, ACM Special Interest Group for Design of Communications*.
70. Ruth Cobos, J.S. *Collaborative Knowledge Transfer by Annotating Documents*. in *Practical Aspects of Knowledge Management: 5th International Conference, Proceedings, Lecture Notes in Computer Science*. 2004.
71. Rye, J.A., *Enhancing Teachers' Use of Technology Through Professional Development on Electronic Concept Mapping*. Journal of Science Education and Technology, 2001. **10**: p. 223-235.
72. SeungMin Rho, S.L., EenJun Hwang, YangKyo Lee. *XCRAB: A Content and Annotation-Based Multimedia Indexing and Retrieval System*. in *Computational Science and Its Applications - ICCSA 2004, Proceedings, Lecture Notes in Computer Science*.
73. Stell, J., and Worboys, M, *Stratified map spaces: A formal basis for multi-resolution spatial databases*, in *In Proc. of the Int. Symp. on Spatial Data Handling*. 1998. p. 180-189.
74. Stephan Bloehdorn, K.P., Carsten Saathoff, Nikos Simou, Vassilis Tzouvaras, Yannis Avrithis, Siegfried Handschuh, Yiannis Kompatsiaris, Steffen Staab and Michael G. Strintzis. *Semantic Annotation of Images and Videos for Multimedia Analysis*. in *Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29–June 1, 2005. Proceedings, Lecture Notes in Computer Science*.
75. Stephan Bloehdorn, P.C., Andreas Hotho, *Learning Ontologies to Improve Text Clustering and Classification*. 2005: Proc of GFKL.
76. T. Athanasiadis, V.T., K. Petridis, F. Precioso, Y. Avrithis and I. Kompatsiaris. *Using a Multimedia Ontology Infrastructure for Semantic Annotation of Multimedia Content*. in *In Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005) at the 4th International Semantic Web Conference*. 2005.
77. Tergan, S.-O., *Digital Concept Maps for Managing Knowledge and Information*, in *Lecture Notes in Computer Science*. 2005. p. 185.
78. Th. Hermes, A.M., O. Herzog, *Graphical Search for Images by PictureFinder*. p. 229- 250.
79. Timo Volkmer, J.R.S., Apostol (Paul) Natsev. *A web-based system for collaborative annotation of large image and video collections*. in *Proceedings of the 13th annual ACM international conference on Multimedia 2005*.
80. V. Lavrenko, R.M., J. Leon: *A Model for Learning the Semantics of Pictures*. in *Preproceedings of NIPS 2003*.

Bibliography

81. Vat, K.H., *Web-based asynchronous support for collaborative learning*. Journal of Computing Sciences in Colleges, 2001. 17(2): p. 326-344.
82. Vlad Tanasescu, J.D. *A Semantic Google Maps Based Emergency Management Graphical User Interface*. in *Proceedings of the 2nd International Workshop on Scripting for the Semantic Web*. 2006.
83. Völkel, M., Krötzsch, M., Vrandečić, D., Haller, H., and Studer, R. *Semantic Wikipedia*. in *In Proceedings of the 15th international Conference on World Wide Web 2006*. New York, NY: ACM Press.
84. Wei Li, M.S. *Automatic Image Annotation Using Maximum Entropy Model*. in *Natural Language Processing – IJCNLP 2005: Second International Joint Conference, Proceedings, Lecture Notes in Computer Science*.
85. Wei Li, M.S. *Automatic Image Annotation Based on WordNet and Hierarchical Ensembles*. in *Computational Linguistics and Intelligent Text Processing: 7th International Conference, CICLing 2006, Proceedings, Lecture Notes in Computer Science*.
86. Wei-Cheng Lai, G.S., Edward Chang. *An Architecture of a Web-Based Collaborative Image Search Engine*. in *Conference: On the Move to Meaningful Internet Systems 2002, Proceedings, Lecture Notes in Computer Science*.
87. Xian Wu, L.Z., Yong Yu. *Exploring social annotations for the semantic web*. in *Proceedings of the 15th international conference on World Wide Web, Social networks*.
88. Xiangdong Zhou, J.Y., Lian Chen, Liang Zhang, Baile Shi. *Automatic Image Annotation Based on Topic-Based Smoothing*. in *Intelligent Data Engineering and Automated Learning - IDEAL 2005: 6th International Conference, Proceedings, Lecture Notes in Computer Science*.
89. Xiangdong Zhou, L.C., Jianye Ye, Qi Zhang, Baile Shi. *Automatic Image Semantic Annotation Based on Image-Keyword Document Model*. in *Image and Video Retrieval: 4th International Conference, CIVR 2005, Proceedings, Lecture Notes in Computer Science*.
90. Yohan Jin, L.W., Latifur Khan. *Improving Image Annotations Using WordNet, Advances in Multimedia Information Systems*. in *11th International Workshop, MIS 2005, Proceedings, Lecture Notes in Computer Science*.
91. Youngchoon Park, P.K., Forouzan Golshani, Sethuraman Panchanathan. *Concept-Based Visual Information Management with Large Lexical Corpus*. in *Database and Expert Systems Applications: 12th International Conference, DEXA 2001, Proceedings, Lecture Notes in Computer Science*.
92. Yuechen Qian, L.M.G.F. *Turning Photo Annotating Tasks into Instant Messaging Fun: Prototyping, User Trials, and Roadmapping*. in *Entertainment Computing – ICEC 2004: Third International Conference, Proceedings, Lecture Notes in Computer Science*.
93. Yuqing Song, W.W., Aidong Zhang, *Automatic Annotation and Retrieval of Images*, in *World Wide Web*. p. 209-231.
94. Zhe-Ming Lu, H.S., Hans Burkhardt. *Image Retrieval Based on a Multipurpose Watermarking Scheme*. in *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, Proceedings, Lecture Notes in Computer Science*.
95. Zheng Chen, L.W., Feng Zhang, Mingjing Li, *Web mining for web image retrieval*. Journal of the American Society for Information Science and Technology. 52(10): p. 831-839.

Appendix I – User interface for SYGY^{TAG} and SYGY^{Map}

As we discussed in chapter 4, SYGY game is divided into two counterpart games called SYGY^{TAG} and SYGY^{Map}. Adobe Flash is used to develop the games thus any browser with the latest installation of Flash is capable of loading a game through the internet. The game is accessible at <http://www.sygy.org>. In this appendix, we have included some screenshots from gaming interface including a short description of each shot.

The games could be played independently but we recommend that if the game is used for educational purposes in the class, the instructor could guide students to play SYGY^{TAG} first and after completion of the game session, they may move forward with SYGY^{Map}.

In SYGY^{TAG}, the player should use provided drawing tools to highlight an object inside a preloaded digital image on the screen. After highlighting the object, he/she has to assign a label to that object and then press “Submit” button. The highlighter region and the assigned label is sent to the server and after score calculation on the server, the current earned point and the total points is returned to the client as a response. The player is supposed to choose and label as many objects as possible during a countdown time period.

SYGY^{Map} is the game for creating facts about objects in digital images. The objects are formerly identified and labeled through SYGY^{Map} game and that is why we call these two games as counterparts. They are actually complemented to each other. In SYGY^{Map}, the player has a limited time (like 200 seconds) to create as many facts as possible. When the player chooses an object among the list of objects in the drop down box, that specific object will be highlighted and a predefined template of facts is shown on the screen. At this time, the player has to choose the correct facts based on his/her own judge. There are usually two choices for each fact. One choice is right and the other one is wrong. After completion of the fact sheet, the player will press the submit button to receive the points from the server. If the selected answers from the player would be matched with larger group of players, the player has a good chance to earn high score. If the player feels that he/she can not work with the loaded image, he/she can press

“PASS” button to skip to next image. Almost 5 images are allowed to be skipped in each game session.

Description of screenshots:

Referring to Figure AI- 1: The game is accessible at www.sygy.org. The player needs to choose which game to play.

Referring to Figure AI- 2: SYGY^{TAG} is chosen. The player should provide an email address and select number of images to play in one game session.

Referring to Figure AI- 3: The game session for SYGY^{TAG} is loaded into game display and it is ready to get started. The player has chosen four images to play with (4 thumbs at left hand bar). Scoring hints and start button are shown on the screen. The countdown time will start counting right after the player clicks on “Start” button. Drawing tools are accessible on the right side of the game display.

Referring to Figure AI- 4: The game is started. The player has provided the label “Ball” for the object seen in the image. The earning point for label “Ball” is 665. A message is shown on top of game display saying: *“Good! Seems you are getting professional”*

Referring to Figure AI- 5: The second image is selected. Remained time is 98 seconds. A new object (“Fish”) is labeled. The score for the label “Fish” is 732. Total score is shown in the left bottom corner.

Referring to Figure AI- 6: The time is expired. Summary of the results is shown at the top right corner. The player may start a new game session. List of high scorers is accessible in this page. The player can also change his/her nickname.

Referring to Figure AI- 7: High score list is shown on the screen.

Referring to Figure AI- 8: SYGY^{Map} is selected and loaded into game display. The 1st random image is shown to player. The player has selected the object “Computer” from drop down menu. The object is highlighted in the screen.

Referring to Figure AI- 9: Drop down menu is expanded. Several objects are on the list. The player has to select one of the objects on the list to create facts for it.

Referring to Figure AI- 10: The player has used “Pass” button to skip to the next image. The object “Woman” is selected from drop down menu. After selection the object, the fact sheet is displayed on the screen. There are two options in front of each fact. The player must click on YES or NO to confirm or reject each fact. Finally the player must submit the fact sheet to receive points.

Figure AI- 1 Choosing Game

Figure AI- 2 Selecting number of images to play in SYGY^{TAG}

Appendix I – User interface for SYGY^{TAG} and SYGY^{Map}

Figure AI- 3 Scoring hints before pressing game start

Figure AI- 4 Labeling an object; earning points for the label "Ball"

Appendix I – User interface for SYGY^{TAG} and SYGY^{Map}

Figure AI- 5 Labeling the “fish”

Figure AI- 6 Time Expired

Appendix I – User interface for SYGY^{TAG} and SYGY^{Map}

Figure AI- 7 List of high scores

Figure AI- 8 Selecting an object in SYGY^{Map}

Appendix I – User interface for SYGY^{TAG} and SYGY^{Map}

Figure AI- 9 List of objects in SYGY^{Map}

Figure AI- 10 Fact sheet for object “Woman”

Appendix II – Scoring algorithm for SYGY^{TAG}

AII.1 Interpolation of Tags in SYGY^{TAG}

As we stated in Chapter 4, we are aimed to come up with a new fair scoring algorithm to generate points for SYGY^{TAG} players. Going back to the mechanism of SYGY^{TAG}, we mentioned that each player is compared against all former players unlike other in which two players are paired online at the same time. There are two information elements which we will gather during the game: A- The location of the object and B- The label. These two elements are the inputs to our system and should be used for calculating scores in SYGY^{TAG}. By having such information, we try to match each player with earlier players to see if they are matched with each other or not. The matching is based on the label and the region. First of all we check the label which is submitted by the player. If the label would be matched with previous player(s) (even one player), we move to the next level which is comparing the regions between current player and all earlier players who had provided the same label for the same image.

Two distinct regions are illustrated in Figure AII- 1: The light region and the dark region. The light region is called Center of Attention (CoA) and the dark region is the player's selected region (which is marked by X). There are three common cells between two regions (CoA and player's selection) and we have numbered them 1, 2, 3. CoA is actually a dynamic region which is shaped based on consensus of players. It shows an average segment of interest inside an image for a specific label among all players who have played with a same image. This CoA is calculated and updated each time a new player uses the same Label for a similar region in one image.

Figure AII- 1 Common region between Center of Attention and player's selected region (3 Cells)

Since players are shown the same set of images while playing, they will probably have similar selections with the same Labels. For example if the first player finds a horse inside an image, he may choose 10 cells (those covering the horse region) and label it “Horse”. The second player may not choose the horse but the third player would choose the horse with a different blend of cells and the same Label. After some time, a few more players would play this image and a group of cells would receive a higher selection rate. This rate is called the *Cell’s Weight*, meaning that the more players selecting a cell, the more weight it gains.

To simplify the CoA detection process, we have used the norm of the maximum weight to select our CoA for each Label. If the maximum cell’s weight for a Label is Z , then the CoA is the group of those cells which have weights higher than $Z/2$. Succinctly, the CoA cells for a specific Label are:

Weight of CoA Cells $\geq 50\%$ of Maximum Cell’s Weight

Looking back at Figure AII- 1, the CoA has 15 cells and the player has selected 11 cells with 3 cells in common with CoA.

In Figure AII- 2, we have illustrated the weight of cells in a bar diagram. A higher weight means a taller bar. Some bars have the same height meaning those cells have the same weight.

Figure AII- 2 Bar diagram of Cell’s weight for a Label

Considering scoring strategy, following question would arise: who gets a higher score among the players who have labeled an image, or simply, what is a good Label? The answer to this question is the foundation of measuring the commonsense consensus among the players and scoring each player.

The attributes of a good Label are:

- The verbal label should not be nonsense or rarely used.

- The verbal label should be as simple as possible in order to have a higher chance of matching with other players.
- The selected region should not be too far from the focused area for the proposed label (distance from CoA).
- The selected region should not be so large or so small compared to CoA.

These attributes clarify that the maximum score should go to a player whose label is exactly same as others and his/her selected region entirely covers the CoA. If the player selects a bigger region comparing to the size of CoA, then he/she does not have a great chance to earn a high score. Vice versa, if the player selects a very small region comparing to CoA, he should not receive a high score either. In SYGY^{TAG}, the score has the same meaning as the distance to consensus. More distance to the consensus point results in a lower score and less distance from the consensus point results in a higher score. Based on these presumptions, we understood that any scoring formula should take all the aspects of consensus into consideration.

Two major factors for measuring the consensus are: 1- Closeness of a player's selection to CoA in terms of number of cells 2- The number of common cells between CoA and the players' selection.

The first factor tells that if the CoA is covering a few cells and the player selects a very big region which could even include CoA inside itself, the player is not so much accurate since he has just tried to make a big selection to cover almost every other possible selection. In this case, the player will not catch a high score because the number of cells he has selected is somewhat larger than the number of cells in CoA and thus he is not that much accurate.

The second factor is more challenging. Each selected cell for a label has a weight and we should take the collective weight of all selected cells into consideration for getting to the right answer. We came up with an idea of interpolating Labels on a 3D surface to resolve the second factor in an easily understandable and applicable mathematical model.

AII.2 Surface interpolation using Gaussian function

As it has been discussed, one part of consensus measurement is based on the cells' weight. Those cells which have higher weights are showing a greater commonsense level. If the player selects N Cells and provides a label, we need to scan all of these

cells, find the weight for each cell and add them up. It seems that the collective weight of all cells is a meaningful figure for showing the consensus level. If we model this system in a three dimensional space, the X-Y plate provides the location of cells and the Z axis gives us the cells' weight. In this case, if we try to pass a surface through all major points in this space, we have actually interpolated all the weights. It is obvious that we will have a different surface for each label. The confined volume between the surface and X-Y plate for the player's selected cells is the commonsense level of this player compared to other players. This interpolated surface will help us have a better understanding of the Labels' density over the image and also gives up a better way to interpret a fair scoring strategy for SYGY^{TAG}.

We studied different surface interpolators. Among them, Radial Basis Function (RBF) was interesting since by using a good basis function for RBF, we would reach a smooth surface. RBF is mostly used in curve-fitting problems and can be presented as:

$$F(x) = \sum_{i=1}^N w_i \phi(\|x - x_i\|)$$

AII- 1

where $\phi(r)$ is the radial basis function. w_i is a real-valued weight, $\|\cdot\|$ denotes the Euclidean norm, ϕ is a *basic function*, and $\|x - x_i\|$ is simply a distance, i.e how far x is from the point x_i . Four popular choices for $\phi(r)$ are:

- a. Multiquadrics: $\phi_m(r) = \sqrt{r^2 + c^2}$, $c > 0$
- b. Inverse multiquadrics: $\phi_i(r) = 1/\sqrt{r^2 + c^2}$, $c > 0$
- c. Gaussian: $\phi_g(r) = \exp\{-r^2/(2\sigma^2)\}$, $\sigma > 0$
- d. The thin-plate spline: $\phi_t(r) = r^2 \log(r)$

Although it is helpful to use RBF to find the surface function, we decided to use an alternate method to simplify the model and to reduce the calculations for finding coefficients of RBF. In our method, 2D Gaussian functions are employed to construct the surface. The surface for a label is presented as a linear summation of gaussian basis functions and has the form:

$$F(r) = \sum_{i=1}^N w_i \exp \left[-\frac{\|r - c_i\|^2}{2\sigma_i^2} \right]$$

AII- 2

in which, the w_i , c_i , and σ_i are the weight of each cell, center of each cell and the variance of each gaussian basis function, respectively. $\|r - c_i\|$ is simply a distance.

Each Gaussian function is located in center of each cell. The Gaussian distribution function has a belly shape with very specific attributes. One of these helpful attributes is the variance (σ_i). As it is shown in Figure AII- 3, the Gaussian function will decline to less than 0.01 in $3\sigma_i$ distance from the center.

Figure AII- 3 A Sample Gaussian Distribution function

As previously mentioned, centers of Gaussian functions are located at the center of each cell, and the σ values of them are all the same. The weight of each basis function is equal to the normalized weight of the cell which contains the center of that basis function.

There is one more condition to take into account and that is semi-accurate consensus. Sometimes a player may not exactly choose those strong cells but the adjacent cells instead. In this case, the player is judged as semi-accurate for his selection and based on that, he may be rewarded a fractional score. The good news about Gaussian basis functions is that we can adjust them to partially cover the adjacent cells, so we meet our requirement in order to produce scores even if the adjacent cells were selected. To achieve this, it is recommended that σ values of Gaussian basis functions should be equal to half of the cell's width:

$$\sigma = \text{Cell's Width} / 2$$

Since the main intensity of Gaussian function is vanished at 3σ so we can say it will be mitigated up to the margins of adjacent cells. In Figure AII- 4, this effect is illustrated.

Appendix II – Scoring algorithm for SYGY^{TAG}

Figure AII- 4 Comparison of two methodologies for adjusting sigma value in Gaussian function.

We have simulated the effect of three different σ values in Figure AII- 5. There are 9 cells in this simulation which have equal weights.

In series a1,a2,a3: $\sigma = \text{Cell's Width}/6$

In series b1,b2,b3: $\sigma = \text{Cell's Width}/3$

In series c1,c2,c3: $\sigma = \text{Cell's Width}/2$

Figure AII- 5 Comparison of three different sigma values for Gaussian interpolation

As it can be seen, in the series (a1,a2,a3) the cells do not overlap, while in the series (b1,b2,b3) the cells are partially overlapping and in the series (c1,c2,c3) the cells are entirely overlapping. The series (c1,c2,c3) is better to manipulate for two reasons:

- 1- The adjacent cells are covered since $\sigma = \text{Cell's Width}/2$
- 2- The interpolated surface is smooth (not bumpy)

In the series (a1,a2,a3) and (b1,b2,b3), some gaps between Gaussian bells are observable because the Gaussian functions have partial overlapping. In the series (c1,c2,c3), the surface is no more bumpy and the gaps are covered.

In Figure AII- 6, we have used a simulation tool for showing the effect of Gaussian function over the “Nose” of the boy. Since most players have labeled the nose, the Center of Attention with the higher cell weight is shaped around the nose area.

Figure AII- 6 Conversion of the nose area using Gaussian function for interpolating the Label “Nose”

In Figure AII- 7, we have shown 9 cells which have equal weights (as a sample). The cells marked by X are adjacent to the CoA and they are partially overlapping the CoA. The strongest cell is the cell number 5, which receives the power from all surrounding cells in the CoA. It is also noticeable that we have 3 groups of Cells in this figure:

Group1: Cell number 5 which is the strongest cell.

Group2: Cells 1,3,7,9 which have the same value.

Group3: Cells 2,4,6,8 which have the same value.

Figure AII- 7 Effect of Gaussian function over nine equally weighted adjacent cells

AII.3 Score Calculation

After interpolating the surface, we define our scoring formula. As we discussed earlier, one of the factors for measuring the commonsense level between the current player and all previous players is the level agreement between player's selection and the others'. The elaborated answer is the confined volume between X-Y plate and interpolated surface for player's selected region. More volume means more common strong cells. This is one clear factor for our scoring formula.

The second factor is the closeness of a player's selection to the CoA. We show this factor by α in following score formula:

$$\text{Score} = \alpha \times V_{\text{confined}}$$

AII- 3

We may call α as the similarity coefficient, and V_{confined} is the confined volume between X-Y Plate and the surface for the selected region. The confined volume can be calculated by a dual integral for the player's selected region in distinct range:

$$V_{\text{confined}} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} F(x, y) dx dy$$

AII- 4

After replacing formula 2 in formula 4, we have:

$$V_{\text{confined}} = \int_{y_1}^{y_2} \int_{x_1}^{x_2} \sum_{i=1}^N w_i \exp \left[-\frac{\|(x-x_i)^2 + (y-y_i)^2\|^2}{2\sigma_i^2} \right] dx dy$$

AII- 5

Now for similarity coefficients we are proposing a formula based on the number of cells selected by the players and the number of cells in CoA. This similarity coefficient measures the correlation between the selected area and the CoA.

$$\alpha = g(n_{user}, n_{CoA})$$

AII- 6

The function we have come up with for α operates like a band-pass filter. It has a flat top for a small range of variance and falls down rapidly by moving far from the center, meaning if $n_{CoA} = n_{user}$, then $\alpha = 1$ and if n_{CoA} and n_{user} have large difference, then the α will be quite low. The function is:

$$g(n_{user}, n_{CoA}) = \frac{1}{1 + \left(\frac{n_{user} - n_{CoA}}{n_{CoA}}\right)^2}$$

AII- 7

The function has been depicted in Figure AII- 8.

Figure AII- 8 The coefficient is like band-pass filter

As an example, if a player selects 10 cells and the CoA has 5 cells, then:

$$n_{CoA} = 10, n_{user} = 5, \alpha = 0.5$$

Looking back on the scoring formula, we realize if a player exactly selects all CoA cells, then $\alpha = 1$ and the confined volume will also be a big figure since the CoA cells have strong weights.

The implementation of this scoring formula in SYGY^{TAG} proved the efficiency of the concept. Historical scores for each player are saved in our repository in order to help analyzing the players' behavior over a period of time. This could be useful for tuning the current scoring formula to include several other hidden elements in calculations.