



SHIBAURA INSTITUTE OF TECHNOLOGY

**A method of graph information
extraction and retrieval for academic
literatures by use of semantic
relationships**

by

Sarunya Kanjanawattana

A dissertation submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Division of Functional Control System
Graduate School of Engineering and Science

September 2017

"Learning is a never-ending process. Those who wish to advance in their work must constantly seek more knowledge, or they could lag behind and become incompetent."

His Majesty King Bhumibol Adulyadej
The Great king of Thailand

To my parent who give me unwavering love and inspiration.

To my family for encouraging me to do my best.

To my friends for supporting me in everything I do.

*To teachers who kindly advise and motivate me to accomplish
my goal and fulfill my dream. . .*

Acknowledgements

Though only my name appears on the cover of this dissertation, many great people have supported to the contribution and cooperated until it has been succeeded. I would like to convey my heartfelt gratitude and sincere appreciation to all people who has made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My experience at Shibaura Institute of Technology (SIT) has been unforgeable. I have felt a warm welcome since my first day in Japan. I have been given unique opportunities and taken advantage of them. Definitely, this dissertation would not have been accomplished without the scholarship of Shibaura Institute of Technology and Japanese Government (Monbukagakusho or MEXT) Scholarship.

My deepest gratitude is to my supervisor, Prof. Masaomi Kimura. He has been supportive since the days I joined his laboratory. I have been incredibly fortunate to have a supervisor who gave me inspiration and motivation. Ever since, he has supported me not only by providing a research assistantship throughout three years but also academically and emotionally through the rough road to finish this dissertation. He guided and helped me when I faced problems during the study. Thanks to him I had the opportunity to practice and acknowledge the research process for my career. In the future, I would follow his step and become a good supervisor to my students as he has shown to me.

I would like to extend my gratitude to my dissertation committees: Prof. Toru Sugimoto, Prof. Michiko Ohkura, Assoc. Prof. Ryota Horie, and Prof. Hideaki Takeda, for precious comments, including the questions which intended me to widen my research from various perspectives. I am gratefully indebted to them for their very valuable comments on this dissertation.

I would also like to thank the participants who were involved in the validation survey for this dissertation. Without their passionate participation and input, the validation survey could not have been successfully conducted.

Dozens of people have helped and encouraged me immensely. I am grateful to friends in Data Engineering Laboratory and SIT friends for many supports, discussions, evaluations, and for all the great time we have had. Further, I would like to show my thank to my friends in Suranaree University of Technology for their encouragement.

Finally, I must express my very profound gratitude to my parents and to my sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you very much.

Sarunya Kanjanawattana

SHIBAURA INSTITUTE OF TECHNOLOGY

Abstract

Graduate School of Engineering and Science
Division of Functional Control System

Doctor of Philosophy

by Sarunya Kanjanawattana

Information retrieval is a fundamental technique for modern search engines. This technology is designed to advocate information discovery. Generally, the data used by this technology is text descriptions. However, images, especially graph images, typically contain much useful information. In academic literature, the graph images are very important to summarize and represent experimental results and statistical data. Therefore, a search engine system to discover the graph images and their information is definitely necessary for researchers to obtain precise and concise knowledge. However, to utilize both graphical and literal information, the problem of the semantic gap should be addressed. To do so, it is necessary to give meaning to the graphical information and link it to the linguistic information; thus, a proper solution is to use an ontology to bridge the gap. Regarding the necessity of my dissertation, the users necessitate employing the proposed system because it provides precise and concise information from relevant graphs with less ambiguity. This should advocate their studies and fulfill their academic inquiries.

The main objectives of this research were to solve the problem of semantic gap by constructing an ontology-based search engine system as well as to design ontology and database schemes to support the search engine and OCR-error correction systems.

In this dissertation, I proposed a novel ontology-based search engine system applied to the graph images and their descriptions. To obtain the graph information, I also introduced several systems: graph-type classification, graph components extraction and identification, OCR-error correction, and graph information extraction. After their processes were completed, much of knowledge have been acquired from the graphs. An ontology and a database were constructed to store the obtained knowledge for utilizing in the search engine systems.

This system contributes several benefits and usefulnesses to society, particularly in academics. Researchers need accurate and reliable information to support their studies. This system can fulfill their requirements by providing the relevant graphs and concise information. The ontology also offers new knowledge; though, a relational database cannot surpass this benefit. The main contribution is that the novel ontology-based search engine system applicable to the new design of ontology storing graph information.

All systems had been tested and presented results, including new findings. The results showed that the performance of each system proposed in this dissertation was highly effective. The F-measure reached to 0.7, which was much higher than the traditional search engine system. It clarified that the ontology-based search engine system provides precise and concise information outperforming than the ES-based search engine system. To sum up, the objectives of each study have been achievable proven by their evaluations.

In my future research, I intend to concentrate on improving the systems to cover the user's needs. I will increase a size of data and extend study domains. To improve the efficiency of the system, an answering question system will be an attractive function because the users can directly query some questions to the system and obtain accurate knowledge. This function may be developed by using a deep learning. Additionally, a keyword recommendation system will provide benefits to the users. This function will analyze the user behavior and suggest some possible keywords relating to a user intention. This will be great, if the system will be published on the Internet. Moreover, it should be assembled to other existing ontologies.

Contents

Acknowledgements	v
Abstract	vii
List of Figures	xiii
List of Tables	xvii
Abbreviations	xix
1 Introduction	1
1.1 Background	1
1.2 Ontology definition of this study	4
1.2.1 A theoretical definition	4
1.2.2 A practical definition	5
1.3 Problem Statement	6
1.4 Objectives	7
1.5 Outline of the dissertation	8
1.6 Contribution	10
1.7 Novelty of the Study	11
1.8 Structure of This Dissertation	11
2 Related works	15
2.1 Classification On Image data	15
2.2 OCR Error correction	18
2.3 Use of Ontology for Image Information Retrieval	21
3 Graph-type classification	23

3.1	Background	24
3.2	Methodology	27
3.2.1	Definition of my datasets	27
3.2.2	A proposed method	30
3.2.2.1	Preprocessing step	30
3.2.2.2	Application of classification	33
3.3	Experiments and results	37
3.3.1	Comprehensive tests	37
3.3.2	Results	40
3.4	Discussion	43
3.5	Conclusions	51
4	Graph Components Extraction and Identification	53
4.1	Background	54
4.2	Methodology	55
4.2.1	Axis description extraction	55
4.2.2	Legend extraction	57
4.3	Experiments and results	62
4.4	Discussion	64
4.5	Conclusions	66
5	Graph-based Optical Character Recognition-error Correction	69
5.1	Background	70
5.2	Methodology	71
5.2.1	Data collection	72
5.2.2	OCR-error correction	73
5.2.2.1	Candidate selection	73
5.2.2.2	Ontology design and creation	74
5.2.2.3	Error correction	75
5.3	Experiments and results	79
5.4	Discussion	84
5.5	Conclusions	88
6	Graph Information Extraction	91
6.1	Background	92
6.2	Methodology	93
6.2.1	Ontology	93
6.2.2	Extraction of graph information	96
6.2.2.1	Data content identification	96
6.2.2.2	Ontology construction	98

6.3	Simulations	100
6.4	Discussion	103
6.5	Conclusions	104
7	A Prototype of Ontology-Based Search Engine System	105
7.1	Background	106
7.2	Methodology	107
7.2.1	Database design	108
7.2.2	Ontology design	111
7.2.3	System implementation	111
7.3	Conclusions	120
8	Experiment and Evaluation	123
8.1	System Evaluation Background	124
8.2	Experiment configuration	125
8.3	Experiment procedures	126
8.4	User Feedback Evaluation	129
8.4.1	Participants	129
8.4.2	Results and Analysis	129
9	Discussion	139
9.1	Findings of this dissertation	140
9.2	Discussion of this dissertation	145
9.3	Limitations and possibilities of the study	146
10	Conclusions and Future Works	157
10.1	Conclusions	157
10.2	Future works	160
A	List of Publications	163
A.1	International Journal Papers	163
A.2	International Conference Papers (Peer-reviewed)	164
A.3	Workshop	165
B	Background of generic search engines	167
B.1	System Evaluation Background (Appendix part)	167
	Bibliography	175

List of Figures

1.1	Content structure of this dissertation	13
3.1	Example displaying two scatter plots with different characteristics and patterns: (a) scatter plot containing only points and (b) scatter plot containing points in different positions than (a) and a line	26
3.2	Illustrating the core process of one-dimensional image construction accomplished by applying a DFT	31
3.3	Demonstrating the process of classification by applying the ANNs, then the SVMs	34
3.4	Processes of all experiments: (a) applying the CNNs to 1Dimg and 2Dimg in CNN_1Dimg and CNN_2Dimg respectively, (b) applying the SVMs to WLHT in SVM_WLHT, (c) applying the ANNs to WLHT in ANN_WLHT, (d) applying the SVMANN to WLHT in SVMANN_WLHT, and (e) applying the ANNSVM to all datasets in ANNSVM_1Dimg, ANNSVM_2Dimg, ANNSVM_WL, ANNSVM_HT, and ANNSVM_WLHT	38
3.5	Results from CNNs and ANNSVM that used 1Dimg and 2Dimg: (a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments	40
3.6	Results from ANNSVN that used WL and HT a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments	41
3.7	Results from SVM, ANN, SVMANN, and ANNSVM that used WLHT: (a) table statistically presenting summarized results, (b) bar graph graphically illustrating results from SVM_WLHT and ANN_WLHT, and (c) bar graph graphically showing results from SVMANN_WLHT and ANNSVM_WLHT	42
3.8	Simulation of Coiflet 1 [73], analyzing as one-dimensional images	45
3.9	Illustration of three different wavelets [73] with three waves that have high amplitude values, as indicated in the dashed red circles: (a) mother wavelet of Coiflet 5, (b) mother wavelet of Symlet 10, and (c) mother wavelet of Symlet 20	47

3.10	Detailed accuracy separated by classes and a confusion matrix which belongs to the dataset of Coiflet 1 applied by my main method (ANNSVM)	48
3.11	Results of the tests for checking an impact of the number of hidden layers	50
4.1	Process to extract X- and Y-titles from graph images based on their location: (a) image partitioning process; (b) pixel projection	56
4.2	Overall legend extraction procedures	57
4.3	Epsilon estimation to analyze the densities of each quarter to obtain the smallest distance to be valued as Epsilon	58
4.4	Examples of DFT results that present the difference between an image (a) with and (b) without a legend	60
4.5	Data transformation from a 2D-DFT image to a single-row numeric dataset used for classification	60
4.6	Performance rates of axis description extraction	64
5.1	Illustration of graph components	72
5.2	Steps of candidate selection	73
5.3	Demonstration of my ontology structure describing entities, properties, and relations	74
5.4	Example of grammar dependency parsing, including POS tags and typed dependencies, and NER classes of each token	76
5.5	Demonstration of OCR-error correction covering possible conditions to filter and correct errors	77
5.6	Illustration of accuracies and noise ratios of all experiments	82
5.7	Illustration of precision, recall and F-measure of all experiments	82
5.8	The number of tokens, including accuracy rates of each condition	83
6.1	Representation of my ontology structure describing classes, properties, and relations	94
6.2	Overall of proposed system	96
6.3	Bar height extraction using pixel projection and a step function	99
6.4	Pixel proportion calculation	99
6.5	SPARQL query command and answers for Question 1	101
6.6	SPARQL query command and answers for Question 2	101
6.7	SPARQL query command and answers for Question 3	101
6.8	SPARQL query command and answers for Question 4	102
6.9	SPARQL query command and answers for Question 5	102
7.1	Illustration of relational database used in this system	108
7.2	A part of database storing generic data related to the graph images	109

7.3	A part of database storing feedback data for the search engine system (a final system) in Feedback mode	110
7.4	Illustration of the updated ontology, i.e., observing at read arrows, used in this system	112
7.5	A user interface of search page with three sections	113
7.6	Selectable conditions for results filtering	114
7.7	Selectable features that can be presented at the result section	114
7.8	Question 1 and its settings	116
7.9	Example of result performed by Question 1	116
7.10	Question 2 and its settings	117
7.11	Example of result performed by Question 2	117
7.12	Question 3 and its settings	118
7.13	Example of result performed by Question 3	118
7.14	Question 4 and its settings	118
7.15	Example of result performed by Question 4	119
7.16	Question 5 and its settings	119
7.17	Example of result performed by Question 5	120
7.18	Question 6 and its settings	120
7.19	Example of result performed by Question 6	121
8.1	System flow in feedback mode	127
8.2	User page	127
8.3	Illustrating the results of query in feedback mode	128
8.4	Questionnaire page	128
8.5	Selected keywords for each participant and experiment iteration	130
8.6	Statistical results analyzed by three performance models: precision, recall, and F-measure	131
8.7	Average precision, recall, and F-measure from ES-based and ontology-based search engine systems	133
8.8	Mean of scores	134
8.9	List of questions	134
8.10	Standard deviation of scores	135
8.11	Scores of each question in Questionnaire page provided by 10 participants	135
8.12	True performance of the search engine system without outliers	136

List of Tables

4.1	Evaluation results of classification for METHOD 1	63
5.1	Settings of my experiments	81
9.1	List of studies and their core findings	140
9.2	List of limitations of all proposed systems.	149
B.1	Summary of classic search engines and their features.	173

Abbreviations

2D-DFT Two-dimensional Discrete Fourier transform

2Dchart Two-dimensional charts

ANNs Artificial neural networks

API Application programming interface

CNNs Convolutional neural networks

DBSCAN Density-based spatial clustering of applications with noise

DFT Discrete Fourier transform

DSL Domain specific language

ES Elasticsearch

GPU Graphics processing unit

IDF Inverse document frequency

LSI Latent semantic indexing

NER Named-entity recognition

NLP Natural language processing

OCR Optical character recognition

OPTICS Ordering points to identify the clustering structure

P-value Probability value

PCA Principal component analysis

POS Part of speech

RBF Radial basis function

RDF Resource Description Framework

Abbreviations

SQL Structured query language

SVMs Support vector machines

TP True positive rate

VAS Visual analysis scale

Chapter 1

Introduction

The aim of this chapter is to introduce background and motivation that lead readers to comprehend an importance of this dissertation, which entitled “A method of graph information extraction and retrieval by use of ontology”. Existing problems and proposed solutions will be tentatively presented in this chapter. Further, I will suggest beneficial contributions and provide some examples. Finally, a summary of this chapter and a structure of this dissertation will be described.

1.1 Background

Information retrieval is a process to obtain information from resource collections, such as files and documents. The process begins when users input a query into a system and finishes when they retrieve some relevant outputs. Nowadays, information retrieval is recognized as being a baseline of search engine system, which examines related documents corresponding to keywords, as seen in Google and other search systems [14, 44]. The search engine system becomes a vital role in people’s life because there are a ton of information publicly provided on world wide web. Without the search system, it is hard to retrieve relevant information from huge resources. Basically, the amount of knowledge is often thoroughly described in body

parts of documents; however, only descriptive detail is inadequate to clearly demonstrate whole information to the readers. Thus, images are used to highlight essential information, particularly graph images.

To extract graphical information from the images, it is unavoidable to manipulate image features, such as shape and texture. However, a way to obtain information from graph images differs to generic photo images, because needed information are data interpretable by human rather than image shapes or colors. Several publications have appeared in recent years documenting about an extraction of low-level image features [24], but a few studies have focused on graph images [38, 54]. Indeed, the extractable information from graph images should be productive to users because they obviously obtain concise knowledge or the main point of a document relating to the graphs.

Traditionally, the graphs, which come from different sources, provide particular information and expression. For example, there are three graphs from physics, brain science, and computer science. All are line graphs presenting data corresponding to an equation. The graph from physics shows a relationship between velocity and time of a car. This can be interpreted that how fast the car is and how long it uses. Moreover, not only a distance computed by an area under a curve but also an acceleration of the car is impliedly presented in the graph. In contrast, if the graph from computer science demonstrates a relationship between time and website use. I can interpret that how the web page can service based on different times. However, an area under a curve may not infer any particular information. For a line graph from brain science, the expression should be different from other line graphs, if it results from electroencephalography, which is a device to monitor and record the electrical activity of the brain. The graph shows an electrical activity in real time; thus, this is simple to express when the brain has been activated. The expression of this line graph is particular because time in X-axis is not fixed to a constant range but real time. Moreover, a kind of graph is variant depended on the study domain. For instance, the graph in several study fields (e.g., mathematics and brain science) may represent a set of vertices and edges. Also, in computer science, Dijkstra algorithm also uses a graph to represent vertices and edges to trace and

find the shortest path. Another example is an ontology. It uses a graph to show relationships among concepts as a network. The graphs used in this research are collected from the scientific literature, especially in computer science and biology, with the following types: bar graph, line graph, and plot graph. The kind of the graph is a general graph representing a means of representing data. In academics, researchers created them to present data summaries from their experiments or some comparisons; thus, many kinds of information can be expressed in the scientific graphs, such as statistical data, category names, a comparison among data, and some critical changes along continuous data tendency.

Only the information graphically extracted from the graph images may be insufficient to acquire completed information. Definitely, descriptions of the graphs also provide necessary information that thoroughly explains about the graphs themselves. To fulfill this requirement, not only graphical but also literal contents should be used. It should be analyzed by a dependency parser, which is a tool for parsing sentences and expressing their relations [65]. This extractable information directly involves the graphs and is also useful to the users.

However, a critical problem, called semantic gap, has been addressed if both graphical and literal information are utilized together. Generally, the gap should be bridged by using ontology. For several years, great effort has been devoted to discovering solutions of semantic gap [88, 93]. In this dissertation, I propose a novel method of ontology-based search engine system that attempts to narrow the semantic gap by introducing solutions to extract and use graph information presented graphically and literally.

The necessity of this study is described here. Although the proposed system aims to solve the problem of the semantic gap, as similar to other existing studies, the users necessitate using this system because it offers accurate and succinct graph information from relevant graphs that should encourage their works and researches and accomplish their academic inquiries. Frequently, the users read academic literature and attempt to comprehend their main points. They analyze the graph images described by captions and paragraphs. By using the proposed system, the users do not need to read whole literature because it precisely introduces concise knowledge

from the graphs, which is extracted from graphical and literal information, without confusion. They obtain fast and elaborate knowledge that helps to instantly capture the main points of the graphs. Hence, the major motivation of this research is to support researchers and students by providing essential knowledge from the graphs precisely. Here, I decide to extract information from the graphs because it is simple to retrieve information from data analysis rather than other kinds of images; in addition, users should acquire useful information much easier.

1.2 Ontology definition of this study

1.2.1 A theoretical definition

Ontology, which originates from philosophy, is a term meaning a study of things of existence [7]. For example, I observe the nature of a thing and identify relative objects belonging to a group of abstract concepts. During recent years, computer science introduces technical definitions of ontology, which is different from the original meaning. A definition by Gruber [33] proposed the most referenced definition: ‘an ontology is an explicit and formal specification of a conceptualization’. He explained the ontology as a formal description of existed concepts and relationships. In this context, a specification is a way to acquire to obtain knowledge from a specific domain. A conceptualization represents a way to organize and structure knowledge by using a finite list of terms (concepts or classes of objects), which are described a domain of discourse, and relationships among terms (hierarchies of classes) [16]. Zhong et al. [94] provided the definition of ontology: ‘An ontology is a specification of an abstract, simplified view of the world’. As reviewed a study in [5], the author summarized some definitions of ontology existing in the previous studies such as Guarino [34].

In this research, an abstract definition of my ontology has been rearranged based on Gruber [33]’s definition. The ontology is a specification of shared knowledge in given domains describing the collaborating representations by using the corresponding concepts and relationships that are displayed in a simplified way of

expression such as a taxonomy. Moreover, it expresses implicit data because it bridges among a variant of concepts which may belong to a different domain but contain partial term similarity.

In conclusion, my ontology means an explicit knowledge structure discoursing collaborating representations such as textual and graphical representations for this research.

1.2.2 A practical definition

An upper ontology is used to support the semantic interoperability and facilitate the semantic integration of domain-specific ontologies by providing a common starting point for the formulation of definitions. It contains very general terms, e.g., object and relation, that are applicable across multiple domains. According to an assumption, when this generalization is performed in ontologies of multiple domains, a small set of generic terms that is the same in all these domains will be come up. For example, the Suggested Upper Merged Ontology or SUMO is an upper ontology intended as a foundation ontology for a variety of computer information processing systems. Commonly, SUMO concerned itself with meta-level concepts, i.e., general entities that do not belong to a specific domain, and thereby would lead naturally to a categorization scheme for encyclopedias. Moreover, a mapping from WordNet synsets to SUMO has also been defined [71].

The domain-specific ontology represents concepts which belong to particular meanings, including their relations. A domain ontology provides controlled and structured terms to annotate data in order to support a system to search desired results. For example, the Gene Ontology introduces a taxonomy and controlled vocabulary for describing genes and gene products. As similar to the ontology of this study, the domain ontologies are computer science and biology. Word terms relate to the domains that may be able to link to the science ontology in the future. However, a main focus of the system is to search relevant graphs based on information existed in the graphs and their descriptions. Therefore, a meaning of word term itself is not the most important matter but relationships of sentences in graph descriptions.

The design of my ontology composed of textual and graphical representations. For textual information, the ontology here described words terms or contextual usage of words as entities and word dependencies as relations. The definition of this textual part is additional information explaining graph images. On the other hand, the definition of graphical representation in this ontology is the realization of the existing data inside the graphs. The ontology describing the graphical representation was created based on the fundamental structure of the graphs. For example, the graph consists of axis titles, a legend, and data section. Moreover, the data section contains some explanation about data such as slope and bar height.

In conclusions, the definition of my ontology is the explanation about the graph image in the scientific literature.

1.3 Problem Statement

There is now ample research on an area of information retrieval, which is the baseline of search engine system. Many systems have been currently developed based on full-text and image searches. After users queried some specific keywords to systems, they returned relevant information to users, such as a list of documents and a collection of images. Those existing studies have registered tremendous success efficient methods to retrieve relevant data [92]. However, they provide only documents or images but do not include further information that may be needed by users, for example, data tendency or the highest data value. A combination of image information extraction and search engine systems should be a proper solution to solve this difficulty.

Based on human intelligence, graph images can be read and comprehended more quickly than the descriptive data. Indeed, they are included much information that may not appear in the descriptive detail of documents, such as the tendency of lines and a difference between two given data categories in a bar graph. This is clear that if graph information is extracted by an efficient system, new knowledge hidden in the graph should be obtained. However, image information extraction has received much attention in the past decade. A preliminary issue focused on many

existing studies is the semantic gap that is the main cause of user misunderstanding. It characterizes the difference between linguistic and graphical representations. If the gap reduces, the ambiguity also diminishes. This problem is also found in the image search [19]. To literately analyze the images, the problem of misunderstanding may occur, because user thoughts turn to mislead author's intention. A number of researchers have addressed the problem of the semantic gap and propose several solutions. [22, 35, 45, 93]. In this dissertation, I introduce the idea to utilize ontology containing both graphical contents from graphs, e.g. graph components and graphical data appeared in data sections, and high-level representation, e.g., captions and cited paragraphs, in order to minimize the problem wisely. This should be a proper solution to minimize the gap of semantic.

1.4 Objectives

The major purpose of this dissertation is to introduce a novel graph search system integrating several techniques that I proposed in the past, such as graph type classification, graph component and information extraction, and OCR-error correction. Therefore, a list of major objectives of this dissertation is presented as follows:

1. To minimize the problem of the semantic gap between linguistic and visual representations by proposing an ontology-based graph search engine system.
2. To design my ontology structure to support the search engine system.

The target of the study is to address the problem of semantic gap by using my ontology. However, only utilizing ontology is inadequate to efficiently mitigate the problem; therefore, I must propose some methods to deal with information from the graphs. The list below describes minor objectives related to proposed systems.

1. To classify the different graph types and identify the dominant characteristics.

2. To identify locations of graph components appearing in the graphs by using clustering as well as extract information residing in the components by using OCR.
3. To cope a limitation of DBSCAN clustering with automatic Epsilon estimation.
4. To propose the OCR-error correction system to suggest correct recognitions by utilizing both local and external ontologies.
5. To extract information located in the data section of the graph in graphical form to quantitative data, including finding extend information such as a relationship of axes titles.
6. To design a database to record primary graph description and user feedback.
7. To determine an effectiveness of my ontology-based search engine system by comparing to a traditional search engine system by using the same data collection.

The performance and efficiency of the ontology-based search engine system are evaluated by users. They must assess the obtained results by deciding them as for whether relevance or irrelevance and response questionnaires. A traditional search engine system, which is an open source software, is selected for a purpose of evaluation comparison; hence, the users clearly examine a difference between both systems. After the evaluation process, I measure and present the performance of the systems, e.g. F-measure, precision, and recall.

1.5 Outline of the dissertation

In this dissertation, I focus on the graph image which graphically represents a set of data by using symbols, such as bars in a bar graph, lines in a line graph, and qualitative data. The graph is often used for ease of understanding of large data quantities and the relationships between parts of the data. In this research project, only three types of graphs are considered: line graph, bar graph, and plot

graph. Note that the plot and line graph are recognized as two-dimensional charts (2Dchart) because I analyze primary structures of both graph types and find several similarities. For example, data are usually displayed on two-dimensional axes with a periodic scale. Moreover, both graph types provide a similar mean of information, because a line is comprised by multiple plots located in data space. A bar graph shows rectangular bars with lengths proportional to the representing values. To compare significant characteristics between the bar graph and 2Dchart, the bar graph presents continuous data at Y-axis and discrete data at X-axis; meanwhile, 2Dchart shows continuous data on both axes.

This dissertation reports the findings of a thorough study and introduces novel methods, including their evaluations. The experimental results have proven that the methods have been effective and succeed to achieve goals of this research. I herein proposed four studies to resolve different problems.

The first method was graph image classification [48] whose main goal was to distinguish different types of graphs based on their characteristics extracted by using discrete Fourier transformation (DFT) and Hough transformation.

Second, I proposed the graph component extraction and identification [52]. The graph components represent X-title, Y-title, and legend. Note that the legend means data labels that should be presented on the graph with multiple data. This proposed system utilized to identify the position of the graph components, especially the position of legend, which might establish in different locations depending on authors' attention. Density-based spatial clustering of applications with noise (DBSCAN) had been used to identify the legend's position.

Third, the OCR-error correction system had been proposed to solve OCR error recognition, as shown in [51]. I extracted the information from graphs by using optical character recognition (OCR) to recognize text characters from the graph components. OCR probably provided error outputs because of misrecognition that leaded misunderstanding to users. Therefore, this OCR-error correction could solve this problem by using an ontology to suggest corrected results.

Finally, to obtain significant information from the data section of graphs, I introduced the graph information extraction [50]. It could detect and transform the graphical data (such as bars and plots) to quantitative values. Moreover, this disclosed explicit and implicit knowledge by the ontology. Note that the explicit information was provided purposely that could easily realize data intention such as bar heights. The implicit information could be acknowledged by using the ontology to investigate relationships or concepts such as graph relationships.

Hence, my entire studies are assembled into one main system, called the ontology-based search engine system for graph images. This search system provides not only graph images but also graph information. A common evaluation method dealing with a search engine system is to collect user feedbacks [5, 78]. For my evaluation, ten participants have been gathered in order to validate the system. User feedbacks and suggestions are collected for evaluation. Further, system performance is computed and represented by performance models, such as F-measure and precision. Note that the participants who attend my evaluation process should have experience about computer or biology.

1.6 Contribution

The graph search engine system offers social benefits, as it can give access to implicit and explicit knowledge. Moreover, it provides extraordinary features because not only relevant images but also new information collected from extraction process and discovered from ontology have been acquired from my ontology-based search engine system. This system has a range of applications, for example in image interpretation system. It also can be applied to other kinds of the image such as color photos and extract such information based on their low-level features.

Regarding this system utilization, it should work well in academics and research areas because reliable graphs often appear in journals or proceedings, and this system can quickly provide concise information to the users. This means the users do not need to read whole documents to understand what the main focuses of the graphs are. Moreover, researchers can use the search system to investigate

graphs that are relevant to their studies, such as a bar graph presenting experimental results of a study. For example, after the researchers completely conducted their experiments and obtained some results. For discussion, they may need results from other related studies to make a comparison. Further, users can specify a graph type for filtering relevant results.

1.7 Novelty of the Study

The method developed here has never been used in previous attempts to extract knowledge from graphical and literal contents in graph images. The novelty of the study are described as follows:

- New method of ontology-based search engine system.
- New ontology and database design supporting the search system.
- New methods of the graph-type classification system, graph component extraction, OCR-error correction using ontologies and graph-content extraction.

There are a lot of existing studies focusing on ontology design to solve the problem of semantic gap, as similar to this study. However, this study focuses on the graph information extraction and designing the new ontology to mitigate the semantic gap problem. Obviously, this is the first trial to use both visual and textual contents from graphs which are recorded into the ontology.

1.8 Structure of This Dissertation

The remainder of the dissertation is organized into 10 chapters as follows:

The next chapter discusses previous researches related to the topics of this dissertation. From Chapter 3 to Chapter 7, I describe methodologies of each proposed study respectively: graph-type classification, graph-based optical character

recognition error correction, graph components extraction and identification, graph information extraction, and prototype of graph-Based search engine system. Chapter 8 presents experiment procedures and results. Chapter 9 is devoted to discussing new findings of each study and their limitations. Finally, I summarize the dissertation and draw conclusions, including future works. Figure 1.1 presents an overall content structure of the dissertation. Note that arrows displaying in Figure 1.1 represent processes, dependencies and data of each system.

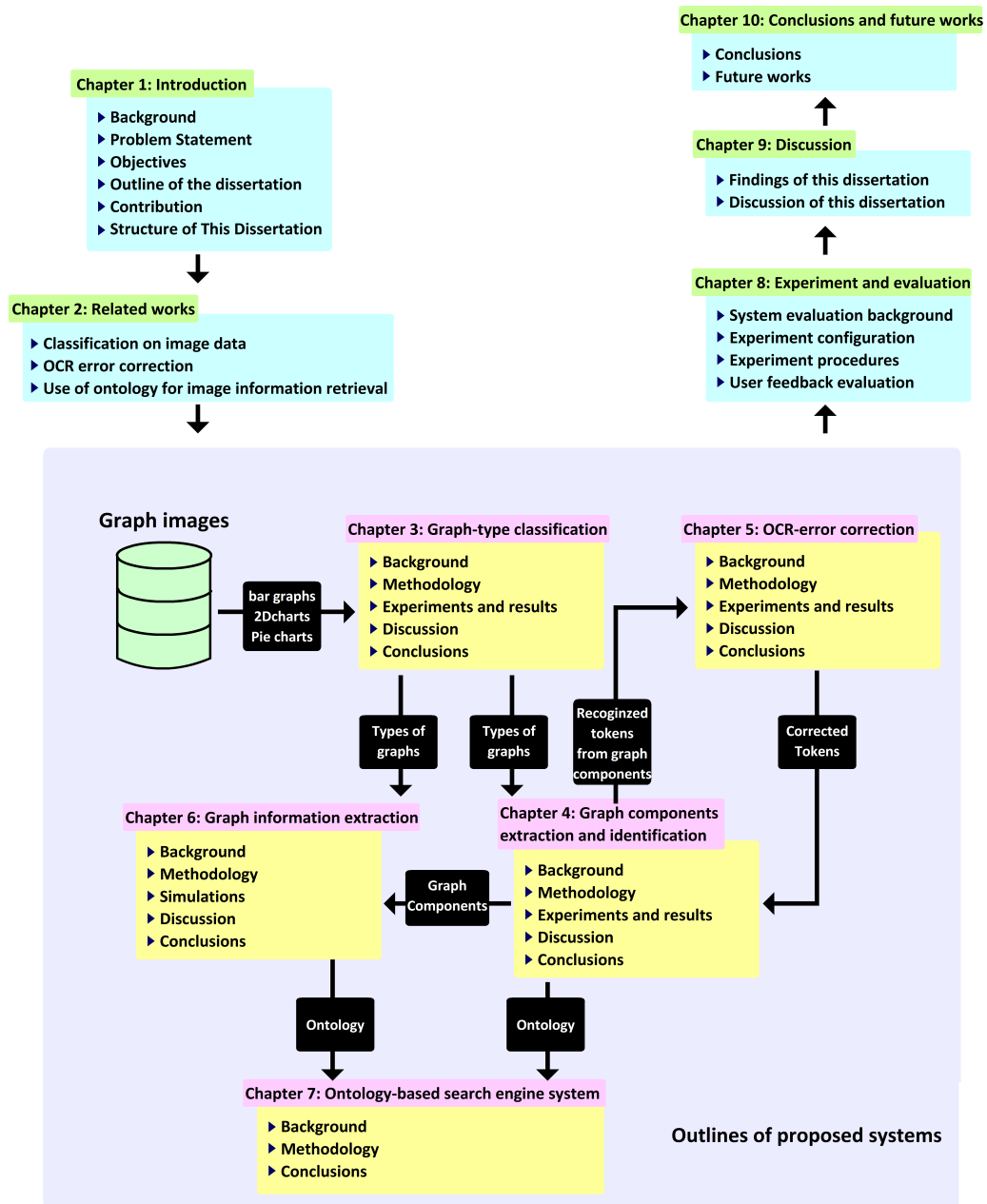


FIGURE 1.1: Content structure of this dissertation

Chapter 2

Related works

In the previous chapter, I described the introduction of this dissertation to describe what my inspiration was and explain what problems were herein addressed, including this research's objectives and contributions. This chapter surveys existing works related to the topic of this dissertation. The review given in this chapter intends to provide an idea of the state of the art of the corresponding areas.

I will review works and thoroughly discuss their studies' solutions and problems. This chapter is organized into three sections: classification on image data, OCR Error Correction, and use of ontology for image information retrieval.

2.1 Classification On Image data

An image is a useful resource that presents meaningful information using the graphical content. Humans can easily grasp the semantical information implied in images without necessarily reading any context appearing in related documents. There has been a significant amount of research on image classification that has made great progress in many different areas, including medical analysis [42, 56] and academia [17, 90].

The image regularly includes a large amount of information; therefore, to use it properly, a suitable approach that can transform the image to usable data should be proposed. Several techniques have been applied to image classification to extract image information and characteristics. During the past few decades, considerable attention has been paid to extending the ideas of extracting image features. A popular method was to extract low-level features instead of using their actual pixel values. Low-level features (e.g., color, texture, and shape) have been reported in many literatures [24, 86, 87]. The visual features were extracted and used as inputs of classification. Sergyan [77] proposed a color histogram based classification approach and contributed to an image search system; however, traditional methods involved the low-level features, are insufficient for my target images, even if the idea of using low-level features was effective in the previous studies, because to classify the graph types, I did not focus on the low-level features, but only the presence of objects in the graphs, such as rectangles and scatter plots, was adequate for graph-type classification.

The Hough transformation [27] was a famous feature extraction technique used to detect objects (e.g., circles, rectangles, etc.) from images. Kwon et al. [59] presented a method for classifying human ages based on facial images. They performed a Hough transform to find a parabolic curve in the human chin. Wavelet transformation was also often employed for image features analysis because it can potentially handle images with different scales and that contain noises [76]. In studies of image classification, wavelet analysis has been applied to images and the distribution of wavelet coefficients is considered to characterize images. Arivazhagan et al. [8] proposed a texture classification system using wavelet analysis on a set of texture images that extracts statistical values, such as mean and standard deviation.

When dealing with images, researchers often encounter a major problem of image classification, called the curse of high-dimensional images. A number of previous studies have developed several potential methods for overcoming this difficulty [1, 81]. Sanchez et al. [75] proposed an image classification on large-scale images including multiple classes. They address the compression of such high-dimensional signatures with two lossy compression schemes: dimensionality reduction based on

hash kernels and data encoding with product quantizers. I realized that the image analysis process was an important procedure for classification because images consist of raw data in an inappropriate input form for existing classification systems. Thus, a preprocessing step used to prepare and construct input data in a ready form became an important procedure in extracting dominant image features. Fan et al. [29] presented a method for classifying of medical images based on their regions by using a nonlinear Support vector machines (SVMs). They built adaptive regional feature extraction and feature selection procedures that consolidate robust features from high-dimensional morphological measurements obtained from brain MR images. Their robust feature selection removed irrelevant and redundant features to improve classification. However,

To enhance image classification, data mining and machine learning algorithms have been gaining importance in recent years. As such, there are some well-known algorithms usually applied to image classification given their simplicity and performance characteristics; these algorithms include Artificial neural networks (ANNs), SVMs, and Convolutional neural networks (CNNs), each of which is described below.

The ANNs is an information-processing paradigm inspired by human neural networks. Veluchamy et al. [87] proposed blood cell classification prediction for normal and abnormal cell classes. They extracted images of blood cells, sorting gray level statistics and algebraic moment invariants, then classified the target images using an ANNs. Frate et al. [30] used an ANNs to remotely sense to distinguish among areas made of artificial coverage including asphalt or buildings, and open spaces, such as bare soil or vegetation. They attempted to apply their classification system to high-resolution images from satellites. These previous studies show that ANNs can handle problems involving high-dimensional images and can work perfectly with nonlinearly separable data.

The main concept underlying SVMs is to find a proper hyperplane, defined by types of kernels, to separate data belonging to multiple categories. Both linear and radial basis function (RBF) kernels have been frequently applied to classification problem via SVMs. Cusano et al. [25] introduced an innovative image annotation tool based on SVMs for classifying image regions into one of seven classes, i.e., sky,

skin, vegetation, snow, water, ground, and buildings, or as an unknown. They used image histograms as be feature vectors. SVMs are effective when handling with low-level features [2], but I concerned about SVMs efficiency when applied to my data because the extracted features of this system were different from existing studies.

In recent years, CNNs have been developed by extending the classical ANNs. Here, CNNs have great potential for image classification and are also faster if a computer conditionally supports a graphics processing unit (GPU) [83]. CNNs can be proficiently applied to large image databases [57]. For example, Kang et al. [47] presented a novel classification based on CNNs to classify document images presented in different layouts. They also employed a technique called dropout to reduce overfitting in the fully connected layers [82]; however, CNNs here had an obvious drawback. Inputs to CNNs should be images comprised edges because the CNNs can give fully effort to analyzes the image by using Gabor filters [66]. In this study, I convolved the input image into one-dimensional images without edges or any objects. Therefore, I needed to conduct experiments to evaluate the performance of CNNs applied to my constructed data.

Finally, from my observations during the existing study, I acknowledge that traditional methods suffered from a problem of parametric, i.e., parameters were very sensitive to changing the value of a parameter changed the results either for the better or for the worse. In this study, I solve this obstacle and obtain the best results that can be produced by my current system.

2.2 OCR Error correction

My focus is to extract information from graph images which often locates in graph components(e.g., axis titles and legend). To extract them separately from the graphs, I need a technique of image segmentation to locate and identify areas where the graph components belong to. This helps my system to obtain only necessary information and avoid unused data such as texts in a data section.

Image segmentation is currently an active research area with several unsolvable problems. This technique can be used to capture and separate dominant objects from image backgrounds. Basically, it deals with many kinds of images, such as outdoor scenes [4, 23] and medical images [37]. In academics, a graph image used to summarize and analyze essential information is another target image for this active field. Bar graphs are my main target in this study. I attempted to separate the basic components to prepare the inputs of OCR-error correction. However, to achieve graph segmentation is difficult for traditional techniques (such as image processing), because positions of graph components are unfixed, especially a legend. A dramatic study addressing this difficulty has been presented by [54]. They aimed to automatically extract elements (e.g., axis labels, legends, and data points) from within a two-dimensional graph and mitigate a problem of overlapping text and data points. They performed an image profiling to detect global features in order to identify coordinate axes. Moreover, they applied an extended K-median to isolate and detect the data points from a curve. However, they confronted a problem when trying to extract a legend. That can be solved by performing a connected component analysis to identify individual letters before applying OCR. The other interesting study is proposed in [38]. whose main targets were to associate recognition results of textual and graphical information in scientific graphs. They individually recognized text and graphical regions of the graph images and then combined their results to achieve a full understanding. However, they encountered OCR errors that were solved by manual correction. Although these previous studies proposed effective methods to extract graph components, it did not identify types of individual components. In fact, each component carries essential information, but its role certainly differs. For example, the X- and Y-titles evince a relationship of the graph. The legend provides particular information regarding data described as data labels. Clearly, to identify the type to each component is surely important for graph interpretation. my graph component extraction can achieve this obstacle. Moreover, I not only extracted graph components using the OCR technique but also addressed an OCR error problem by correcting errors based on my methods.

To obtain information in the graph components basically written by text characters, symbols, and numbers, OCR is unavoidably used. It is a technique to recognize graphical alphabet characters and transform it into digital characters. However, OCR may provide wrongly recognition due to many obstacles, for example, low image quality and unsupported language package for OCR. A great deal of effort has developed many approaches to correct the OCR errors over several years. Nagata [69] emphasized his work to correct misrecognized characters using character shape similarity and statistical language model. He attempted to challenge to Japanese whose sentences did not include word delimiters (e.g., space). However, I realized that this previous study cannot correct such items as acronyms and transliterated foreign words because they often show in English (such as ISO and SONY) that cannot recognize by OCR included by Japanese language package. It differs from my method because ours can correct words universally as long as they appear in the source document.

Semantic-based techniques (e.g., context-based analysis and ontology) are proper solutions addressing the OCR problem. Wick et al. [89] realized that conventional systems identified low-confidence outputs that were insufficient to correct misrecognition errors. They used topic models automatically detecting the semantic context of scanned documents and specified the word frequency to correct the errors. However, a limitation of topic models is high training time required, because users must classify documents to acquire their corresponding topics prior applying OCR. An interesting method related to correct OCR errors is also described in [12]. They developed a context-based method based on Googles online spelling suggestion to correct the OCR errors. They avoided using an offline dictionary because a huge volume of terms needed to gather in a source computer, which consumed a lot of resources. Google is a massive online database containing a large collection of word sequences. It is suitable to be a data source of correcting word suggestion. However, this technique is limited to use via online that need to concern about network availability and efficiency, e.g., speed and bandwidth.

Recent studies addressing the problem of OCR errors tend to use ontology and semantics. Jobbins et al. [46] developed a system of automatic semantic-relation

identification between words in Rogets Thesaurus. This knowledge source contains explicit links between words and related vocabulary items for each part of speech, unlike an ordinary dictionary. Their method depended on Relation algorithm that located semantic relations between words and calculated a relatedness score of each word. However, this technique possibly encountered a difficulty, if dealing with words in a sentence. They may obtain a real-word error in the same category or cross reference. To solve this problem, not only word categories but also sentence dependencies should be used, because each word in the sentence definitely contains at least one dependency linking to some other words in the same sentence. Zhuang et al. [95] introduced an OCR post-processing method based on multiple forms of knowledge, for example, language knowledge and candidate distance information given by the OCR engine. They focused on Chinese characters. A similarity between this existing study and my study is to find candidates depended on similarity distances. However, this previous study was limited to long sentences containing many dependencies, because it used an n-gram supportable contiguous sequence of n items from given sentences.

2.3 Use of Ontology for Image Information Retrieval

Searching useful information is a broad central area in information retrieval and knowledge acquisition. Many applications have been active currently, such as Google [15]. Hearst et al. [36] developed a search engine that provided a way to access biological scientific literature. They used Lucene open source search engine to index, retrieve, and rank the text. Definitely, everyone admits that they are very useful and influent their daily life because data now are online and can be easily searched on world wide web. Although regular search engine offers much information based on the use of keywords, the ontology-based search engine can provide more relative knowledge due to its semantic structure that improves search precision. Gauch et al. [32] introduced an ontology-based method to suggest information navigation using a user profile structured as a weighted concept hierarchy. Their system automatically created user profiles reflecting the user's interests that produce moderate improvements of search results.

Furthermore, not only text [68] but also images can be searched by using ontology. Semantic web ontology and image information extraction offer a new way to annotate and retrieve image data [40]. As presented in [39], Hyvonen et al. considered several situations when users were encountered complicated and semantical images and knew how ontology can be used to realize them. To prove their concept, they implemented a system for image annotation depended on ontology and the same conceptualization. Finally, their system provided a recommendation of semantically related images to users. However, most system dealing images often face the problem of the semantic gap.

Capturing image semantics opens up a new field of study by integrating multi-discipline to overcome the existing problems [26, 67]. A typical solution to minimize the gap is to utilize both graphical and textual information in order to obtain relevant knowledge. Zhao et al. [93] proposed a method to extract the underlying semantic structure of web documents by latent semantic indexing (LSI) for textual information to cluster co-occurring keywords or concepts. Users used a particular keyword to retrieve documents that may not include the keyword but contain other keywords in the same cluster. For graphical content, they extracted low-level image features using color histograms and color anglograms. Chen et al. [22] developed a vertical image search engine integrating both textual and visual features to improve retrieval performance. To bridge the semantic gap, they captured a meaning of each text term in the visual feature space and repeatedly measured the weight of visual features according to their significance to the query terms. Moreover, they considered user intention gap that can infer visual meanings behind the textual queries. The previous studies above conducted experiments and their results showed the improvement of precision and recall.

Chapter 3

Graph-type classification

In the previous chapter, I mentioned about related works and their addressed problems. Many state-of-art studies attempted to propose solutions to solve existing obstacles effectively.

In this dissertation, I also introduced novel methods to address critical problems relating to graph images information. My methodology will be described in Chapter 3 to 7 divided by my proposed systems that have particular purposes and address different problems. Chapter 3 presents an idea of graph-type classification using several techniques to distinguish graph types based on their characteristics discovered in the frequency domain. Next, graph component extraction and identification will be explained in Chapter 4. This system can be utilized to identify and extract graph components which sometimes locate in various positions in graphs, especially graph legends. Chapter 5 will depict regard graph-based OCR-error correction. It uses ontologies to correct OCR results acquiring from the extractable graph components because the OCR result may not be accurate. Next, graph information extraction will be described in Chapter 6. It is a method used to extract necessary information found in a data section of the graph. Finally, I will introduce a prototype of graph-based search engine system that integrates entire proposed systems.

Here, in this chapter, I present the background of graph-type classification including a process of the method. Next, I will describe experiments and results. Finally, I will discuss new findings and conclude the study.

3.1 Background

In past decades, there has been a growing interest in image classification. In much of literature, researchers have encountered problems with image classification and have attempted to find solutions by utilizing a variety of classification techniques, such as SVMs [11, 20] and ANNs [30, 87], as well as including image analysis techniques, such as wavelet transformation. Typically, a large input dataset size is a serious problem in the study of image classification because images, which are inputs of a system, generally contain many features and attributes, in particular, two-dimensional images. In fact, the performance of a classification system is inverse to the size of the input dataset. Many existing studies have developed image classification methods based on large-scale datasets, attempting to somehow mitigate the problem of high-dimensional data, as two-dimensional images [75]. If the two-dimensional images are used in an inappropriate way, this may significantly decrease the classification performance. Therefore, a plausible solution needs to be proposed due to effectively solve this problem.

An initial input used in my study involves a collection of graph images. A graph is a graphical representation of a set of objects, and there are a variety of graph types, such as line graphs, plots and pie charts. Such graphs map dependent or independent quantitative variables and represent the essential content that summarizes the given data. An initial form of my data collection is also two-dimensional image; however, I avoid to use it directly to my proposed method but convert it to more suitable and usable form, which should enhance the quality of my classification.

Low-level features (e.g., color, texture, etc.) are extensively used to classify and analyze generic images [9, 72]. Notwithstanding, these are not essential properties for categorizing graph types, yet other objects, such as lines, plots, and primitive shapes. Furthermore, a crucial problem focused on in this study is the differences

in graph characteristics. Naturally, the same type of graph may include several different objects and characteristics. For example, there are many points in a scatter plot (e.g., see Figure 3.1a), but the positions of these points are certainly different from one scatter plot to another (e.g., compare to 3.1b) depending on the real data. Moreover, there is also a line in the scatter plot shown in Figure 3.1b. Unfortunately, these uncertain characteristics of graphs cause difficulties for traditional classification systems [6, 48]. Assuming that I use convolution neural network CNNs to classify the example images showing in Figure 3.1. Based on a basic process on CNN, it convolves the images using multiple filters and feature maps. A typical random kernel is similar to an edge detector; hence, the edge is an important feature for CNN. Moreover, a result of convolution process provides an output matrix containing edge characteristics. However, it is the inessential property for graph-type classification, yet dominant objects, such as lines and circles. I realize that CNN may be unsuitable for graph images. Regarding my proposed method, I also convolve the graph images from two-dimension to one-dimension, still include significant characteristics, such as a profile of pixel appearance and the focused objects in the graphs. I emphasize to classify the images based on their essential characteristics rather than image features.

This system classifies graph types based on the presence of dominant objects in images. For example, it checks a presence of plots in data space to classify plot graph but does not focus on their plots' positions or direction of correlation because I do not currently emphasize data interpretation, but only classify the types. Concisely, I address the problem of classification with images containing particular characteristics, even they are grouped in the same category. Further, I realize a difficulty of the curse of dimensionality that always occurs when handling to image data. Thus, to minimize the problems, a productive classification method has been introduced to create a new data representative that replaces the two-dimensional image that can handle problems of dimensionality and different characteristics.

The main focus of my study is therefore devoted to a graph classification system that classifies graphs into their types based on dominant different characteristics. My method involves the image convolution that transforms a two-dimensional image into a one-dimensional image while retaining necessary information, including

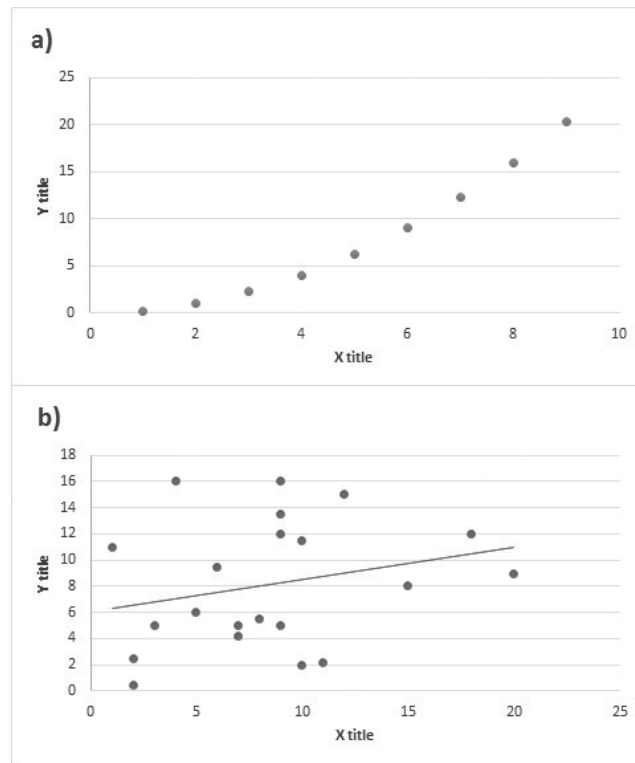


FIGURE 3.1: Example displaying two scatter plots with different characteristics and patterns: (a) scatter plot containing only points and (b) scatter plot containing points in different positions than (a) and a line

constructed numeric datasets that represent results of wavelet and Hough transformations. I propose a new classifier called ANNSVM that is a combination of ANNs and SVMs. This system aims to provide benefits to researchers who give interests to study other's studies and need to prove validation of their studies by comparing their results with other related studies. Demonstrating that they acquire a bar graph illustrating an accuracy of each software from their experiments; thus, to investigate other software and their accuracies in other studies, they need to specify a graph type as a bar graph which is same as theirs for effective comparison. A finding in my previous study [49] already proved that significant knowledge can be discovered in graphs images, such as a relationship between axis titles. In fact, different types of the graph also offer particular knowledge based on their dominant graphical characteristics. Therefore, a process to distinguish graph type can enhance a capability of

image information extraction system [50] and acquire wider and specific information. For example, as the simulated case showing above, its results should be displayed in a bar graph because the bar graph can impliedly express categorized data (e.g., their accuracies corresponding to each software) much better than a line graph. On the other hand, a line graph should be a better choice when dealing with continuous data, such as vehicle speed. The graph-type classification possibly applies to a range of applications, such as search engine and image interpretation systems.

The major objectives of my study are (1) to propose a novel method for classifying graph types with greater accuracy, (2) to extract dominant characteristics of graphs to be a new data representative suitable for identifying graph types, and (3) to indicate what features make my data separable, which improves the quality of a classification system.

To evaluate my proposed method, I conducted several experiments to compare my approach with classical methods, for example, ANNs, CNNs, and SVMs. Both ANNs and SVMs are extensively recognized as being high-potential tools for classification. The important part of ANNs are their ability in learning iterations, which defines how weights of each neuron should be periodically adjusted. The SVMs use a boundary to isolate data that is different depending on a kernel. Here, the kernels tested in this study are RBF and linear kernels. The CNNs is a powerful tool to categorize images by analyzing image edges. The CNNs is based on the ANNs with more one additional step called a convolution step. Typically, the CNNs provides great results as shown in previous studies [61], particularly when working with two-dimensional images containing low-level image features, e.g., image edges, as dominant characteristics.

3.2 Methodology

3.2.1 Definition of my datasets

The target data in this study consisted of a collection of graphs or diagrams. Graphs are classified into three distinguishable types, i.e., bar graphs, pie charts,

and 2Dchart, the latter including line graphs, plot graphs, and area graphs. I merge and set them to the two-dimensional chart because the regular graph structure from those graph types was similar. For example, the line graph and plot graph's structure contains titles that appear on both axes, and there are lines and plots presented in a data section. Note that lines in line graph can be recognized as continuous data plots; thus, it can be likely identified as another kind of plot graph. The graph types also contain apparently own characteristics. For example, for a pie chart, at least one circle should be apparent and reside in the graph; further, axis titles do not appear. For a bar graph, its main structure is comprised of a title along the y-axis and categories along the x-axis. For a 2Dchart, its structure contains titles that appear on both axes. Further, this system is a part of graph information extraction [50]; thus, to precisely extract information, I must use particular methods to extract knowledge depending on the types that contain their own general structures, such as a presence of axis titles and dominant objects. I realize that the same method can be used to extract information from both the line and plot graphs; whilst, another particular method is also specifically applicable to the type of bar graph. Therefore, in other words, I choose the graph types and limit them to three particular types because not only they contain separable graph type characteristics, but it is also convenient for my graph information extraction [50] to accurately extract their information.

In this study, I convert the two-dimensional image dataset, which is squared to a resolution of 64 x 64, to one-dimensional images. The one-dimensional image is a constructed image with a size of approximately 1 x 64 that is applied to my proposed method. Moreover, I create numeric datasets assembling wavelet coefficients and outputs from the Hough transformation. The main motivation for using these techniques is to reduce data dimensionality and gain only necessary information used for classifying. my input data is two-dimension images that contain a huge volume of information; thus, the data size is reduced because only partial information is necessary for classification. For example, the dominant objects in the graphs can be detected by using Hough transformation that is an important information to classify the types. In contrast, image background, such as texts in the graphs, is an inessential part of classification; therefore, it should be ignored.

In my previous study [48], I performed experiments with these image sizes, including other bigger image sizes. As reasonable results, It is found that these sizes (i.e., 64 x 64 and 1 x 64) provided the most accurate results as compared to other sizes because they contain adequate information for classifying. Image sizes that were too large were inappropriate for my experiments because of the curse of high-dimensional data and the problem of sparsity. Similarly, I avoided using image sizes that were too small due to the difficulty of unclear image expression.

To create numeric datasets, I use a wavelet transformation to analyze the one-dimensional images and acquire the sequences of wavelet coefficients based on several wavelet families applied in this study, i.e., Coiflet1, Coiflet3, Coiflet5, Daubechies2, Daubechies10, Daubechies20, Haar, Symlet2, Symlet10, and Symlet20. I select them for two reasons. First, the coefficients can provide significant characteristics for the classification. Each wavelet family has a different oscillation. If the level of the wavelet increases, the wavelet provides much better compression results [43]. Second, they are used in several previous studies [10]. Overall, they have proven to work well for image classification.

All datasets used in this study are summarized below.

- 1Dimg dataset: the original one-dimensional images
- 2Dimg dataset: the converted two-dimensional images
- WL dataset: the numeric datasets that contain only results from the wavelet transformation
- HT dataset: the numeric datasets that contain only results from the Hough transformation
- WLHT dataset: the numeric datasets that contain both results from both the wavelet and Hough transformations

The main dataset I currently emphasize in this study is WLHT, i.e., the numeric datasets that include results from both the wavelet and Hough transformations. To classify images, conventional methods directly use 2Dimg, i.e., the two-dimensional images, as inputs to the systems. Hence, for evaluation, I applied my proposed method to images in 1Dimg and 2Dimg, then compare to my main dataset, WLHT. Finally, I use WL and HT to evaluate which is better for yielding separable data. Note that the method was not applicable to all data in the world. Only bar graphs and 2Dchart were available to the system.

3.2.2 A proposed method

In this subsection, I explicitly propose a new method for graph-type classification using a combination of SVMs and ANNs that include several techniques, such as the discrete Fourier transform (DFT), Hough transformation, and wavelet transformation. I divide my system into two major steps, i.e., a preprocessing step and an application of classification.

3.2.2.1 Preprocessing step

The preprocessing step is a crucial part of my approach. It is used to construct one-dimensional images (i.e., 1Dimg) and numeric datasets containing wavelet coefficients and Hough transformations (i.e., WLHT). To generate the one-dimensional images, the essential procedures are shown in Figure 3.2. Initial inputs of this process are two-dimensional graph images that have already been cleaned and converted to grayscale. My proposed method consists of four steps, each of which is described below.

First, graph images are collected as raw data, which contain different scales and sizes, and therefore need to be normalized. I clean the images by omitting irrelevant areas. For example, I omit unnecessary text that has nothing to do with my classification procedure. Moreover, to standardize the sizes and shapes of the images, I resize and reshape them to be 64 x 64 squares.

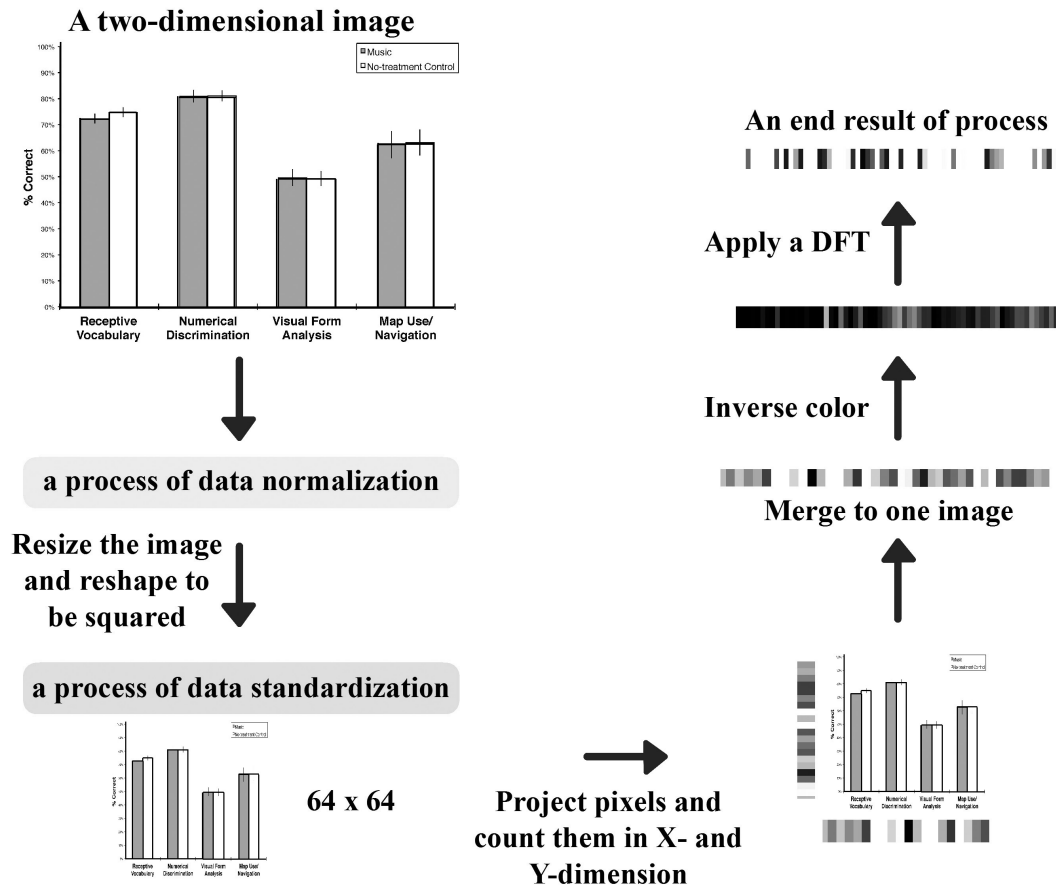


FIGURE 3.2: Illustrating the core process of one-dimensional image construction accomplished by applying a DFT

Second, I examine each image pixel, each of which contains one color value. After each pixel is projected along the x- and y-axes, the number of projected pixels are counted, if a color value is greater than zero, to reduce image dimensionality. Therefore two one-dimensional images should be acquired from the x- and y-axes.

Third, the one-dimensional images acquired from the previous step are combined into one piece by concatenating the one-dimensional image of the x-axis to the one-dimensional image of the y-axis.

Finally, I apply a DFT to the obtained one-dimensional images with inverted colors. I invert the colors here because following the previous step, I gather some

high values along axes, which are represented as white or light colors, that obviously present existing information. Ordinarily, white color indicates no information, while in contrast, a black or dark color instinctively represents meaningful information. Therefore, the process of inverting colors helps to clarify the data expression and prevent confusion in the interpretation.

A DFT is used for two reasons. First, the DFT can uncover dominant information from images but does not need to concern itself with the position of how the objects changed. For example, an important part of pie charts is located in the low-frequency domain because the level of pixel distribution is low. On the contrary, the high-frequency domain presents an important part of plot graphs. Second, a regular factor for classifying graph types is a similarity of characteristics in each type of graph, but my target inputs probably offer different characteristics, even though they are of the same type, because some objects represented in the graphs depend on real data, as shown in the example cases of Figure 3.1.

After completing the process of one-dimensional image construction, numeric datasets including wavelet coefficients and results from the Hough transformation (i.e., WLHT) should be generated. The one-dimensional DFT images are presented in the form of frequencies by the procedure illustrated in Figure 3.2; therefore, they are regarded as signals that can be analyzed via wavelet analysis, which analyzes and decomposes signals into elementary forms at different scales and positions. The prospective results of wavelet analysis here are sequences of wavelet coefficients that represent the similarity extent comparing the examined section of signals to the scaled and shifted wavelets. Note that the wavelet coefficients are calculated at every possible scale and along every position of time. Further, a variety of wavelet families is used to obtain the corresponding coefficients. I apply the wavelet transformation in my study because correlated information in signals is obtained by using wavelet families. Moreover, a sequence of wavelet coefficients is substantially divergent in different images and is considered to characterize images. Thus, the dominant patterns are acquired from various types of images based on different wavelet families.

Hough transformation is a basic technique in image processing that is used to detect features of particular shapes within target images, such as circles, lines, and

single plots. The basic Hough transformation identifies lines in an image, but the later Hough transformation has been extended to be able to identify arbitrary shapes, most commonly circles and rectangles. Moreover, it can deal with the scatter plots in plot graph. Detected objects are counted and collated into the object detection attributes of the given dataset. To reduce variations, I categorize the number of detectable objects into five categories, i.e., C_0 , C_1 , C_2 , C_3 , and C_4 . If the number of detectable objects is between zero and five, I assign its number as C_1 . If it is between six and 10, C_2 is assigned. If it is between 11 and 15, C_3 is assigned. If it is greater than 16, I assigned C_4 . Otherwise, it is set to C_0 . Further, to determine the graph's containing areas, I recognize that the filled areas are often located from middle to bottom with horizontal alignment. Thus, I allocate a specific region inside the image and calculate area density, which is measured by counting the number of color pixels divided by the total number of pixels in the given region. If the density exceeds a predefined threshold the value of the area attribute is set to C_1 . Conversely, if the density is lower than the threshold, the value of the area attribute is set to C_0 .

Comparing 2Dimg and WLHT, the characteristics of WLHT should be more reasonable for my classification system, as opposed to ordinary images, for two key reasons. First, the sizes of images from WLHT are smaller, because I construct my numeric datasets based on the one-dimensional images, thus the number of dimensions certainly decreases. Indeed, the image classification system often provides better results if working with smaller image-scaling sizes [75]. Further, processing time should be substantially less since the number of pixels correlates to the amount of information to be processed. Second, my data can handle problems of different graph characteristics better than ordinary images due to the benefits of DFT.

3.2.2.2 Application of classification

In this study, I propose a new classification algorithm that is a combination of ANNs and SVMs called ANNSVM. As shown in Figure 3, WLHT, i.e., with the one-dimensional images generated in the preprocessing step, serves as input to my classification system. The application of classification consists of two steps, each of which is described below. The SVMs and ANNs models are trained individually.

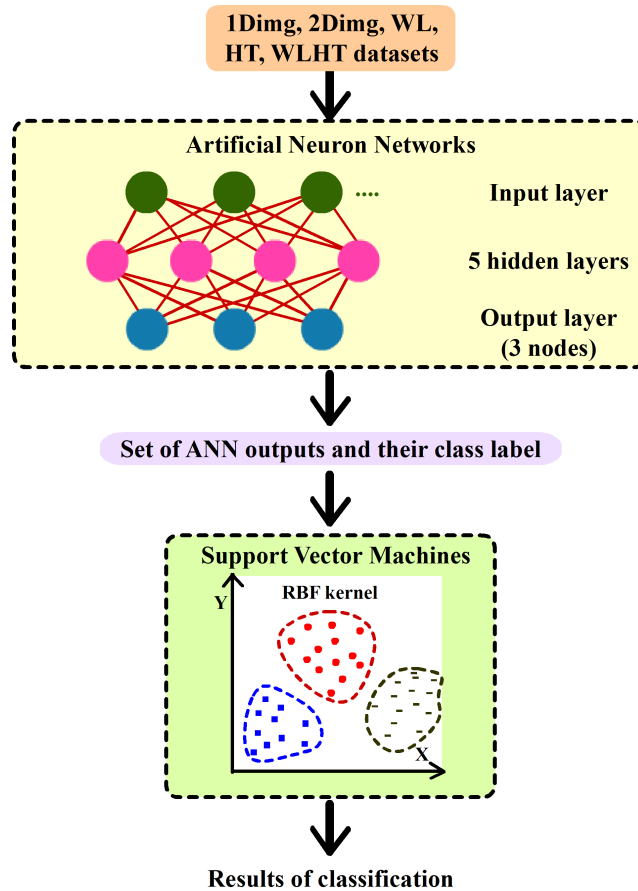


FIGURE 3.3: Demonstrating the process of classification by applying the ANNs, then the SVMs

First, I applied the ANNs to WLHT. To obtain reasonable results from the ANNs, I configured and tuned the following five ANNs parameters: number of hidden layers, the number of nodes in each hidden layer, the number of nodes in the output layer, learning rate, and momentum. Essentially, if the number of nodes in the hidden layers increases, processing time increases, and the resultant ANNs will suffer from overfitting. Conversely, too small of a number of hidden layers will cause underfitting for the ANNs. In my setting, the number of hidden layers and the number of nodes in each hidden layer were fixed at five. Concerning the learning rate and momentum settings, these sensitively impact training performance and are set to optimal values obtained via a grid search technique. The number of nodes in the output layer was

three because there are three different class labels (i.e., 2Dchart, bar, and pie) in my datasets.

I used the ANNs here because my datasets have nonlinear separation, and the ANNs is also highly applicable to nonlinear modeling. Thus the ANNs with multiple hidden layers was an optimal candidate; however, since the ANNs is a black box learning approach, it is difficult to interpret implicit relationships between inputs and outputs.

After this first stage, I used three outputs from the ANNs as new temporary datasets. Note that the three numeric outputs here are the results from three output nodes of the output layer. Typically, these values represent a classification result by justifying a predicted class. For my case, the system must forward proceed to the second classifier, i.e., SVMs; since the system did not justify a predicted class yet but kept the numeric results for being an input of SVMs. Via a training process, I combined them with a class label to obtain training datasets.

Second, I applied a SVMs to the new temporary numeric dataset. The SVMs uses a technique called kernels to find an optimal boundary between the training data. The tested kernel was the RBF kernel. I used this nonlinear kernel because it can capture more complex relationships among data; in contrast, the training time is slightly longer. Fortunately, since my new temporary datasets contain a small number of attributes, the speed of kernel processing was reasonably fast; however, the SVMs using the RBF kernel practically encounters the hyperparameter problem. Significant parameters required for the RBF kernel are cost parameter and gamma. Cost parameter C determines the influence of the misclassification of each training example. If C is large, a boundary correctly classified the training example, but a margin of a boundary is smaller and not smooth. Conversely, a small C provides a smooth boundary but incorrectly classifies more examples. The gamma parameter defines how far the influence of a single training example reaches. A larger gamma represents a close distance from the boundary to support vectors and vice versa. Further, gamma affects the shape of the boundary separating the training examples. these parameters are assigned by optimal results using a grid search [74]. Moreover,

from a practical viewpoint, the SVMs has a high algorithmic complexity that causes the testing phase to be longer.

I used a SVMs classifier for three reasons. First, the SVMs guarantees a global optimum solution, i.e., it can capture the lowest values from a given domain. Second, the dimensionality of the input space does not explicitly affect to computational complexity, still, the smaller data size is surely outperformed. I preferred to use a smaller size of data because this system was a combination of two classifiers. Even though the problem of high dimensionality slightly affects to SVMs, but it might cause bad results to the entire system, in both cases of speed and generation performances. Third, there is a sparse density of pixels in an image, i.e., a low density of pixels that describes information in my one-dimensional image. The SVMs automatically gives a sparse solution, because the Lagrange multipliers are equal to zero for the non-support vector; therefore, the corresponding input vector can be omitted in the summation.

Primarily, no particular classifier exists for all data distributions; furthermore, if there are numerous data, only one classifier may not be discriminative well enough.

In this study, I combined ANNs and SVMs together because using either SVMs or ANNs individually has limitations. Fusion of these two algorithms helps to enhance their abilities of classification and mitigate their drawbacks. Based on the data used in this study, the features in input data had been assembled from various sources, such as results of wavelet transformation and results of a presence of objects provided by Hough transformation; thus, training a single classifier may provide inappropriate results. Moreover, integrating outputs from the multiple classifiers reduces a risk of classification error because the first classifier (ANNs) analyzed the data and provided the results with an empirical data pattern, including some small errors. After that, SVMs would take a place to handle the output of ANNs, which already uncovered the data pattern, and mitigate the errors.

3.3 Experiments and results

3.3.1 Comprehensive tests

In this study, I conducted several experiments to address the various questions of this study, which are described below. The experiments should support a feasibility of the method.

- Which method is the best solution for classifying graph types?
- Which features of the data improve the performance of my proposed method and cause the data to be separable?
- What is the most appropriate dataset to serve as a new representative for classification?
- What significant differences of results exist in my experiments?

I divided my experiments into five major tests that include several minor tests. The CNN_1Dimg and CNN_2Dimg (i.e., Figure 3.4a) was designed to utilize the CNNs with sets of one- and two-dimensional images (i.e., 1Dimg and 2Dimg) to compare performance between it and my main method for 1Dimg and 2Dimg. In SVM_WLHT (i.e., Figure 3.4b) and ANN_WLHT (i.e., Figure 3.4c), I applied the SVMs and the ANNs respectively to WLHT because both of them are popular algorithms used for image classification. I implemented a method that combined these two algorithms, i.e., the SVMs and ANNs approaches, in SVMANN and ANNSVM. The difference between these two experiments was the ordering of the algorithms. The SVMANN (i.e., Figure 3.4d) consisted of the same two steps as my proposed method, but the order of algorithms differed. The first step of SVMANN was to get the raw decision values of the SVMs that presented the actual outputs from the SVMs, which were used to decide which class an instance should belong to. Since I had three different classes in this study, outputs of the SVMs contained three numeric values, which were inputs of the ANNs. Note that I used only the RBF kernel

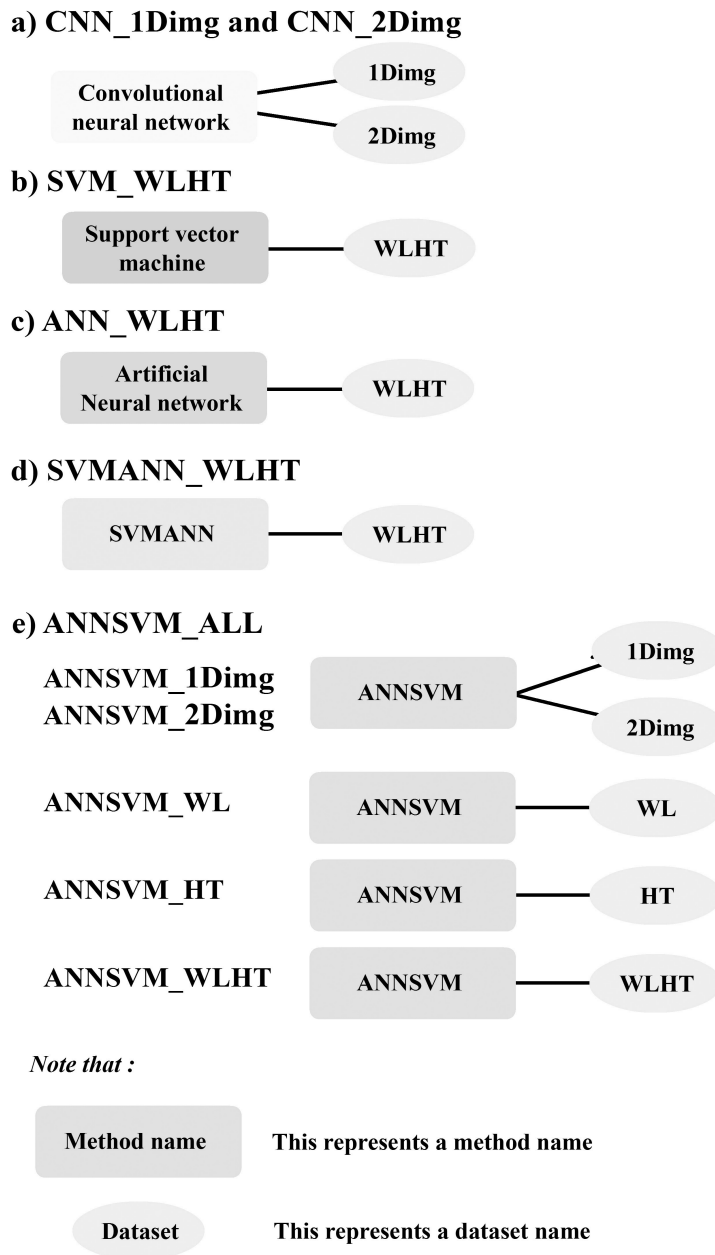


FIGURE 3.4: Processes of all experiments: (a) applying the CNNs to 1Dimg and 2Dimg in CNN_1Dimg and CNN_2Dimg respectively, (b) applying the SVMs to WLHT in SVM_WLHT, (c) applying the ANNs to WLHT in ANN_WLHT, (d) applying the SVMANN to WLHT in SVMANN_WLHT, and (e) applying the ANNSVM to all datasets in ANNSVM_1Dimg, ANNSVM_2Dimg, ANNSVM_WL, ANNSVM_HT, and ANNSVM_WLHT

for this method because it often provides good results when performing with complicated information, and my datasets are nonlinear data. After I combined the outputs of the SVMs with a class label, I applied the ANNs to the outputs and obtained classification results. The last experiment was ANNSVM_ALL (i.e., Figure 3.4e) in which I used my proposed method, i.e., the ANNSVM. Note that ANNSVM_ALL represents the experiments conducted by ANNSVM with all datasets used in this study. In ANNSVM_1Dimg and ANNSVM_2Dimg, the ANNSVM was applied to 1Dimg and 2Dimg to compare results with those of CNN_1Dimg and CNN_2Dimg. In ANNSVM_WL and ANNSVM_HT, I also applied the ANNSVM to WL and HT because I needed to evaluate how data affected the system. Further, the most significant experiment was ANNSVM_WLHT, in which I presented the performance of my main method applied to WLHT to indicate the effectiveness of my proposed idea. To evaluate my approach, I compared results to other experiments that also used WLHT. Regarding SVMANN_WLHT and ANNSVM_WLHT, I conducted experiments to show the difference of performance in a case that I switched their orders. A motivation for rearranging the order of algorithms was to examine whether the results had been influenced by algorithms switched.

In this study, accuracy values of each dataset showed the performance of each method. These values represent are the proportion of the total number of predictions that were correctly classified.

Initially, I classified training instances into three classes that are bar graph, 2Dchart and pie chart, with 303, 322 and 297 images for each class respectively. The number of images is 922 images in total. The graphs had been selectively gathered from the Internet because an amount number of graph images could be collected for training process comfortably. Moreover, I needed the data to find the suitable classification model for the graph images; hence, it should be not matter wherever graph images originally came from. I manually normalized the collected images by eliminating unused areas, such as unnecessary text. Moreover, I evaluated the experiments with tenfolds cross-validation because such an approach can mitigate the problem of overfitting.

Note that I trained the ANNSVM models individually. Each model used independent parameters estimated by SVMs and ANNs parameter estimations. In practice, I applied ANNs to my data. Then, I obtained a new dataset from ANNs outputs, it will be an input for SVMs for classification.

3.3.2 Results

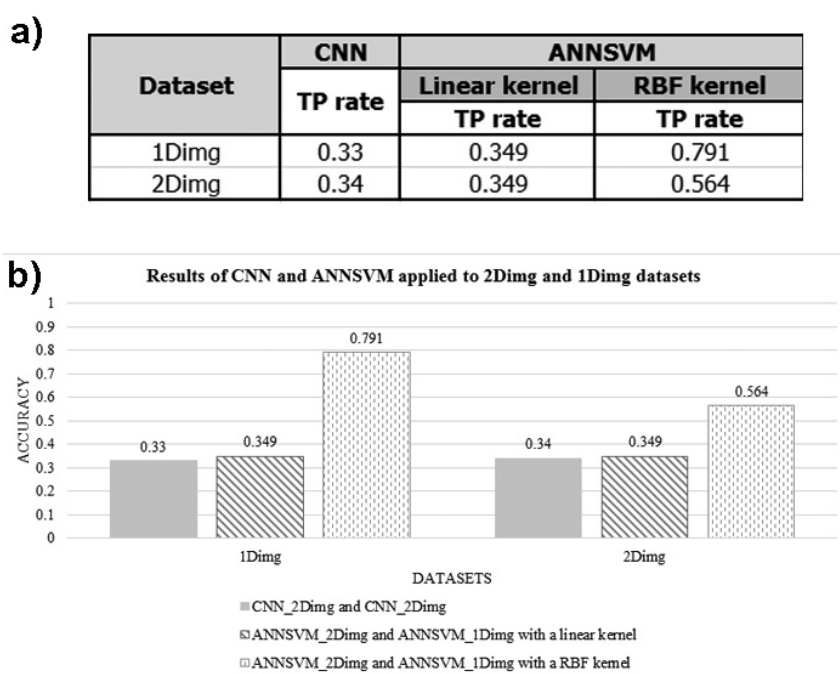


FIGURE 3.5: Results from CNNs and ANNSVM that used 1Dimg and 2Dimg: (a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments

Here, I checked the obtained results by myself. I compared the results of CNN_1Dimg, CNN_2Dimg, ANNSVM_1Dimg, and ANNSVM_2Dimg to confirm the validity of ANNSVM when applied to images. I compared my classification system to CNNs because it is a powerful and popular image classifier. The 1Dimg represented the dataset of one-dimensional images, while 2Dimg represented the dataset of two-dimensional images. Results are shown in Figure 3.5. The CNN_1Dimg and CNN_2Dimg provided similar accuracies, approximately 0.33, which were close to the

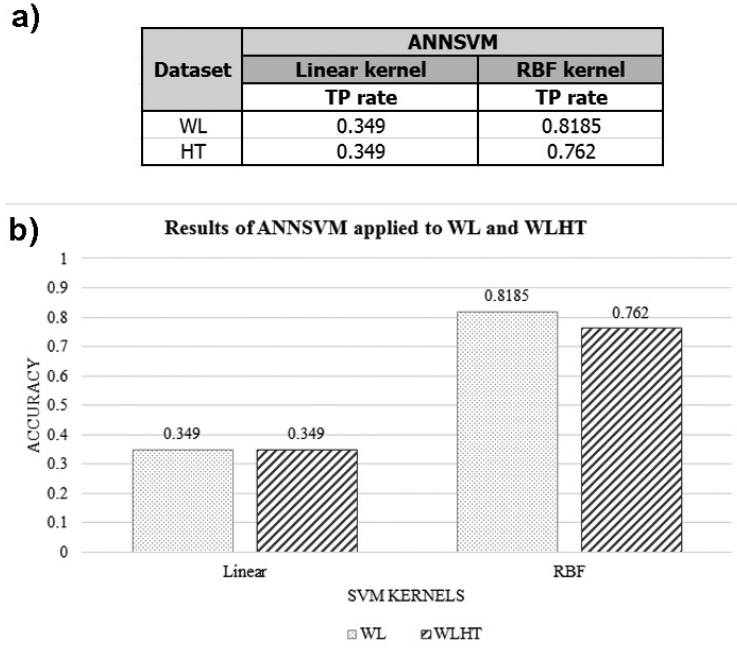


FIGURE 3.6: Results from ANNSVM that used WL and HT a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments

results of ANNSVM_1Dim and ANNSVM_2Dim with the linear kernel, i.e., approximately 0.35; however, the experiment of my proposed method (i.e., ANNSVM with the RBF kernel) presented largely different results. In 1Dim, the accuracy increased to 0.79. Comparing this results to those of 2Dim applied to my proposed method, the accuracy was approximately 0.56. Thus, compared to two-dimensional images, the one-dimensional images were a better candidate for graph-type classification using ANNSVM with the RBF kernel.

To identify which features of data influentially impacted data separability, I conducted experiments for ANNSVM with WL and HT (i.e., Figure 3.6). The WL contained only wavelet coefficients, whereas HT included only results of the Hough transformation. I found that, again, results obtained via the linear kernel were not significant; however, using the RBF kernel, accuracy for WL was higher than that of HT, indicating that wavelet coefficients provide influential features that make data separable.

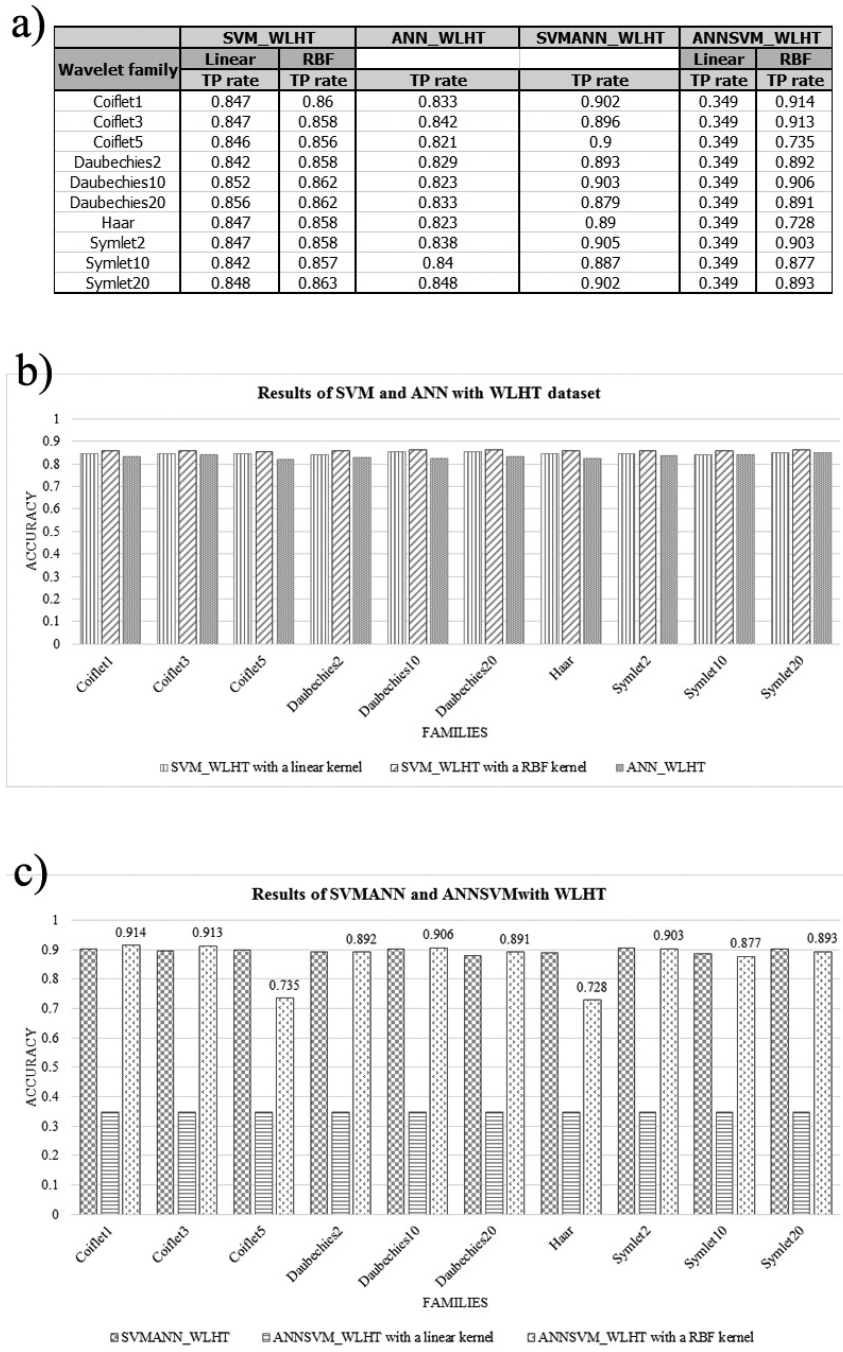


FIGURE 3.7: Results from SVM, ANN, SVMANN, and ANNSVM that used WLHT: (a) table statistically presenting summarized results, (b) bar graph graphically illustrating results from SVM_WLHT and ANN_WLHT, and (c) bar graph graphically showing results from SVMANN_WLHT and ANNSVM_WLHT

I conducted SVMs, ANNs, SVMANN, and ANNSVM with WLHT constructed from my preprocessing method. Results are shown in Figure 3.7.

The results of SVM_WLHT showed accuracy for a RBF kernel as slightly better than that of a linear kernel as indicated in Figure 3.7b. They were on average 0.85 for the linear kernel and 0.86 for the RBF kernel. As for the outcome from ANN_WLHT, it was moderately 0.83. Apparently, accuracy in SVM_WLHT which used the SVMs was slightly more appropriate.

In SVMANN_WLHT, I found accuracies for WLHT were stably high, with an average of 0.9, independent of wavelet families. The highest accuracy in SVMANN_WLHT was 0.905 in the case of Symlet 2. As for ANNSVM_WLHT, the average accuracy was 0.87 for the RBF kernel (i.e., my proposed method) and 0.35 for the linear kernel. Though the average accuracy for my proposed method was slightly lower than that of the SVMANN in SVMANN_WLHT, the highest accuracy among all experiments was 0.91 in the case of ANNSVM_WLHT in which ANNSVM was applied to data obtained by Coiflet 1 (i.e., Figure 3.7c).

3.4 Discussion

Reviewing my results for the CNNs applied to 1Dimg and 2Dimg, I obtained only low accuracies for both datasets. This fact disagrees with other studies but agrees with my assumption. This situation occurred for two reasons. First, my target images were the graphs that contained different characteristics, even though they belonged to the same class. It was difficult for the traditional classification system (i.e., the CNNs here) to reliably classify these data. Second, after it was converted to a one-dimensional image by my preprocessing method, it did not contain any visual image features, only the frequency domain of images obtained via DFT. Since CNNs filters commonly work to detect image edges, it is not suitable for the CNNs to handle my data; however, after observing results of CNN_1Dimg and CNN_2Dimg, I discovered that they provided similar accuracies, i.e., 0.33 and 0.34, for 1Dimg and 2Dimg, respectively. Results of CNN_1Dimg and CNN_2Dimg suggested that both one-dimensional and two-dimensional images contained the same

information. During a convolution process, I possibly obtained the similar convolved images to used in a CNNs classification. This is because the values of pixels in the one-dimensional images roughly substitute for the location of objects in the two-dimensional images. For example, if black pixels, which stand for a larger number of counted pixels, are continuously concatenated in a portion of a one-dimensional image, the DFT decomposes them as low frequencies. As results of ANNSVM_1Dim and ANNSVM_2Dim showed, I obtained highly accurate results from my proposed method, particularly with a RBF kernel. my datasets were not linearly separable, thus a linear kernel did not work well. my proposed method offered substantially better results when it was applied to 1Dim, whose dimensionality was reduced but the important information was preserved.

From the results of ANNSVM_WL and ANNSVM_HT, wavelet coefficients had a larger impact on classification than the Hough transformation data, because the results from my proposed method applied to WL were more accurate than those of HT. The wavelet coefficients can capture the dominant characteristics from the graphs better than the Hough transformation. The one-dimensional image represented in the frequency domain had oscillations with different amplitudes depending on the graph types. For example, a dominant part of a pie chart should be in the low-frequency domain, because there is a large island of concatenated pixels in a one-dimensional image, and it has only a few changes. Conversely, since the scatter plot contains many widely spread points, its dominant part should be located in the high-frequency domain. Performing the wavelet transformation, if a mother wavelet and a part of the wavelet function have a close match, the wavelet coefficient will be large. Assuming that I use a suitable wavelet family with the example pie chart case, the wavelet coefficients in the low-frequency domain should be large as compared to other parts of the domain. These coefficients represented the location of objects in the graph, including their frequencies. The Hough transformation cannot detect the position of objects or frequencies, only the shape of objects. Considering the problem of noise, the wavelet transformation can handle noise better than the Hough transformation, because the Hough transformation is sensitive to noise if the image has low quality.

Using only the wavelet coefficients was inadequate for classification. For example, for the pie chart, I obtained large wavelet coefficients located in the low-frequency domain; however, if I changed a circle in the pie chart to other shapes, such as a radar chart, the wavelet transformation gave results that were similar to those of the original pie chart. The Hough transformation can solve this problem since it detects the shapes of objects.

I therefore assembled these two features in order to make my data more separable.

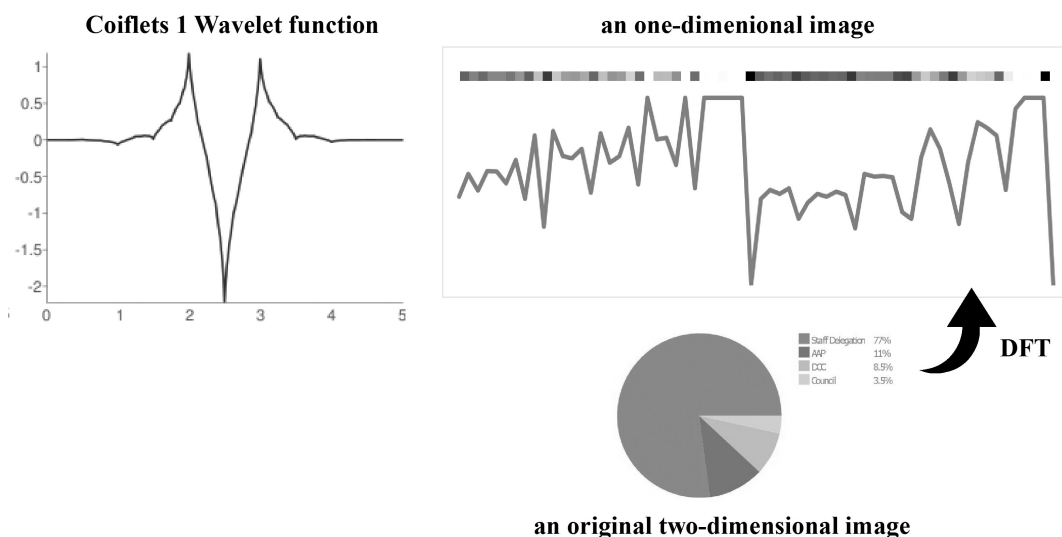


FIGURE 3.8: Simulation of Coiflet 1 [73], analyzing as one-dimensional images

Comparing results from SVMs and ANNs applied to WLHT, the SVMs with the RBF kernel clearly outperformed the ANNs. The difference here comes from the fact that the ANNs can get stuck in local minima, while the SVMs is guaranteed to find a global optimal value. Moreover, the results of each experiment were rather similar. To confirm significant differences between them, I statistically analyzed the results using ANOVA and the T-test. I primarily performed the ANOVA to check the popularity equality, then tested via the T-test for each pair. Finally, I rejected the null hypothesis in all cases, which showed that the results of SVM.WLHT and ANN.WLHT certainly differed.

Further, I interpreted results from SVMANN_WLHT and ANNSVM_WLHT, which were the combination of a SVMs and an ANNs. In SVMANN_WLHT, I consistently received good accuracy values, but the highest accuracy was provided by my proposed method. To analyze results from ANNSVM_WLHT, I compared the linear and RBF kernels. Results showed that the linear kernel was not appropriate for my proposed method because my datasets are not linearly separable, whereas the RBF kernel provided higher accuracy values. The RBF kernel generally outperforms the linear kernel because the linear kernel is suitable if the number of features is larger than the number of instances or a dataset is a very large-scale dataset; however, in general, to obtain a good model, many instances should be employed for training. In this study, WLHT contained 198 attributes and 917 instances, and the temporary datasets produced by my proposed method contained four features and 917 instances. Because of these, the linear kernel was not appropriate. Results of ANNSVM_WLHT suggested that the most suitable wavelet family was Coiflet 1 because the wavelet functions resemble the distribution of frequency in the one-dimensional images, as illustrated in Figure 3.8. Statistical analyses via ANOVA and the T-test showed that there was no significant difference between the results of SVMANN_WLHT and ANNSVM_WLHT with the RBF kernel. Therefore, based on this statistical evidence, I do not need to be concerned about the order of these methods. In other words, both ANNSVM and SVMANN can effectively classify graph images.

From the results of ANNSVM_WLHT, shown in Figure 3.7a, the results of Coiflet 5 and Haar were considerably lower than others in the same experiment, whereas all accuracy values in SVMANN_WLHT were consistently stable. Analyzing these results, I found two possible reasons here. First, the ANNs, which is the first stage of the ANNSVM, is not suitable to provide a temporary dataset that is separable by the SVMs if I input data generated by these two wavelets. Second, the unsuitable mother wavelets were generated from my datasets. The mother wavelet of Coiflet 5 contained triple-high oscillation amplitude (i.e., Figure 3.9a). This mother wavelet was inappropriate for my data because overall my data possibly contained only a few matches with the mother wavelet of Coiflet 5. Moreover, the Symlet 10 (i.e., Figure 3.9b) and 20 (i.e., Figure 3.9c) also provided supportive results that were lower than others in ANNSVM_WLHT (i.e., Figure 3.7a) because their mother

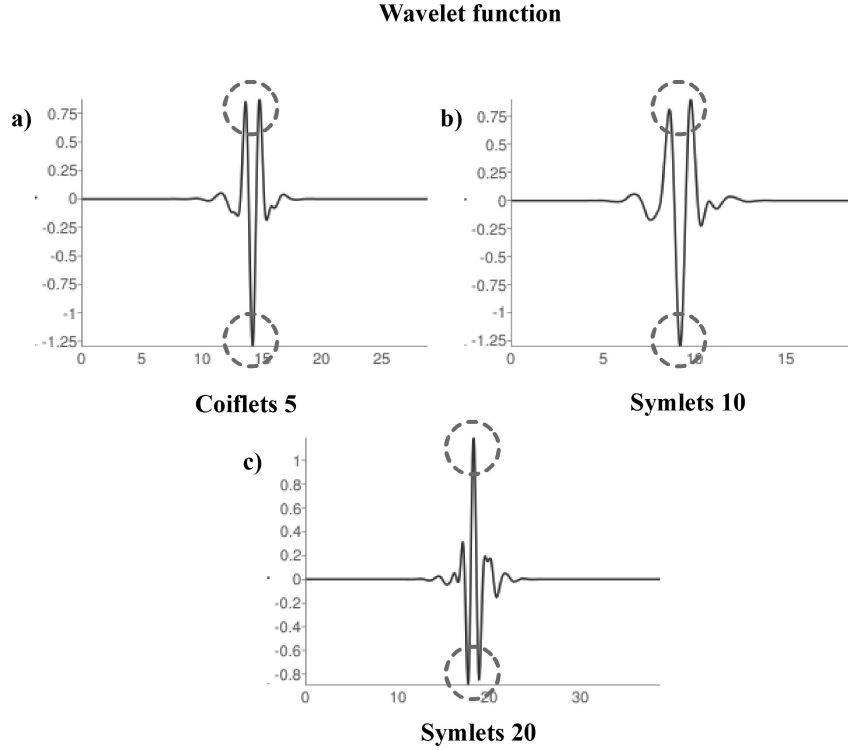


FIGURE 3.9: Illustration of three different wavelets [73] with three waves that have high amplitude values, as indicated in the dashed red circles: (a) mother wavelet of Coiflet 5, (b) mother wavelet of Symlet 10, and (c) mother wavelet of Symlet 20

wavelets also had a similar shape as that of Coiflet 5. For similar reasons, the Haar wavelet was not proper because it is a step function.

After considering the unconventional results from Haar, Coiflet 5, Symlet 10, and Symlet 20 as described above, I again examined the significant differences between SVMANN_WLHT and ANNSVM_WLHT after omitting these wavelet families from my experiments; I did so in order to verify their effects. I performed the T-test on the results without the omitted wavelet families. Statistical results showed that the results of SVMANN_WLHT and ANNSVM_WLHT are equal, even if those wavelets are properly omitted; however, during the T-test, I observed that the average true positive rate (TP) of my proposed method remarkably improved to 0.90 which is greater than the mean of SVMANN_WLHT, i.e., 0.89. From these results, for graph-type classification, my proposed method is clearly more suitable

because the highest accuracy and an acceptable average value were obtained, both outperforming results of SVMANN_WLHT.

=== Detailed Accuracy By Class ===

	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.875	0.875	0.05	0.903	0.875	0.889	2Dchart
	0.95	0.95	0.05	0.902	0.95	0.925	bar
	0.919	0.919	0.029	0.938	0.919	0.929	pie
Avg.	0.914	0.914	0.043	0.914	0.914	0.914	

=== Confusion Matrix ===

a	b	c	<-- classified as
280	25	15	a = 2Dchart
12	285	3	b = bar
18	6	273	c = pie

FIGURE 3.10: Detailed accuracy separated by classes and a confusion matrix which belongs to the dataset of Coiflet 1 applied by my main method (ANNSVM)

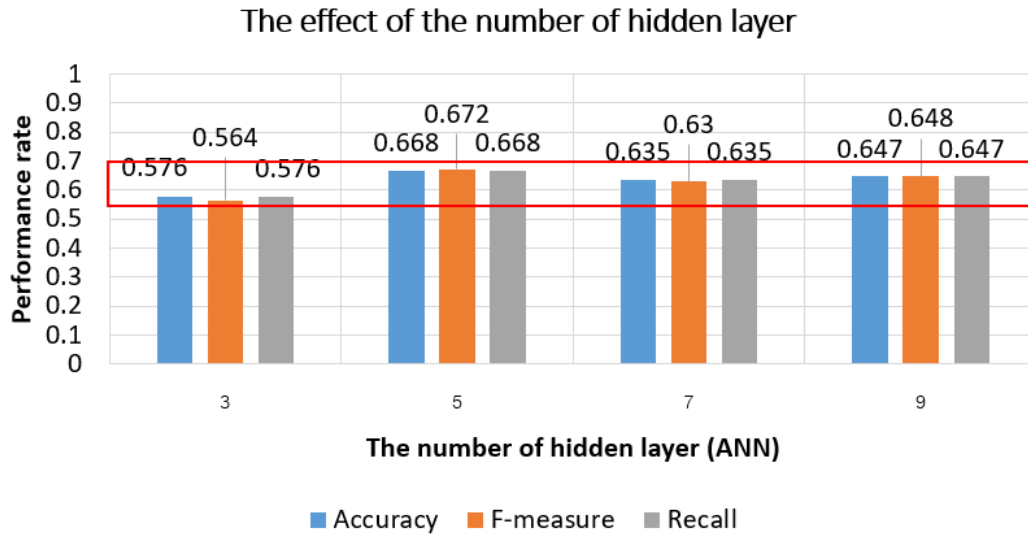
With regard to accuracy values of each class as presented in Figure 3.10, I observed that the accuracy of the two-dimensional chart class was the lowest (i.e., 0.875), while others were over 0.9. Results here suggested that both the bar and pie classes have their own unique characteristics, as opposed to the 2Dchart class. For example, the graph images that contained some rectangles were individually categorized in the bar graph class. A similar phenomenon occurred for circles in the pie chart class. In contrast, the 2Dchart class contained mixed types of graphs; hence, the graph characteristics belonging to the 2Dchart class varied.

Our proposed method is a combination of the ANNs and the SVMs, thus called the ANNSVM. I used the ANNs to construct a temporary dataset, then applied the SVMs for graph-type classification. Results from my proposed method showed that my approach outperformed the traditional methods because when I concurrently use two effective algorithms the strengths of both are encouraged and the weaknesses mitigated. For example, the ANNs suffered from a problem of local minima, but the SVMs strategy solves this problem; hence, the results from my proposed method guarantee the global optimization.

To regard possibilities of this system, I obtained some comments concerning about a complexity of this system. Reducing some processes should be made this system simpler. If I ignore results of Hough transformation and use only results

of wavelet transformation because, based on the finding, the wavelet coefficients identify the dominant characteristics better than Hough transformation, this should increase the speed of the system and reduce its complexity. However, it is important to maintain the system performance after omitting the Hough transformation process. In my idea, I should clean irrelevant image features, e.g., image background, from the images before the classification process in order to emit the graph characteristics as much as possible. The algorithms, i.e., SVMs and ANNs require predefined parameters; to advance the performance of the system, a system of parameter estimation should be assembled and automatically assigned to the system based on input data. Currently, the number of hidden layers in ANNs had been fixed to five layers. Basically, if data is certainly separable, the number of layers can be small. Therefore, based on my finding, even if I increase the number of layers, the classification results may be similar to the results from five-layers ANNs because my data is separable because of a contribution of wavelet coefficients. Moreover, the ANNs uses back propagation with an active function sigmoid to classify data. It is a non-linear activation function great to deal with nonlinear data. Moreover, I conducted a small test to prove my assumption regards to how the number of hidden layers affects to the classification. The experiments used the same dataset and assigned parameters but changed the number of layers. Figure 3.11 presents the experiment results. It shows that even the number of layers is changed, the classification results are similar. In particular, the 5-layer ANN provides the best performance compared to others, as corresponding to the finding of this study. As observed in Figure 3.11, the accuracy and recall values of each the number of hidden layers are same because Weka uses the same formula to measure these values. For each class, an idea to obtain the accuracy and recall are similar. For example, the recall for 2Dchart is equaled to $197/(197+96+27)$, as same to its accuracy. The accuracy is a measurement to identify a correctness of a model; thus, it should compute by using a correct prediction divided by a total number of instances in an actual class. Therefore, for the 2Dchart, it uses the same equation and obtains the same value. With this cause, these two performance measurements use the same formula.

I used the results of wavelet coefficients for classification; actually, they possibly apply to other algorithms, such as clustering. For example, I will use a clustering



5-layer ANN		Predicted			Total
		2Dchart	bar	pie	
Actual	2Dchart	197	96	27	320
	bar	46	217	37	300
	pie	17	81	199	297
Total		260	394	263	

FIGURE 3.11: Results of the tests for checking an impact of the number of hidden layers

algorithm to analyze the graphs belonging to the same group and identify correlated characteristics; moreover, I may realize exceptional characteristics from outliers. Regarding CNNs, As described in the discussion above, it was unsuitable to cope the graph images; however, it effectively classified the photo images. This supports my idea that CNNs should be used to classify the graph types whose dominant characteristics was color, such as pie charts, area chart, and 3-dimensional bar graph.

3.5 Conclusions

In this paper, I proposed a new method of graph-type classification by introducing a new preprocessing step to establish novel representative datasets instead of generic two-dimensional images and a novel classifier based on a combination of traditional algorithms. I conducted several experiments to verify my proposed method and the constructed datasets. Moreover, I extended the scope of my experiments to test data separability. To my knowledge, this is the first study to classify graph images belonging to the same type with different characteristics. I investigated a reliable solution that can be applied to real-world data. Moreover, most results obtained from my experiments showed good agreement with my assumption.

As notes above, I conducted several experiments to evaluate my proposed method. I applied the CNNs to two-dimensional graph images; however, I obtained very low accuracy values from these CNNs experiments. Therefore, I state that the CNNs were not a powerful algorithm for classifying graph types. To compare my proposed method to the CNNs, I applied it to my constructed data, which combines the wavelet coefficients and outputs from the Hough transformation. The dataset consisted of three classes: bar, pie, and 2Dchart, with about 300 images per class. From my experimental results, I obtained the highest accuracy values in my experiments based on my proposed method, up to 0.91. Further, the most proper wavelet family applied to my data was Coiflet 1. As shown in the SVMANN_WLHT and ANNSVM_WLHT, the order of algorithms had not affected the results. It denotes that, regardless of using ANNSVM or SVMANN, the results were acceptable. Obviously, my proposed method has been successful in classifying graph images. Moreover, the difficulty of different image characteristics has been overcome via my approach. The findings of my study suggest that my proposed method greatly contributes to graph-type classification.

In my future research, I will continuously develop my graph-type classification methodology. I will extract significant information from the graphs, such as axis titles and data point labels. The method for extracting such information will be different based on different graph types because of the dissimilarity of graph structures.

Moreover, I will extend my study to be a semantical system using an ontology. The extractable information will be an essential part of creating the ontology. I expect that my future system will be able to extract explicit and implicit information that represent intended relationships hidden in the graphs.

Chapter 4

Graph Components Extraction and Identification

In the previous chapter, I introduced the method of graph-type classification based on each graph types characteristics analyzed by wavelets and hough transformation. My focus of this dissertation was to extract useful information from graphs and minimize the semantic gap. To archive my goal, I needed information establishing at graph components, such as axis titles and legend. However, to acquire them, it is difficult to specific a position of each component, particularly legend, because their locations depend on authors and presenting data.

In this chapter, I describe an effective method to identify and extract the graph components using data mining technique, called Density-based spatial clustering of applications with noise (DBSCAN). It is effective for image clustering because it clusters neighbor objects that are located within a radius of an Epsilon parameter. However, identifying this parameter correctly requires expert knowledge. I propose methods to estimate Epsilon values sufficiently based on the density of each area wherein objects are located in order to extract graph components, such as axis descriptions (e.g., X- and Y-axis titles) and legends. The main objectives are to estimate ϵ with sufficient accuracy to obtain good clusters of graph images and extract the graph components (i.e., X-title, Y-title, and legend).

I will begin by explaining a background of the study and then move to seeing how the method works. I conducted several experiments to evaluate results to prove a validation of the method. Findings will be discussed in a discussion section in this chapter. Finally, I present a conclusion.

4.1 Background

A graph can represent data visually in many different ways, e.g., bar and line graphs, and pie charts. In this study, I focus on bar graphs because they are relatively easy to interpret. Typically, the legend and axes descriptions provide helpful information, such as measurement units that clarify the relationships represented by the graph. Extracting such graph components should contribute to an intelligent system that can interpret latent information in a graph. However, such components, particularly legends, are positioned in various locations, and important graph characteristics are contours and texts. Clearly, to extract graph components is difficult for traditional methods such as spectral clustering.

DBSCAN is a simple data clustering algorithm that is robust against noise [91]. DBSCAN is most suitable for the extraction of graph components. The DBSCAN algorithm requires two predefined parameters, i.e., Epsilon (ϵ), which specifies how close together the points must be to be considered part of a cluster, and MinPts, the minimum number of points required to form a dense region, i.e., within the ϵ distance. In addition to data inputs and clustering procedures, quality of result strongly depends on the values of the parameters. Therefore, determining suitable parameters is time-consuming, because several tests are required to manually examine the most suitable parameter. Moreover, only experts with prior in-depth knowledge about the given dataset can estimate parameter values correctly. To mitigate this difficulty, parameter estimation methods have been proposed [28], [53].

4.2 Methodology

I propose an effective method to extract graph components using DBSCAN algorithm with automatic ϵ estimation. The dataset used in this study is a collection of two-dimensional bar graphs that include X- and Y-titles and optionally a legend. DBSCAN is used because the input images contain many data points intensively packed together in some areas. DBSCAN is a very proficient algorithm when dealing with high-density images.

The proposed method is separated into two parts, i.e., axis description extraction and legend extraction. Note that parameter estimation is included in the legend extraction component.

4.2.1 Axis description extraction

Axis description extraction is based on the actual location of the axis descriptions. Typically, the X-title is positioned at the bottom of the X-axis. Similarly, the Y-title is typically positioned near the Y-axis, usually on the left side of the graph. To obtain the X-axis description, the graph images were partitioned downward and selected the last partition. For the Y-axis description, the graph was also partitioned from left to right and selected the first partition.

However, the initial results obtained from the above process can include irrelevant objects, such as a part of a bar and some numeric values, as illustrated in Figure 4.1a. To address this problem, a pixel projection method should be integrated into my system to eliminate irrelevant parts from prior results (Figure 4.1b). For the X-title, after performing pixel projection in the horizontal direction, positions of peaks were identified. The height of the peaks denotes how many points exist along the horizontal direction. The first peak was neglected; while the rest were retained because the first peak often represents an internal part of a bar. The approach was similar for the Y-title; however, the first peak was retained, and the rest were discarded. Finally, cleaned X- and Y-titles had been acquired.

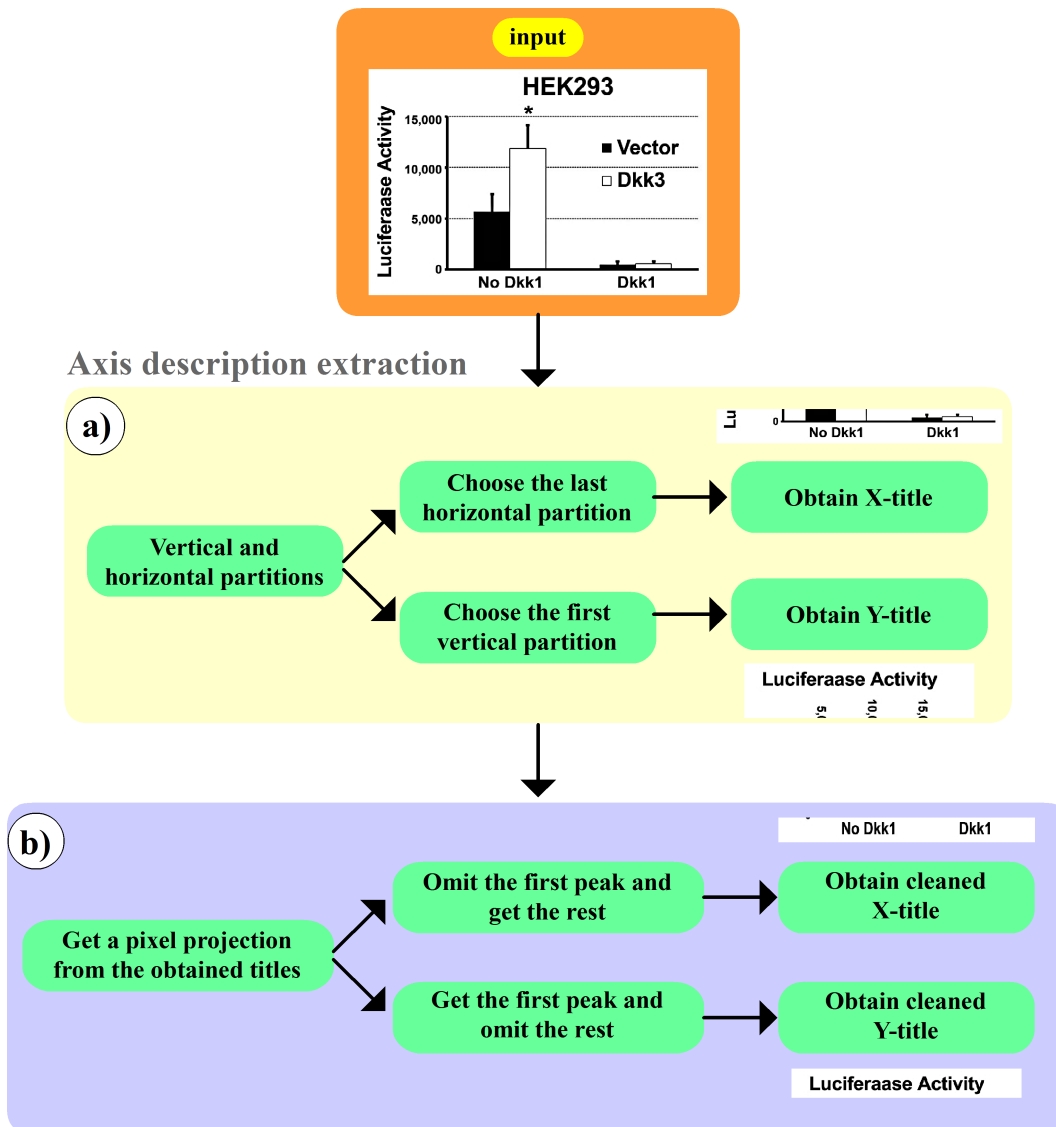


FIGURE 4.1: Process to extract X- and Y-titles from graph images based on their location: (a) image partitioning process; (b) pixel projection

4.2.2 Legend extraction

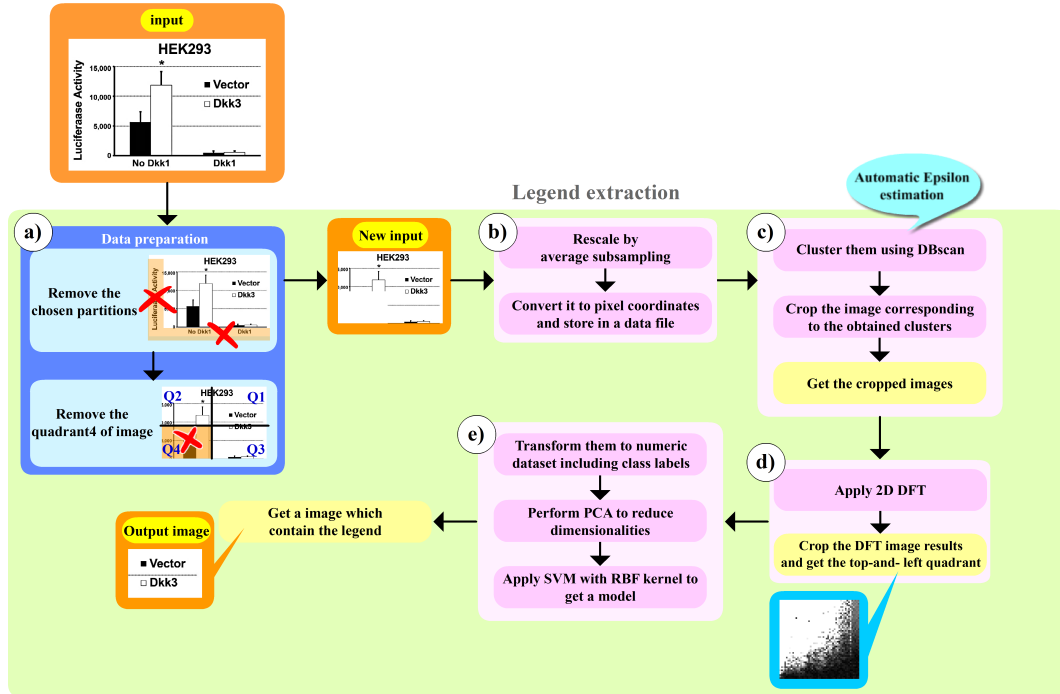


FIGURE 4.2: Overall legend extraction procedures

A legend is a component of a graph that presents data labels. Generally, the legend is optional, and its position is unfixed; thus, a method to extract a legend is more complex than axis description extraction. Figure 4.2 shows a procedure of legend extraction that presents results of each step. I divided this part into five steps: data preprocessing, data transformation, clustering, DFT, and classification. It is very important to preprocess my data because it helps to improve data quality. This process is based on the fact that the legend is typically located at the top-right of the graph rather than the bottom-left. Axis titles were removed because irrelevant parts should be eliminated as many as possible. Moreover, I divided the graph into four quarters: Q1 (top-right), Q2 (top-left), Q3 (bottom-right), and Q4 (bottom-left). After considering the generality of legend position, the Q4 was omitted because the legends are never displayed in this area. Finally, new inputs were obtained for my system. Figure 4.2a illustrates the data-preprocessing step.

The second step is data transformation. After the new input images were obtained, it is difficult for existing systems to process them directly. The images were transformed to a reasonable data format, such as numeric data. However, since the resolutions of the input images are quite high, they were rescaled to smaller sizes using average subsampling, which is a well-known technique that takes the average of a box of pixels. I employed this technique because it is fast and simple. Moreover, the rescaled results also retain the required information. Then they are transformed into numeric datasets. An instance of the dataset is represented by XY-coordinates where data points are located in the given graph. Eventually, numeric datasets for each graph are acquired. Figure 4.2b shows the procedures of this step.

In the clustering step, DBSCAN was applied to the numeric datasets to cluster the data based on their densities, and I cropped the graph corresponding to the clustering results, as shown in Figure 4.2c. Commonly, DBSCAN always requires two parameters, i.e., ϵ and MinPts. The value of MinPts is a constant, and ϵ of the corresponding datasets should be assigned automatically. I designed the method for ϵ estimation based on the empirical fact that, in order to separate objects using DBSCAN clustering, the shortest distance should be found that keep them separated. The target of this study is to detect the specific area containing the legend and extract it from the graph. As mentioned previously, it is usually located at the top-right side of the graph. I introduce my idea with five minor steps systematically.

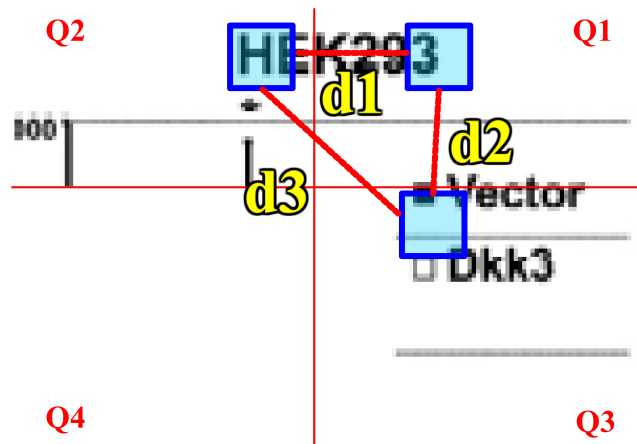


FIGURE 4.3: Epsilon estimation to analyze the densities of each quarter to obtain the smallest distance to be valued as Epsilon

First, a squared window with odd number rows and columns is defined that is used to identify the densities of each shifting area of each quarter. Second, after obtaining the densities, the system must find the highest density of each quarter. However, it is possible to obtain many areas that equally contain the highest density in each quarter. In this case, the density calculation is repeated by using a larger window applied to the resulting areas to determine the new highest density. This operation is continued until the last final area can be defined. Third, when the highest density area of each quarter is obtained, the center of the window is set as the center of the selected area. Then, the furthest point in the same area is investigated, which is measured by the Euclidean distance between the center and other data points of the area to be a representative of the area in each quarter. Fourth, the Euclidean distance among the obtained representatives is computed. Figure 4.3 shows the distance measurement for each quarter. Finally, the shortest distance should be selected, including obtaining ϵ by dividing this distance by the image width. However, there is an exception by which the system cannot obtain any density value in some quarters. This problem had been solved by defining a default value for ϵ . If only a single cluster using the default ϵ is retrieved, the value should steadily decrease until acquiring at least two clusters.

After completing clustering, the part of graph image corresponding to the clustering result is cropped. Note that I use the rescaling method. Thus, when the resolutions to the original size are returned for cropping, the scale may be different than the original image. To address this problem, a margin is assigned by a constant value to expand the leftmost and rightmost clusters.

The fourth step is the DFT process. After the previous step, several cropped images are obtained, including both relevant and irrelevant results. To accomplish my objectives, a relevant part comprising the legend is identified. To support a classification process, I apply two-dimensional-DFT (2D-DFT) to the cropped images to reduce image features in order to expose some dominant characteristics in the frequency domain (Figure 4.2d). With DFT, the image is transformed to its frequency domain. I observed that the characteristics of each quarter of the DFT

image can contain similar information; thus, in order to classify the legend, only a single quarter is selected to be an input for classification, as is shown in Figure 4.2d.

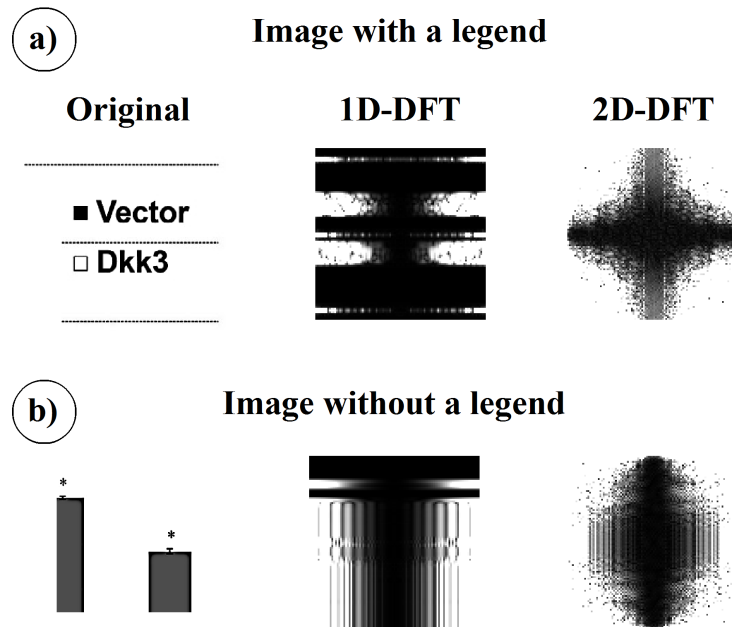


FIGURE 4.4: Examples of DFT results that present the difference between an image (a) with and (b) without a legend

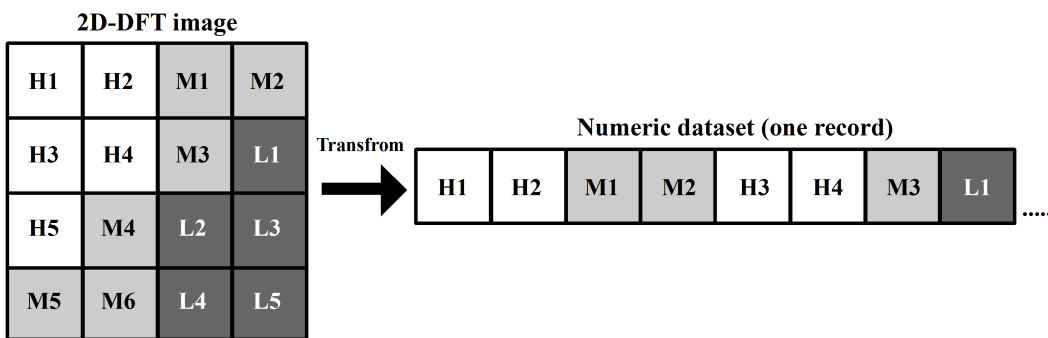


FIGURE 4.5: Data transformation from a 2D-DFT image to a single-row numeric dataset used for classification

According to analyzing the frequency of my data, the dominant characteristics obviously showed in both 1D- and 2D-DFT. Figure 4.4 shows the different characteristics of DFT images based on the presence of a legend. Using 1D-DFT was inadequate for my purpose because it transformed the image into frequency only

along the horizontal direction that does not cover important characteristics presenting in the vertical direction. To address this problem, I applied 2D-DFT to my input data because the image is analyzed in both the horizontal and vertical directions. By analyzing the horizontal direction of 1D-DFT (Figure 4.4a), it can be seen that some white bands appear horizontally. These white bands represent the frequency of text. Note that most values are located in the middle- and high-frequency domains. In contrast, after observing the vertical direction, there are some changes from black to white. With the high-frequency 1D-DFT result, such changes do not occur frequently, whereas the changes in the middle-frequency domain frequently. Thus, I realized that, in 2D-DFT, the dominant characteristics should be located around the middle- to high-frequency domains. With the case shown in Figure 4.4b, significant characteristics were also located in the middle- or high-frequency domains; however, the frequency patterns obviously differ. Thus, it is possible for classifying both cases separately. In my opinion, the low-frequency domain may be unnecessary for classification, particularly for classification of a legend. To obtain better results, this part may be omitted by setting a threshold in order to discard it from the DFT images.

The last step is classification (Figure 4.2e). The results from the previous step are converted to numeric data in order to perform classification. However, the dimensionality of my data was large, which affected classification performance significantly. Data that is too large greatly reduces performance. A simple solution is to use a dimensionality reduction technique before processing the data for classification. Here, I use principal component analysis (PCA) to emphasize the variation of each dimension and identify dominant patterns in a dataset. I analyzed the significant characteristics of the DFT images and realized that, at the middle and high 2D-DFT frequency domains, the variations were significant because there were many frequency changes. To obtain appropriate characteristics, I must retain features that contain high-frequency variation. Clearly, PCA can properly address this problem because it ranks variations and selects the top features. To perform this properly, the number of desired dimensions should be specified, and a new numeric dataset with much lower features is obtained. Then, I apply SVMs to the dataset using a RBF kernel to classify the images with a legend. SVMs is a powerful technique that can handle data whose characteristics are separable by a hyperplane. My

dataset contained high dimensionality and was numeric data. Moreover, as shown in Figure 4.4, the characteristics of my input data with and without a legend are clearly distinguishable. Thus, the SVMs is a good candidate for my classification. Regarding the kernel used in this study, after analyzing the characteristics of my data, I realized that a linear kernel was inappropriate because my data cannot be separated by a linear hyperplane. As mentioned previously, the dominant characteristics of my data are located in the middle or high-frequency domains. When the DFT images are transformed to a numeric dataset, those characteristic features were scattered along a single dataset record, as illustrated in Figure 4.5. According to the distribution of frequencies, it is difficult to use a linear hyperplane to separate the middle or high-frequency domains from the low-frequency domain. Based on my numeric dataset, I use a nonlinear kernel, e.g., the RBF kernel. I observed that the middle or high frequency of each image is often located at nearby features; therefore, the RBF kernel can be used to split such features using a hyperplane.

4.3 Experiments and results

A following experiment would be prove a validity of the system. I conducted an experiment to evaluate whether the proposed method can automatically introduce suitable ϵ values to DBSCAN for effective clustering. I compared the proposed method with automatic parameter estimation (METHOD 1) to a method with a default ϵ value of 0.6 (METHOD 2). I selected a fixed Epsilon value of 0.6 because, after conducting several tests, this was the most suitable fixed Epsilon values to cluster data that could separate data. After conducting the experiment, I checked the evaluation results by myself. The number of graph images used in this experiment was 100 images with 54 images contained a legend. The rest of graph images (46 images) did not contain a legend; since, for high accuracy, my method should recognize them as no legend existed correctly. A type of data was only bar graph collected from academic literatures. Specifically, for this experiment, the data source was PubMed. Also, the data applicable to the system was only graph images whose types were bar graph and 2Dchart. However, I used only a type of the bar graph for this experiment.

TABLE 4.1: Evaluation results of classification for METHOD 1

Method	Accuracy	Precision "Yes"	Recall "Yes"	F-measure "Yes"
METHOD1	93%	79%	82%	81%

Accuracy, precision, recall, and F-measure are discussed in this section. Accuracy is a statistical measurement of how well a classification test precisely identifies corrected instances. A higher accuracy rate means that the predicted values are very similar to the given values. Precision statistically presents the measurement of how many outputs are classified as positives. The recall is another statistical measurement of how well the outputs cover the positives. F-measure is an averaged combination of precision and recall.

For the axis description extraction, I extracted the X- and Y-titles from the input images and manually evaluated the extraction results and verified the number of obtainable and the number of relevant axis-titles. Figure 4.6 shows the accuracy, precision, recall, and F-measure results. Obviously, the proposed method effectively extracted the axis titles. However, after considering relevant results, the precision of both titles was reduced slightly compared to the accuracy rates because some images contained unnecessary parts (such as part of a bar). Meanwhile, the recall rates were remarkable.

For legend extraction, I analyzed both legend identification and classification. Legend identification identifies and extracts the legend from the graph. Here, legend classification indicates how well the outputs are classified as a legend. After checking manually, the results of METHOD 1 indicated that this method gave correct clusters that can be used to properly detect the legend (identification rate of 93%). METHOD 2 gave some unsatisfactory results and a low identification rate, i.e., approximately 31%. I evaluated legend classification using a 50% split of the dataset, i.e., one-half was used to create a model and the other half was used to test the model. Typically, an SVMs with an RBF kernel requires two parameters, i.e., *cost* and *margin* (γ). To define optimal SVMs parameters, I utilized a grid search [74]. I used ten fold cross-validation to validate the classification procedure. For my dataset, I defined the *cost* and γ as 2 and 0.00049, respectively. The accuracy rate

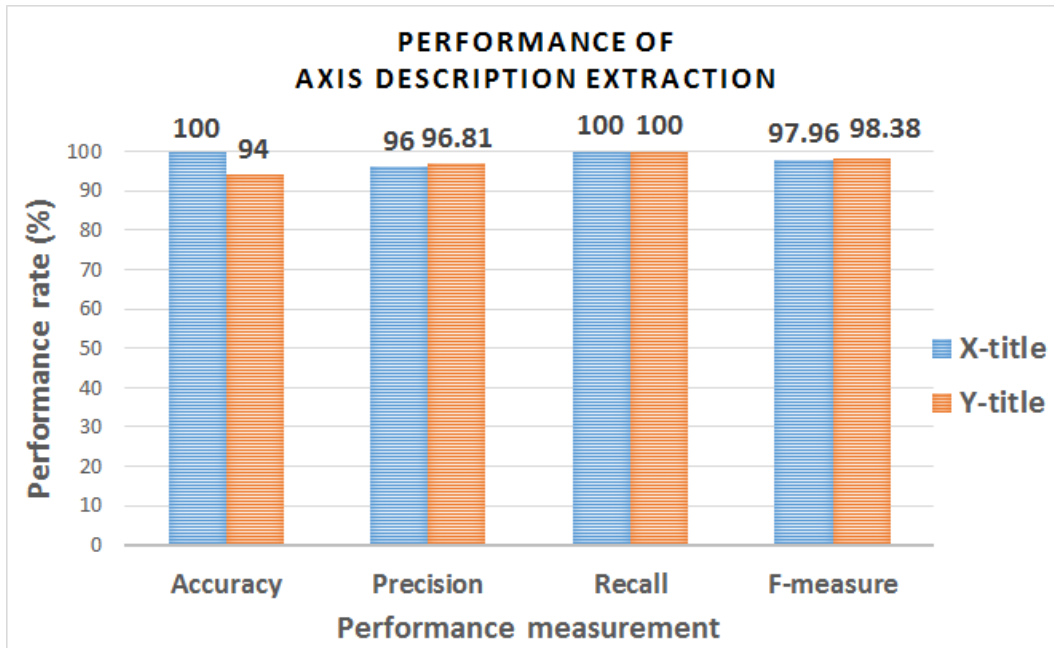


FIGURE 4.6: Performance rates of axis description extraction

of METHOD 1 was 93%, and the accuracy rate of METHOD 2 was 83%. Clearly, METHOD 1 provided much higher accuracy than METHOD 2.

Table 4.1 shows the results of METHOD 1. Note that the class label Yes represents correctly classified images containing a legend.

4.4 Discussion

I have focused on developing a method to extract graph components and have introduced an effective method to automatically identify ϵ values for DBSCAN. This study contributes to my earlier research [4] in an effective manner. In an experiment, I tested the performance of the proposed method by observing the number of axis descriptions and legends that were extracted correctly. Moreover, automatic ϵ estimation was evaluated by comparing another method using a default ϵ value of 0.6.

The accuracy, precision, and recall of my results for axis description extraction were greater than 90%, as shown in Figure 4.6, because the extraction process for the axis description is based on the nature of a graph's structure. The X-title is always located at the bottom of graphs, whereas the Y-title is at the left side. Therefore, my idea is effective to achieve my objective. However, the accuracy rate of Y-title extraction decreased trivially compared to X-title extraction because image noise led my system to misunderstand to select a wrong position to cut a peak. Thus, the error results (e.g., incomplete titles) were presented occasionally. The precision rates were insignificantly lower than the accuracy rates because I found some results that still contained unnecessary parts, even if I had already cleaned them. Such errors were caused by image noise. Note that the relevant result represents the extracted title containing only text, not included the irrelevant parts.

METHOD 1 provided a very high identification rate (93%) compared to METHOD 2. The identification rate of METHOD 2 was low (31%) because most outputs were original images that contained many irrelevant parts. My target is only the part of the image with a legend. The clustering of METHOD 2 could not provide good results. However, METHOD 1 is suitable for extraction of a legend from a graph.

Regarding legend classification accuracy, METHOD 1 provided very high accuracy compared to METHOD 2 because the proposed method (i.e., METHOD 1) introduced suitable ϵ parameters for clustering relative to the input data (METHOD 2 used a default ϵ value). However, the recall of class Yes was slightly low because some graphs with a legend were misclassified. After observing that the obtained results caused these errors, I found that the images contained the legend including irrelevant parts. Thus, when using 2D-DFT, most dominant characteristics came from unrelated areas rather than the legend, which negatively affected classification performance.

Regarding possibilities of this study, this system should be improved if an amount of processed data is reduced. Subsampling is one of an idea to decrease a size of data efficiently. During the clustering process, I realize that the inputted image should be cleaned and removed noise and irrelevant parts beforehand due to avoid clustering errors. Basically, the legend is an optional component; thus, it

does not always contain in the graphs. In this case, an existence of legend should be confirmed in advance by analyzing descriptive information of the graphs. This helps to decide the graphs containing whether legend or not. To regard another parameter named MinPts, it is another parameter defined by users. If this parameter is assigned with a suitable value corresponding to the inputted data, DBSCAN may offer better clustering results. Additionally, I reviewed literature about DBSCAN, I acknowledged another algorithm, named Ordering points to identify the clustering structure (OPTICS) It is developed based on the basic idea of DBSCAN and addresses a problem of detecting meaningful clusters in data of varying density. However, it requires two parameters, i.e., Epsilon and MinPts, as similar to DBSCAN. Assuming that I use OPTICS instead of DBSCAN in this study, the results may be nearly indistinguishable because the suitable Epsilon is used for clustering even either algorithms is applied. This system is used for general purpose; hence, it is possible to freely apply different kinds of data to this system.

4.5 Conclusions

I have proposed a graph component extraction method that uses DBSCAN, 2D-DFT, and an SVMs. I have also introduced automatic ϵ estimation to obtain a suitable parameter value for each input image. I conducted an experiment to evaluate the performance of the proposed method by comparing it to another method that used a default ϵ value. I have investigated an effective idea that can be applied to my previously proposed method [51].

The accuracy, precision, and recall rates of the proposed axis description extraction were greater than 90%. I measured accuracy by checking the number of correct and complete components obtained by my system. However, due to the presence of irrelevant parts, the precision rates were slightly reduced comparing to other rates. For legend extraction, I have discussed the quality of the proposed method using suitable ϵ . Typically, this has been a problem in many previous studies that cannot identify the true ϵ relative to their data. As a result, I conclude that the proposed method can provide reasonable performance comparing to METHOD

2. The accuracy rate of the proposed method was up to 93%, and the precision rates were greater than 80%.

Based on the future of this study, the concept of this method can be applied to other applications. For example, a developer creates a software for measuring population density based on a top-view picture. This method can provide suitable clusters to group residents and estimate the value. Moreover, in the future direction, this method will be possibly improved somehow to deal with image data included background. It is effective to capture the data which stick together, but it may be difficult to use it directly to image data with the background, because the method itself cannot identify which is an object or background.

Chapter 5

Graph-based Optical Character Recognition-error Correction

The previous chapter introduced the method to identify and extract graph components, such as axis titles and legend. I acknowledged that significant information clearly resided at the components; therefore, I needed to realize it by using a tool that can recognize text characters. A typical tool used to transform image-based characters to computer editable characters is OCR. Unfortunately, OCR cannot guarantee perfect results, because it is sensitive to noise and input quality. This becomes a serious problem because misrecognition provides misunderstanding information to readers and causes misleading communication.

In this chapter, I present a novel method for OCR-error correction based on bar graphs using semantics, such as ontologies and dependency parsing. Moreover, the graph component extraction has been used to omit irrelevant parts from graph components. It is applied to clean and prepare input data for this OCR-error correction. The objectives of this study are to extract significant information from the graph using OCR and to correct OCR errors using semantics.

First of all, I will describe background and problem of this study, including some existing researches relating to this topic. I then explain the methodology that

I used in this study. Moreover, experiments and results will be presented, findings will be discussed in the discussion section. Finally, I will summarize the study in a conclusion.

5.1 Background

Nowadays, with the advent of digital optical scanners, a digital document has been required on account of ease of use and search. Over several years, a great effort has been devoted to the study of image-based information extraction. Images, especially graph images, typically contain much expedient information. For example, authors usually use graphs to present their experimental results, including measurement data for clear explanations. Graphs graphically provide data summarization presenting essential information that is simply interpreted by acquiring small descriptive details. Thus, an automatic system extractable latent information from the graphs provides many contributions to society for disclosing explicit and implicit knowledge. To obtain a primary interpretation, initial graph components analyzed are axis descriptions (i.e., X- and Y-titles) and a legend. OCR is an approving solution used for acquiring them as a digital format of character letters.

OCR is extensively used in several applications, such as medical article citation database MEDLINE [60] and text extraction from image and video frames [21]. In regard to academics, countless documents have been converted from paper-based to digitized information using OCR. However, OCR cannot guarantee accurate outputs. Generally, a quality of OCR outputs is fairly decreased, if OCR inputs have various defects, e.g., poor printing quality, small image resolution, specific language requirement, and image noises. These are main causes of misrecognition that produce OCR errors. For example, a word "BED may be incorrectly recognized as "8ED. Regarding the OCR errors, there have been two types of word errors that may be found in my study, non-word and real-word errors [84]. A non-word error occurs when OCR recognizes a source-text as a string that invalidly corresponds to any vocabulary item in the dictionary. A real-word error occurs when OCR applies to the source-text and provides incorrect output strings which coincidentally match to

an item in the dictionary. For example, if OCR renders the source-text "Today is hot as "Toolav is not, then "Toolav is a non-word error, and "not is a real-word error. To mitigate these errors, there has been a great deal of study focusing on addressing them and proposing methods based on practical techniques, such as semantic utilization [46] and statistical similarity measurement [69]. To my knowledge, using the semantics is a reliable solution to alleviate the OCR errors, because it analyzes not only the words themselves but also the context of corresponding sentences. my OCR-error correction method utilizes a concept of semantics, including ontologies and natural language processing (NLP) to identify and correct the errors.

However, OCR is unsuitable to directly apply to graph images, because there are irrelevant parts in the graphs, which do not necessitate for the primary interpretation, such as parts of bars and some numeric data. They may cause recognition noise (e.g., special characters and number); hence, they should be eliminated in advance by my graph component extraction for improving a quality of OCR results [38, 54].

The input of this method is a collection of bar graphs which contains at least axis descriptions (i.e., X- and Y-titles), and optionally, a legend. I here highlight only the bar graphs because its characteristics are dominant and easy to comprehend by both human and machine. Moreover, I gather related contents of documents for creating my ontology, such as image captions and cited paragraphs.

5.2 Methodology

I here propose a novel method which is a combination of a graph component extraction and an OCR-error correction. I also used the method for graph component extraction, which aims to extract and omit irrelevant parts from graph components, presented in the previous chapter. I focus on only three basic components, i.e., an X-title, a Y-title, and a legend as presented in Figure 5.1; thus other parts of graphs are assigned as irrelevant parts that should be omitted beforehand to reduce noise by the graph component extraction. To improve the OCR results, I also present the method of OCR-error correction in this study which is a post-processing system to

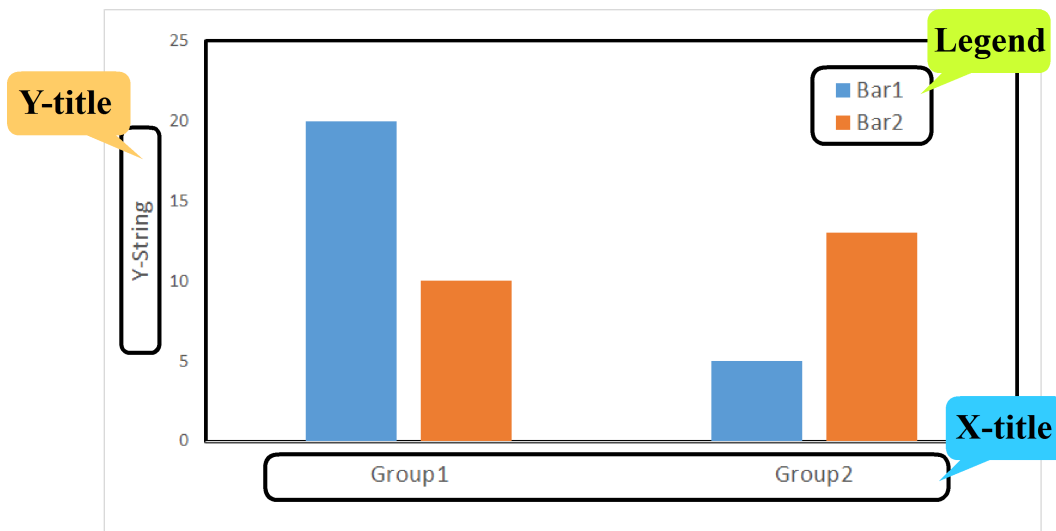


FIGURE 5.1: Illustration of graph components

analyze the results and correct errors based on ontologies, NLP, and edit distance. I designed and created my ontology supporting dependency parsing of English context, including word categories queried from DBpedia.

5.2.1 Data collection

The dataset used in this study is a collection of two-dimensional bar graphs from journal articles. A bar graph is a chart that represents data grouped in categories by bars with lengths proportional to their corresponding values. Typically, a bar graph in my study has two axes, X and Y. For the Y-axis, the bar graph presents an axis title as a sentence, a noun phrase, or a single word. In contrast, the X-axis contains several words representing categories, for example, names of medicines or periods of time. In addition, a legend identifies a label for each bar. Extracting characters from the legend is a challenging task, because its position is changeable, depending on the graph space and the author.

5.2.2 OCR-error correction

On the subject of OCR-error correction, ontologies has been utilized to solve OCR problems. Moreover, an edit distance and NLP are integrated to my correction system, because of an usefulness of sentences context to predict unknown or misspelling vocabulary items based on a word suggestion from the edit distance. My ontology is created to support results of parsed sentences, i.e., part of speech (POS) tags and sentence dependencies, as well as Named-entity recognition (NER) queried from DBpedia. I divide procedures into three steps.

5.2.2.1 Candidate selection

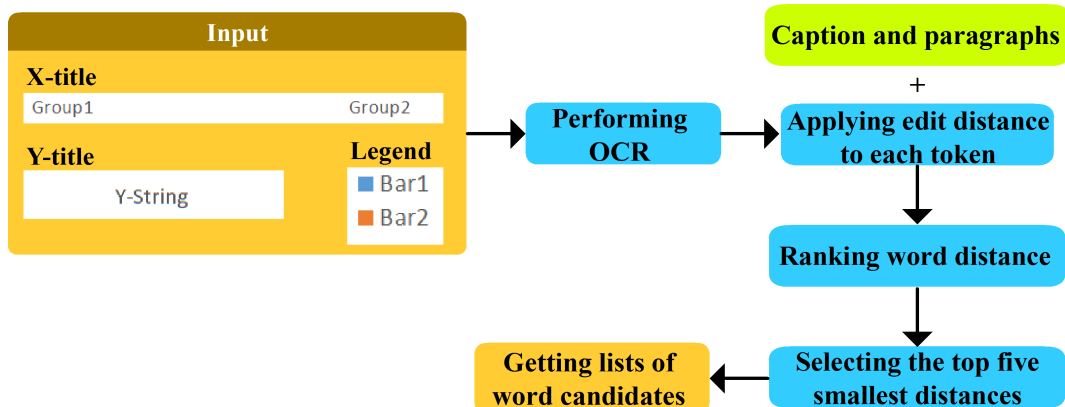


FIGURE 5.2: Steps of candidate selection

The input is the cleaned graph components acquired from the method in the previous chapter. I apply OCR to them and obtain OCR results. I utilize the edit distance technique to measure word distances and rank them in ascending score. Then, the top five words were selected as candidates to be used to replace incorrect OCR results (Figure 5.2). Only five words were collected because this quantity is reasonable for utilization and resource management. For example, a graph image contains a word "well" at its X-title that is incorrectly rendered as "woll". My system can select top five candidates ordered by ascending distance scores as follows: welt, will, wall, well, and with. Obviously, if the number of candidates is too small (e.g., one or three), I definitely miss a correct word "well", which also appears

on the list. Moreover, a high quantity of candidates causes unnecessary processes. Their distances are calculated between tokens from OCR results and the other from corresponding caption or paragraphs. Note that the distance inversely varies to a similarity. A higher distance represents a smaller similarity and vice versa. The selected top five candidates for each OCR token are stored into the list of candidates in ascending order of distance scores. Finally, I acquire the OCR results, including their lists of candidates.

5.2.2.2 Ontology design and creation

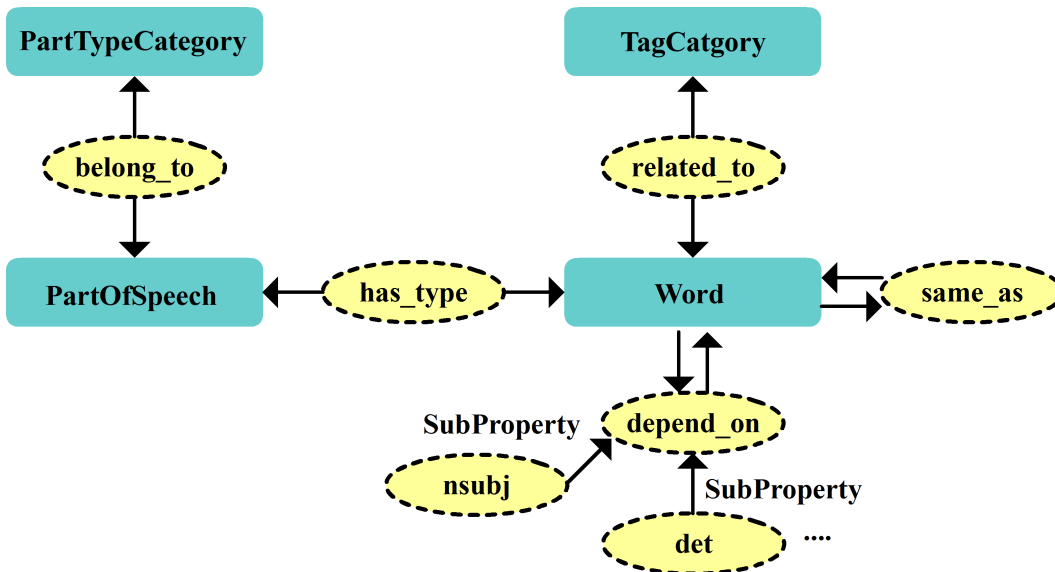


FIGURE 5.3: Demonstration of my ontology structure describing entities, properties, and relations

To fully support my OCR-error correction, my own ontology should be created based on the design, illustrated in Figure 5.3, which includes four entities (i.e., Word, TagCategory, PartOfSpeech, and PartTypeCategory classes) and several object properties (e.g., `belong_to`, `has_type`, and `depend_on`). The Word entity represents every individual token from captions and cited paragraphs of the images used in this study. The TagCategory gathers category names or NER attached to each token, such as person name, location, and animal, by querying DBpedia via its

SPARQL endpoint. Furthermore, I use Stanford Named Entity Recognizer (Stanford NER) to identify the category of the tokens organized into seven categories, i.e., Location, Person, Organization, Money, Percent, Date, and Time. The PartOfSpeech collects POS tagging of each token. For this entity, the total number of individuals is fixed at 36 instances, whose names are from Penn treebank nodes, such as CC, VB, and NNP. The PartTypeCategory represents groups of POS taggings. For example, a singular proper noun indicates NNP belonging to the Noun group. Regarding properties in my ontology, I design several properties, which states relations among entities. The `same_as` represents the relations of at least two synonymous tokens that are stored in the Word entity. For example, Japan and Nihon are synonyms that refer to the same concept. my ontology covers the synonyms expressing the same concept. Moreover, the `depend_on` property is a crucial property because it presents dependency relationships between paired tokens parsed from sentences in the captions and the paragraphs. The number of sub-properties of `depend_on` relations is fixed at 67 properties representing typed dependencies, such as `conj`, `dep`, and `nsubj`. To prepare individuals for my ontology, entire sentences included in the captions and paragraphs are tokenized into tokens. Afterward, I utilize a dependency parser (Stanford parser) to analyze the sentences in order to obtain their dependencies, POS tags, and NER classes. As mentioned above, NER classes are obtained by the parser and the SPARQL query processed in DBpedia. All prepared data are gathered as instances of my ontology.

5.2.2.3 Error correction

The main purpose of this step is to correct the OCR errors using the ontology and the lists of candidates from the previous steps. The basic idea is that the tokens from graph components should appear in corresponding captions or cited paragraphs because authors generally explain information based on the graphs in their documents.

Initially, a dependency dictionary is created, called `DepDic` that records the chain dependencies of the tokens. This dictionary is created, if at least one OCR token identically matches to the first candidate in its own list, and the token is used

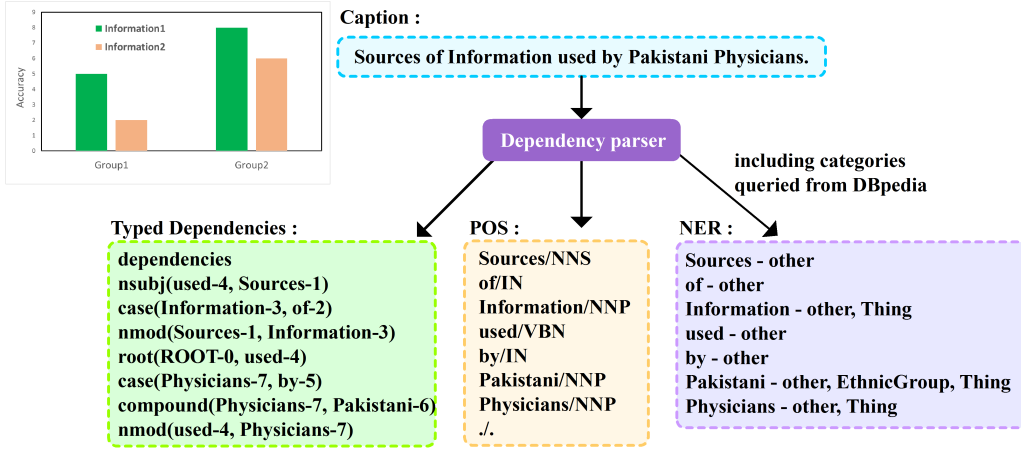


FIGURE 5.4: Example of grammar dependency parsing, including POS tags and typed dependencies, and NER classes of each token

as a head of dependency chains. As an example shown in Figure 5.4, I obtain a word Information from the graphs legend. Suppose that it can be found in its caption, and OCR provides a correct recognition. A parser provides results as typed dependencies based on a caption, including POS tags and NER. I acquire a dependency chain of Information that includes Sources, of, used, Physicians, Pakistani, and by. This chain is recorded into DepDic.

To cover all possible situations for correcting OCR errors, I divided my proposed method into four major conditions, as presented in Figure 5.5.

For the first condition, I focus on eliminating recognition noises from my inputs, such as special characters and numbers. This condition is used to filter unused characters. Numeric characters are represented as recognition noises because the main targets in this study are the axis descriptions and the legend, which generally describe in alphabet rather than numeric characters. Moreover, escape characters (e.g., /, j, and *) should be ignored because they are reserved characters of SPARQL.

The second condition is whether the OCR result finds an exact match in its own list. my system examines the similarity between the OCR result and the first word of its list whose distance is minimal. Typically, the paired words are identical, if the distance score is equal to zero. If this condition is satisfied, a replacement is

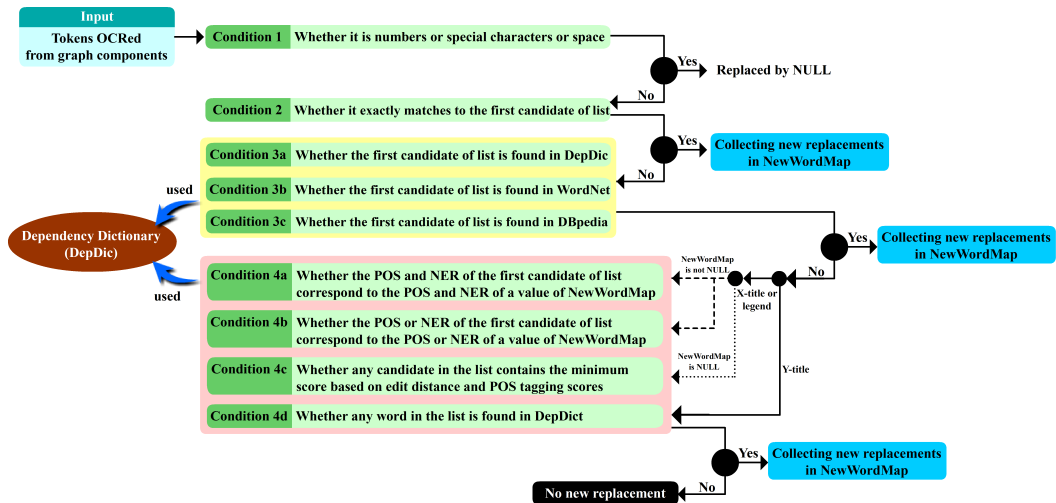


FIGURE 5.5: Demonstration of OCR-error correction covering possible conditions to filter and correct errors

unnecessary due to the accurate OCR result obtained. Afterward, it is collected in a mapping list, called NewWordMap, used to store the OCR results and their new replacements.

The third condition is whether OCR provides a correct recognition that matches nothing in its own list. As aforementioned, my basic idea is that the descriptions of graph components should correspondingly appear in either caption or paragraphs. Unfortunately, the descriptions possibly appear nowhere in the document. Under this situation, I obtain the list of candidates with high distances, which has a low chance to find a match from the list. This condition handles the problem of missing matching tokens by analyzing three minor conditions.

Condition 3a is whether the first candidate of the list has been found in DepDic. Clearly, the first candidate of the list contains the highest chance of matching because of the lowest distance score provided. Moreover, if the candidate is discovered in DepDic, its chance to be selected as a new replacement should be increased. Therefore, if this minor condition is satisfied, my proposed method suggests the first candidate of the list as a new replacement, because not only the smallest distance score is provided, but it also appears in the same chain of dependency. Condition 3b

is processed to check whether the OCR result is actually existed by querying WordNet. Condition 3c has a procedure similar to Condition 3b, but it differs in using DBpedia instead of WordNet. If these two minor conditions are satisfied, I receive no null values returned from their SPARQL endpoints, and the new replacement is not needed. Regards an order of Condition 3, I normally apply these conditions following by this order, Condition 3a, 3b, and 3c respectively. However, if the distance score is over than a threshold, the order of the condition is switched to the following order, i.e., 3b, 3c, and 3a.

The conditions were reordered because I need to prevent errors resulting from the length of OCR result that is lower than a threshold, especially for two or three characters. The short-length words generally represent as prepositions (e.g., in, on, or at), conjunctions (e.g., so or as) and abbreviations (e.g., POS and NLP). Every sentence regularly includes at least one preposition or conjunction, since the short-length words are ordinarily stored in DepDic. Based on this explanation, they may be assigned as an incorrect replacement accidentally. For example, I obtain a word is from OCR, and the first candidate of its list is on already recorded in DepDic. A distance score of them is only two, but their appearances are totally different. Following the original order, the word is is wrongly assigned by the word on as a new replacement. In order to reduce a chance to encounter this situation, a rearrangement of condition orders has been suggested. According to the new condition order, I obtain a correct replacement as the word is itself, because it has been found in WordNet.

The last condition is Condition 4 separately processed based on types of components. Initially, my system checks on the NewWordMap list. If the list is not null, Condition 4a and 4b are processed. Otherwise, Condition 4c is operated. A basic idea of this condition is that X-title and legend should be described by a corresponding category. For example, a financial bar graph contains an X-title, a Y-title, and a legend. The X-title describes product names, which are in the same category (e.g., apple, orange, or banana).

The NewWordMap is available if there is at least one word stored. Condition 4a is whether the POS and NER of the first candidate of the list corresponded to

both of a value stored in `NewWordMap`. To satisfy this condition, I check the POS tag and NER of the first candidate of the list and the POS tag and NER of the word stored in the `NewWordMap`. If their POS and NER are consistent, a new replacement is acquired. Condition 4b is similar to Condition 4a. If either POS tag or NER has been matched, the first candidate of the list is flexibly accepted as the new replacement. Condition 4c is operated if the `NewWordMap` is unavailable or null. I cannot find any comparison from the list; thus, I introduce another solution that utilizes only the list of candidates. This condition is whether any candidate of the list contains the minimum score which sums up from both the edit distance score and a POS tagging score. Regards the POS tagging score, a score to each POS tag is assigned depended on the priorities of word replacement selection based on my experience. The tagging scores are assigned as following: noun (score = 0), adjective (score = 1), verb (score = 2), article (score = 3), adverb (score = 4), preposition (score = 5), conjunction (score = 6), interjection (score = 7), others (score = 8) and number (score = 9). Noun provides the highest chance to appear at the X-title or the legend; since its score should be minimum. I select the replacement assigned by the smallest score, which basically comes from a summation of the smallest distance score and Noun tagging score.

Y-title is described as a sentence or a noun phrase that is different from the X-title or the legend. Tokens from Y-title connect to other tokens by their dependencies; therefore using `DepDic` should be an appropriate option for selecting the most similar word in the list as a new replacement. Condition 4d is whether any word in the list appeared in `DepDic`. Every candidate in the list is iteratively explored in the list of `DepDic` until a match is retrieved and is selected as the replacement.

Otherwise, if I cannot obtain any new replacement from the above conditions, the OCR tokens are used as their own replacements.

5.3 Experiments and results

The experiments supported a feasibility of the method. I conducted experiments to evaluate my proposed method. I divided them into four tests as presented

in Table 5.1. To evaluate the graph component extraction, I compared results obtained from a traditional method and the graph component extraction. The traditional method, namely image partition method, extracted the X- and Y-titles by image partitioning similar to my proposed method, but an idea to extract the legend was different. It extracted the legend by cropping all possible areas where located the legend, such as the top and right side of the image, including irrelevant or relevant parts. A comparison between Experiment 1 and 2 revealed the significant experimental results expressing the performance of the different graph component extraction methods. To evaluate my OCR-error correction, I observed the results from Experiment 1 and 3 to compare the performance between the edit distance technique and my OCR-error correction. I used the edit distance to compare with my system because it reflects the ordering of tokens in the string; and allows non-trivial alignment. These properties make edit distance a good measure in many application domains, e.g., to capture typos for text documents. After OCR processed to the graph components, for the image partition, I approximately obtain 1900 tokens from 100 bar graphs collected from academic literatures, as same to the dataset used in the previous chapter. For my graph component extraction, there were around 1580 tokens. Also, the data applicable to the system was only graph images whose types were bar graph and 2Dchart. The performance of my study was represented in Experiment 4, which was a combination of the graph component extraction and the OCR-error correction.

Several performance rates (i.e., accuracy, precision, recall, and F-measure) were evaluated in this study. The accuracy is a statistical measurement to identify how well a method tests correctly. A higher accuracy rate represents to the consistency of predicted values which are same as given values. The precision is the measurement of given data to present how many outputs are positively classified. The recall is to define how well the outputs cover the positives. F-measure is an averaged combination of precision and recall. Noise ratio is an evaluated measurement to identify how much recognition noises are produced by the system, such as numbers and special characters. The overall measurement results are summarized in Figure 5.6 and 5.7.

TABLE 5.1: Settings of my experiments

Experiment	Method of graph component extraction	Method of OCR-error correction
1	Image partition method	Edit distance
2	Graph component extraction	Edit distance
3	Image partition method	OCR-error correction
4	Graph component extraction	OCR-error correction

As Experiment 1, it was a combination of the image partition method and edit distance. It was said to be a fundamental idea to acquire the information from graphs and correcting OCR results. As the results, all performance rates were presented the lowest values, except the noise ratio. The noise ratio was up to 29.48% that was the maximum ratio comparing to other experiments. However, after examining the noise ratio from Experiment 2 which was a combination between my graph component extraction and the edit distance, my graph component extraction could efficiently handle the noises of irrelevant parts better than the image partition method because the noise ratio obviously presented a lower rate, 19%. Moreover, the accuracy and F-measure were increased to 57.28% and 50.54% respectively, whereas the performance rates of Experiment 1 were only 46.98% and 39.77%. Experiment 3 was a combination of the image partition method and my OCR-error correction. All performance rates were dramatically improved comparing to the first experiment. The accuracy was up to 80.75%, and the F-measure reached to 82.28%. For Experiment 4, I combined the graph component extraction and the OCR-error correction proposed by this study. The performance was better than others. I obtained the highest accuracy rates, 84.23%, and F-measure, 86.02%, including less of recognition noises.

I obtained the number of errors 249 tokens from total 1579 tokens in Experiment 4. I analytically observed causes of errors separated into three types: missing token error, real-word error, and suggestion error. The missing token error presents the number of tokens unable to extract from the graph. The real-word error represents the error of misrecognition but accidentally matches to a vocabulary item in a dictionary. The suggestion error means the error from my system suggesting an

incorrect result. To realize a portion of errors, the percentages of each error proportioned to the total number of errors were presented as follows: 27.71% for the missing error, 37.75% for the real-word error, and 34.54% for the suggestion error. Clearly, among the errors obtained during the experiment, the real-word error and the suggestion error needed to be concerned and mitigated.

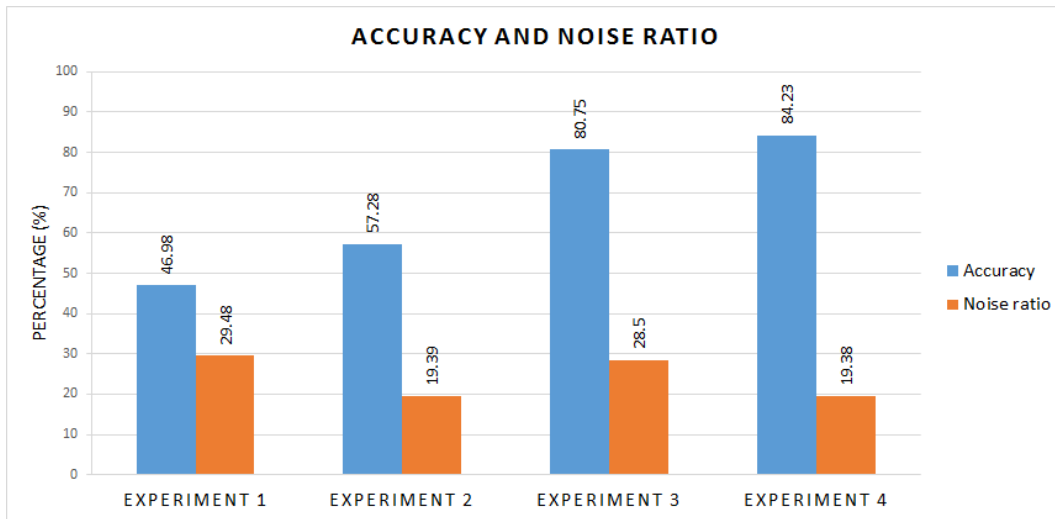


FIGURE 5.6: Illustration of accuracies and noise ratios of all experiments

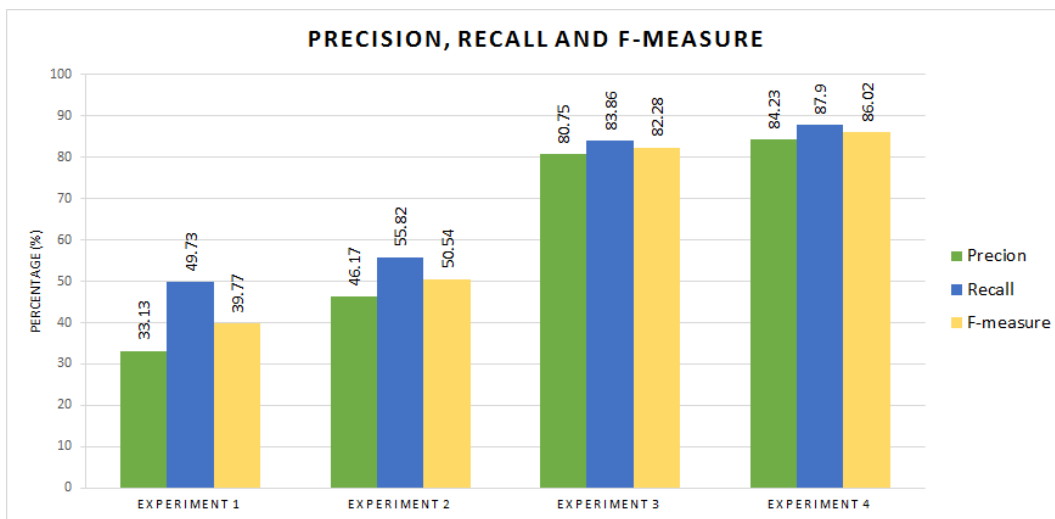


FIGURE 5.7: Illustration of precision, recall and F-measure of all experiments

Moreover, I investigated the number of missing tokens for Experiment 3 and 4. Note that the total number of tokens, which should be able extracted by OCR, was 1165. In Experiment 3, the missing tokens were 151 tokens or 13% of total tokens missing. Meanwhile, in Experiment 4, I obtained missing tokens only 69 tokens (6.92%). Apparently, the missing tokens were decreased, if I applied my data to the graph component extraction.

Figure 5.8 presents accuracy rates of all conditions proposed in my OCR-error correction. Condition 1 used to detect and omit the recognition noises provided the 100% correction. Due to my effective graph component extraction, the accuracy rate of Condition 2 reached 99.15%. Condition 3 presented a reasonable accuracy rate about 81.11%; therefore, using ontologies to investigate a meaning of a word was also appropriate. However, the lowest accuracy was found at Condition 4, 29.47%.

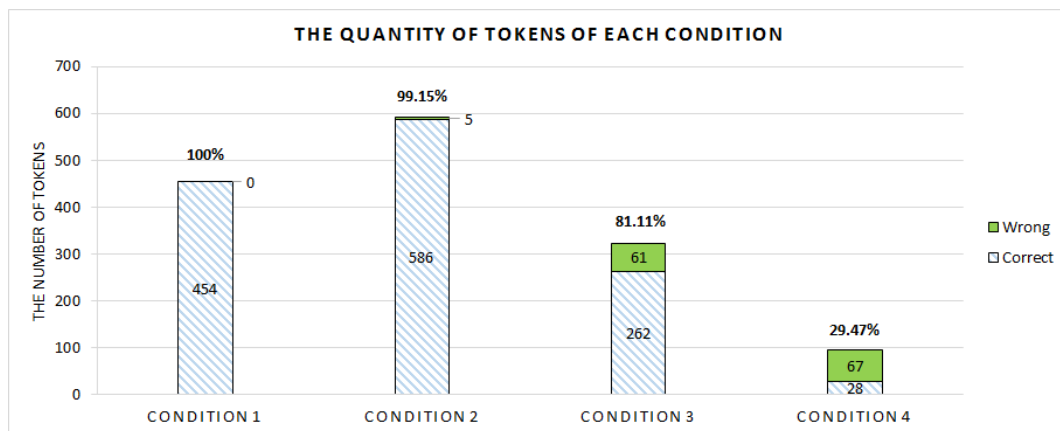


FIGURE 5.8: The number of tokens, including accuracy rates of each condition

Furthermore, I statistically calculated the significant difference between results from Experiment 2 and 4 by McNemar's test. This tool uses for statistically testing on paired nominal data to examine a significant change from two sets of data obtained before and after treatments. In my case, the before treatment referred to Experiment 2 using the edit distance, and the after treatment was my OCR-error correction in Experiment 4. Note that I ignored results corrected by Condition 1 to stable the statistical data because the edit distance cannot handle the noises. I

calculated a two-tailed probability value (P-value), which used to determine to accept or reject a null hypothesis. The small P-value represents a significant difference between two sets of data. The two-tailed P-value is less than 0.0001 that means the results from both experiments are considered to be extremely statistically significant.

5.4 Discussion

I proposed a new method of OCR-error correction based on bar graph images using semantics. I improved my idea by using a graph component extraction proposed in my previous study to reduce irrelevant parts from extracted components before applying to OCR. I aimed to advance the quality of OCR results and proposed an effective method to accurately detect and extract graph components. I conducted four tests to evaluate the performance of my proposed method (Table 5.1). I calculated several performance rates, i.e., accuracy, precision, recall, F-measure, and noise ratio. The experiment representing my proposed method was Experiment 4.

As presented in Figure 5.6 and 5.7, Experiment 1 provided the lowest performance rates, including the highest noise ratio; thus, a combination of fundamental methods (i.e., the image partition method and edit distance) is inappropriate to solve this study's problems. For Experiment 2, I observed that the noise ratio reduced around 10% from the first experiment. Moreover, the accuracy and F-measure had 10% increased. This positive situation happened because I changed the image partition method to graph component extraction. Obviously, the performances are improved, if the noiseless data are used.

Typically, during a recognition process of OCR, it analyzes many objects inside images that sometimes are not character strings. They may lead OCR misunderstanding because they interfere the recognition process and mainly cause errors; hence, my data cleaned by my graph component extraction definitely enhance the performance of the system.

Experiment 3 provided high accuracy and F-measure: 80.75% and 82.28% respectively. To compare between Experiment 1 and 3, the results showed that my

OCR-error correct dominantly affects the performance of the system, because the accuracy rate and F-measure of Experiment 3 were much greater than Experiment 1 about 33.77% and 42.51% respectively.

The performance of my graph component extraction supports the system to reduce noises around 10% as resulted in Experiment 1 and 2. Moreover, according to the different performance of Experiment 1 and 3, it is dramatically increased to around 38% averagely. This evidence contributes the quality of my OCR-error correction that is much better than the edit distance in a case of suggesting the correct tokens because ours can handle a limitation of edit distance.

As mentioned above, the basic idea of edit distance is to use two tokens compared to measure their distance. The comparable tokens are selected from OCR results and the captions or paragraphs. In a case of edit distance providing a correct suggestion, the tokens from graphs are required to have a match to tokens in the captions or paragraphs. However, the tokens may not be mentioned in any contents of documents because of two reasons. First, they are general words that should be known by readers. Thus, it is unnecessary to explain them. Second, they may not directly relate to their topics or studies. Based on this story, the edit distance cannot guarantee the correct suggestion, particularly in the situation of no matched tokens in the captions and paragraphs. On the other hand, my study provided the good quality of OCR-error suggestion which utilized the ontologies to check the meaning of tokens; therefore, it properly mitigates the shortcoming of edit distance. I also obtain a correct suggestion, even if I cannot find any match in the captions or paragraphs.

As the results of Experiment 4, I acquired high accuracy and F-measure: 84.23% and 86.02% respectively; in addition, the noise ratio was decreased comparing to Experiment 1, 19.38%. The accuracy and F-measure of my proposed method were significantly higher than the first experiment about 37.25% and 46.25% respectively. The main purpose of this experiment is to prove the performance of the combination of my proposed methods, i.e., my graph component extraction and

OCR-error correction. An important implication of these findings is that a cooperation of my graph component extraction and OCR-error correction are supportive each other because the performance also improves comparing to other experiments.

I endeavored to compare the results from Experiment 4 to a state-of-art study. Zhuang et al. [95] presented a remarkable study to correct OCR errors based on semantics similar to ours. Their results reported that, after I compared between their method and a basic method, an error reduction was 29%, and the accuracy was 83.73%. Likewise, I analyzed the error reduction comparing between my proposed method and the edit distance. my error reduction was about 27%, and my accuracy was slightly higher than the previous study, 84.23%. Eventually, the idea to use the semantics to mitigate the OCR problem was agreeable, because, to compare to non-semantic methods, the higher performance were presented from my and the previous studies; furthermore, the results were corresponding.

To analyze the errors obtained from Experiment 4, there were three types of errors occurred during the experiments, i.e., the missing error, the real-word error, and the suggestion error. As the results of Experiment 4, the highest proportion of errors was the real-word error. This error happens when the OCR incorrectly recognizes tokens, but it has been accidentally found in ontologies. The possible solution is to use a specific ontology. The ontologies currently used in this study are general ontologies related to English vocabularies. Since an opportunity to get incorrect suggestion should be reduced if I use the specific ontology rather than the general ones because the suggested results relate to a domain of the graph. For example, if a graph relates to biology; thus the biology ontology should be utilized. The next error that I should concern is the suggestion error. It occurs, if the recognized token is incorrect and not found in the ontologies; hence, my system suggests the most similar token which is mostly incorrect, because there is not an identical token appearing in the captions and the paragraphs. One possible solution to this problem is to extend the content of documents to increase a probability to find an identical token, such as using whole contents of the document, not limited to only the captions or corresponding paragraphs. However, it is time-consuming, because there are a lot of tokens that need to measure the distances, including

querying to ontologies. For the missing error, it happens because of the limitation of OCR and a mistake of partitioning process of my graph component extraction. OCR sometimes cannot recognize any characters, if the font size is too small or too big; since OCR returns a null value, which causes a missing token. Furthermore, I partitioned images by a constant; thus, it is possible to cut them at wrong positions. For example, I have a graph containing a two-sentence Y-title. The system may mistake to cut at the middle of the title. I retrieve an incomplete title that causes missing token errors.

To deeply analyze the results of each condition as presented in Figure 5.8, Condition 1 offered the 100% correction. It means that my system has a high capability to detect and omit the recognition noises. For Condition 2, I obtained the high accuracy because of the benefit of my graph component extraction. It extracts the relevant components, which help to enhance the OCR performance to correctly recognize character strings. The reasonable rate is presented in Condition 3. This condition is proved that a viewpoint to use grammar dependencies and ontologies is acceptable. However, most errors occurred in this condition are the real-word error. The final condition is Condition 4 suggesting the lowest accuracy rate comparing to among conditions. The suggestion errors have been mostly found in this condition. The causes and solutions of these errors have been described above.

As the statistical evidence, I concluded that the difference between my OCR-error correction and the edit distance was the significant difference due to a small P-value obtained.

As regards to possibilities of this study, this system may need a support from other ontologies for useful purposes. It is difficult to deal with either meaningless vocabularies or the ones not found in a general dictionary, such as words from mathematics or biology. If other ontologies had been merged to my ontology, this problem should be mitigated. Moreover, a problem of a foreign language should be discussed here. Basically, my ontology had been supported globalization. However, some local tools should be changed due to being compatible with specific languages, such as dependency parser and OCR language pack. To discuss accurately selecting corrected suggestions, there are several ways to improve system efficiency. First, a

suggestion based on an analysis of sentence context may be necessary. Illustrating that, in a sentence describing the weather, there are two words suggested by the ontology. The system should select the one that is most relates to the sentence. Second, I may use Google word suggestion system to select possible candidates. Third, genetic algorithm may be a good option to enhance an effective of words suggestion because it is used for optimization which helps to offer the most suitable word to the system. In this study, I proposed DepDic that was used to records the chain dependencies of the tokens. I have another idea that may help this process. N-gram is a technique to create a vocabulary storage. It decomposes each string in sentences into letters. It should be used to investigate word candidates somehow. Moreover, I discuss a situation that I change the selected ontologies, i.e., DBpedia and WordNet, to others. The proposed system should be still applicable but may need to modify queries because they depend on ontology structures.

5.5 Conclusions

In this study, I proposed the method of OCR-error correction based on semantics, including the graph component extraction. The objectives of this study were to extract their significant information by using OCR and to correct OCR results by utilizing the ontologies and dependency parsing.

From the experiments that have been carried out, it is possible to conclude that my proposed methods achieve the objectives of this study. As the experimental results, my proposed method presented the highest performance rates greater than other methods, and the noise ratio was small. Simultaneously, the fundamental method showed the highest noise ratio and the lowest performance rates, because many irrelevant objects were included in the input data, which interfered the recognition process. Furthermore, the edit distance suggested a correction based on strings appearing in the captions and paragraphs; therefore, suggestion errors certainly occurred, if identical tokens were missing from them. The OCR-error correction introduced in this study contained four conditions to cover all possible situations that might occur during the correction processing. It has clearly shown that my proposed

method can perfectly handle and omit the recognition noises, such as numbers and special characters. Moreover, due to my effective graph component extraction, the cleaned components omitted irrelevant parts were acquired. Indeed, with this data, OCR provided accurate recognitions. Using the semantics to correct OCR errors is also appropriate because it mitigates the problems of missing identical tokens in the captions or paragraphs.

This study can be improved by integrating other sources. For example, I currently used ontology to correct the errors only. To increase the performance of the system, thesauruses or dictionaries can be integrated into the system and together cooperate with the ontology.

Chapter 6

Graph Information Extraction

In the previous chapter, I presented a method to extract graph components and introduced a suitable solution to correct OCR results after text characters from the graph components had been recognized by OCR. However, not only the graph components [49] but the data section of the graph is also useful because most statistical data literately present in the data section. Explicit information such as the relationship between the X- and Y-axes can be easily extracted from a graph by applying human intelligence. However, implicit knowledge such as information obtained from other related concepts in an ontology also resides in the graph. As this is less accessible, automatic graph information extraction could prove beneficial to users.

This chapter presents the novel method for extracting information from the data section of the graph as well as generate ontology to store the extracted data and their relationships. This method can extract explicit and implicit knowledge from graphs. Note that the explicit information obviously shows in the graph, such as axis-titles; meanwhile, the implicit information can be discovered by ontology inquiry. This is based on my ontology that uses essential information pertaining to the graph and sentence dependency parsing. I focus on two graph types: bar graphs and 2Dchart. Different graph types require different extraction methods and have different extractable features. From the bar graph, I extract axis labels, the global

trend in the data, and the height of the bars. From the 2Dcharts, I additionally obtain local trends and regression types. The objective is to propose a method for acquiring the implicit and explicit information available in the graphs and entering this into my ontology.

Herein, I will present the background of the study and then introduce the methodology of graph information extraction. Next, I will describe several simulations based on possible user inquiries. Finally, I will summarize the study in a conclusion section.

6.1 Background

Data reported in the academic literature is presented in many formats, including both digital and hard copy. Although readers must read the literature extensively to comprehend the data, its conclusions may be unclear if only descriptive details are available. Graphs are a form of data representation that helps readers analyze and extract the information they need, making understanding easier. In my previous study [49], I attempted to interpret explicit and implicit information in a graph based on a strong relationship between the X- and Y-axes labels. The information provided by the axis labels includes implicit knowledge; although not presented directly in the graph, it can be extracted by applying ontology. Human readers find it easier to interpret explicit information presented in a graph; comprehending implicit information is more difficult. Thus, a system that allows information to be extracted from a graph can be expected to provide a powerful new approach to knowledge acquisition.

However, I acknowledge that not only the axis labels but also data section can provide essential information. To cover significant knowledge, it is necessary to identify and extract graph statistical data presenting in the data section. For example, this system can identify data tendency and height of bars; thus, I expect to realize wider information when I compare with the previous study [49].

The image data used in this study was a collection of the line, plot, and bar graphs from academic literatures. Specifically, the data source of this study was PubMed. A bar graph represents the data as bars with lengths proportional to their values. Line and plot graphs are called 2Dchart in this study. I selected graphs containing only single data sets to simplify interpretation.

6.2 Methodology

I present a novel method for extracting the explicit and implicit information present in the data part of the graph. I used a combination of techniques, including ontology, OCR, and NLP. I addressed the core problem of the semantic gap by making use of both the context of the graph based on the wider document and the graphical content of the graph itself. A novelty of the study is that my proposed method was able to extract useful information from the data section of the graphs as well as obtain explicit and implicit information from the relationships within the graph.

6.2.1 Ontology

The ontology used was an extension of that in a previous study [51]. As shown in Figure 6.1, it supports not only sentence dependency parsing but also graph components and data extracted from graphs. Protg was used to build the RDF files expressing the ontology. I had already tested its reasoner to validate the generated ontology.

Our ontology included 26 classes and many relations. The main class was the GRAPH class, representing the concept of images from the graph. I used the TYPE class to identify the type of the graph such as bar graph or 2Dchart. The 2Dchart represents two different graph types: line and plot. I merged these into a single type because of their similar characteristics. Lines in a line graph are formed by combining a large number of plotted points.

Most images were described by their captions and optionally by links to paragraphs. These were represented as CAPTION and PARAGRAPH classes, respectively, and were related to a TOKEN class that stored the concepts of the tokens. my system assigned POS tags and NER to each token. I also created dependency relations to represent a typed dependency connecting the tokens in a sentence such as determiner (det) and nominal subject (nsubj).

I identified the basic graph components of axis labels and legends because all graphs use these to represent significant information. For example, the legends of the X- and Y-axes show the relationship between two dimensions. These were therefore made a central part of my ontology. The GRAPH class was related to the COMPONENTS class by a HAS property. The COMPONENTS class comprised three subclasses: X-TITLE, Y-TITLE, and LEGEND. Note that I used only graphs presenting a single data set so that the legend, which shows data labels, was not always essential.

The real information appears in the data presented in the graph and was recorded as a DATA_PART class. This part of the graph displays a graphical representation of the data, for example by the height of the bar or the slope of the line. The data in a bar graph is represented by rectangular bars corresponding to the categories shown in the X-axis title. A BAR_HEIGHT class was introduced to represent the bar height. 2Dcharts use plots to show statistical data in a dimensional space. my approach explored the types of lines used (e.g., linear or non-linear) to represent the data in the graph. This helped predict unseen directions in the data and provide new information that was not described in the caption and paragraphs. I also analyzed and collected both global and local tendencies in a SLOPE class comprising three different trends: an increase (INCREASE class), a decrease (DECREASE class), and no change (STATIC class). The global tendency represents the overall trend in the data while the local tendency provides information about where and how the trend changes. These concepts were described in a CHANGE class.



FIGURE 6.2: Overall of proposed system

6.2.2 Extraction of graph information

6.2.2.1 Data content identification

I first identified the existing graph components (e.g., X-axis title, Y-axis title, and optionally the legend), including the actual data. As different types of graph provide different information, my system needed a method for analyzing information from each type. Figure 6.2 demonstrates an overall of proposed system.

The features generally used for interpreting a bar graph are the X-axis title, the Y-axis title, the height of the bars, and a global tendency corresponding to the centers of the bars. To extract the graph components, the graph image must be partitioned horizontally to acquire the X-axis title and vertically to acquire the Y-axis title. I used OCR to recognize these. However, the occasional presence of irrelevant information such as parts of the bars or numbers may cause misrecognition by the OCR. To address this, I applied a method of automatic graph component extraction described in my previous study [52]. This method uses a technique of pixel projection to obtain a horizontal profile and removes unnecessary information. This provided cleaned graph components. To interpret bar graphs, I analyzed the height of the bars and the categories on the X-axis.

Our system was able to extract the height of the bars automatically, as shown in Figure 6.3. After acquiring the cleaned X-axis legend, I used pixel projection with vertical profiling to locate the positions of the bars and their labels. Note that the position of the bars and the labels correspond. When identifying the height of the bars, I applied a step function to smooth the results of the pixel projection and find the center of each bar. A specific range was measured, equal to half the distance between two neighboring centers, which independently covered each center; the value of the highest peak within the range was identified. Finally, the graphical bar heights were acquired. However, these values do not match the true scale of the bars, because the proportion of pixels used in each graph varies depending on the data presented. Therefore, the actual bar height must be computed by multiplying the pixel proportion.

I introduced the two-step method of calculating the pixel proportion shown in Figure 6.4; the steps are data preparation and Y-scale measurement. For data preparation, the leftmost partition containing both the Y-axis title and axis measurement was initially selected after partitioning the graph image. The Y-axis title is irrelevant to the pixel proportion and only the measurement part was retained. Numbers and their respective positions were recognized using OCR. The next step was Y-scale measurement. I obtained the position of each result identified by OCR and measured the difference between two neighboring recognitions, including the

difference in vertical distance. I then divided the difference between the two neighbor recognitions by the difference in vertical distance to obtain the actual number of scale units per pixel. The actual value of bars could be calculated by multiplying the height of the bars with the scale units obtained. The global tendency was analyzed from the centers of the bars by calculating the slope.

The main feature of a 2Dchart is a line or group of data points. Hence, I analyzed the graph components, the global and the local tendencies as well as the regression type. The extraction process for a 2Dchart was the same as that for a bar graph component. The titles of both axes were initially neglected to capture the data part. The image was converted to pixel values representing data points in the graph. The global tendency was identified using a global slope derived from the data points. A regression analysis was performed by using a mathematical library and identify the type of regression that was best suited to the data points using the smallest squared error. Both linear and non-linear regressions were used, including logarithmic, polynomial, quadratic, and exponential regressions.

A discontinuity in the slope may represent critical information. For example, a line graph may show the oxidation of a chemical substance against temperature and time, while a slope change indicates the saturation point. In recognition of the importance of such local tendencies, I analyzed the trend at each pair of pixel values. If a change was noted between any pair, the change in slope and the position were recorded.

6.2.2.2 Ontology construction

I constructed the classes and relations following my earlier ontology design. The graph contents, such as captions and paragraphs, were stored in a database. These graph descriptions were given in sentences produced by tokenization, as a first step in building the ontology. A dependency parser identified the sentence structures, NER, and POS tags. I endeavored to allocate each word to a category using queries in DBpedia. The queried categories were represented as the NER of tokens.

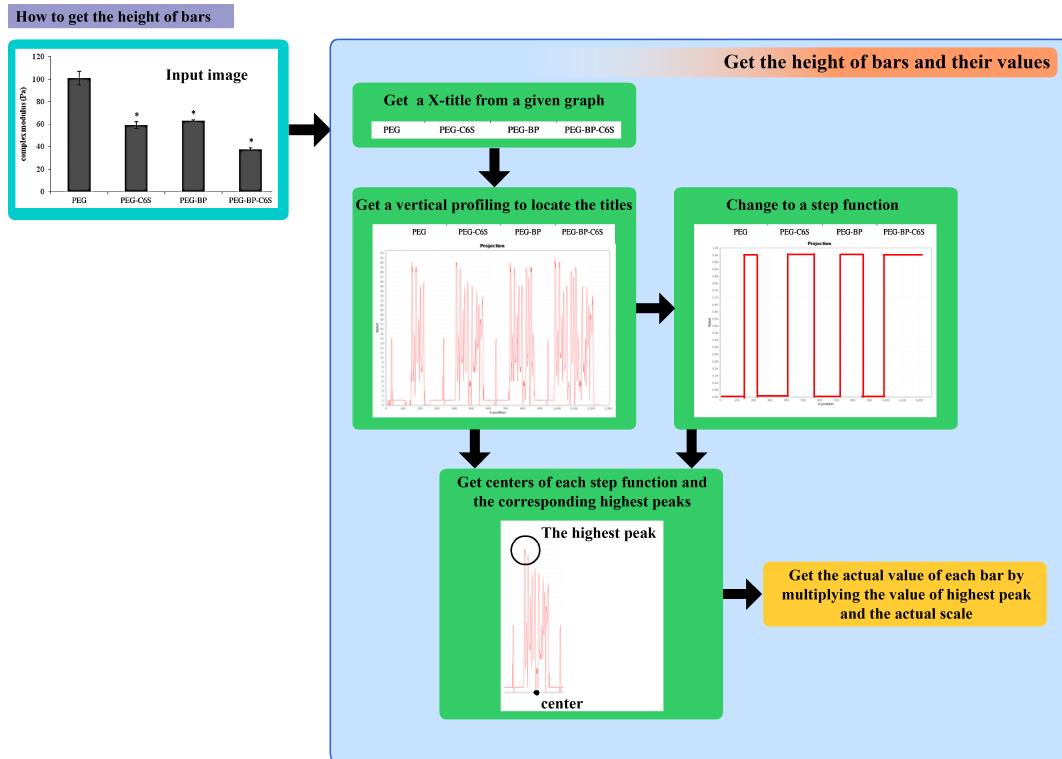


FIGURE 6.3: Bar height extraction using pixel projection and a step function

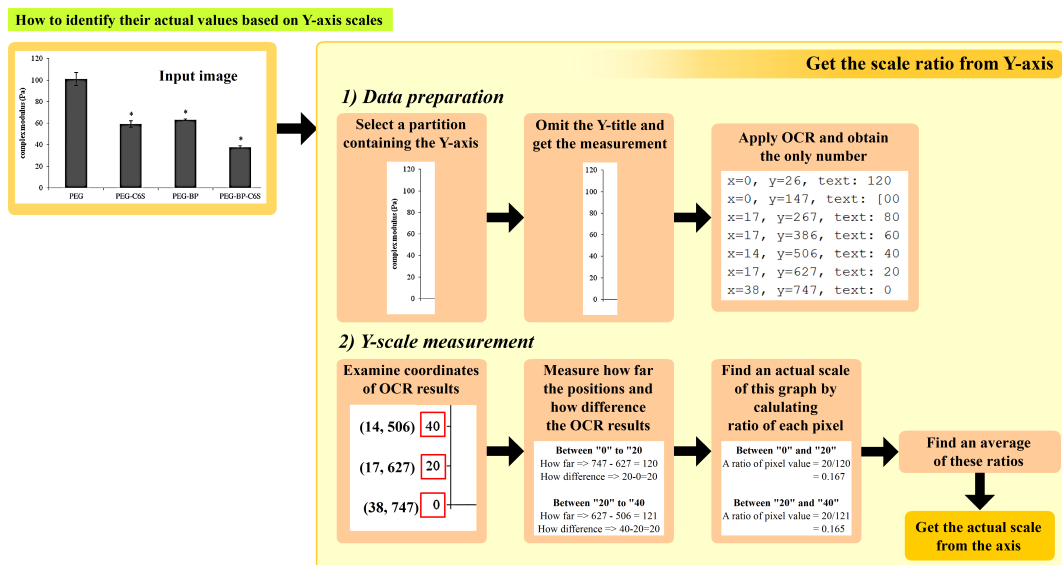


FIGURE 6.4: Pixel proportion calculation

6.3 Simulations

This section would advocate a feasibility of the method. In this study, the expected outputs were an ontology. My method provided precise information for the construction of the ontology. The data, which were stored into the ontology, came from 99 images in total that contained 35 bar graphs and 64 2Dcharts. Only the specific graph types, i.e., bar graph and 2Dchart, were applicable to the system. To validate my proposed method and ontology, the following questions were applied to the ontology:

1. What graph are both “blood” and “Hemoglobin” related to?
2. How do aphid populations impact sugar?
3. What is the tendency of the number of genes related to green fluorescent protein (EGFP) expression?
4. What value of Lipopolysaccharide (LPS) is described in all graphs and what is its relation?
5. What is a relation between Hemoglobin and Hemoglobin A1c (HbA1c)?

Note that all my input graphs were in the field of biology, as the data were collected from journals available through PubMed. SPARQL queries were built to retrieve the related classes and relations of the ontology. The simulation was meant to model a user attempting to use my ontology and deciding what kind of question should be asked.

The SPARQL queries and their results are presented in Figures figs. 6.5 to 6.9. Figure 6.5 shows the query command and the results obtained for the first question. Three graphs presented by Nekooeian et al. [70] and Sinha et al. [79] mentioned blood or Hemoglobin in their captions. Figure 6.6 shows a graph by Cao et al. [18], with the values of each bar representing the impact of aphid populations on sugar. Figure 6.7 presents answers to the third question from a graph relating the number of genes and the EGFP expression by Gao et al. [31]. Figure 6.8 shows

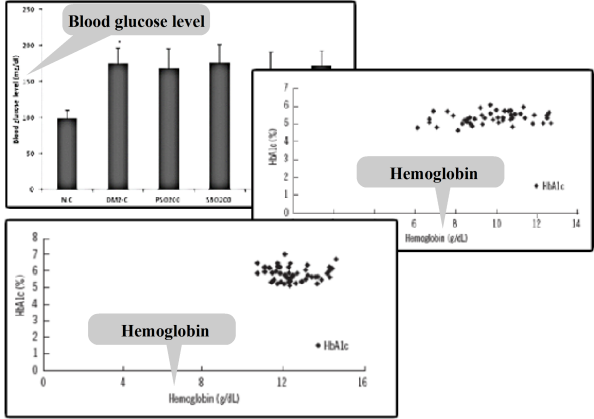
Query	Results
<pre> SELECT ?graph WHERE { { base:blood rdf:type graphContentOnto:Token ; graphContentOnto:tokenize_caption ?caption. ?graph rdf:type graphContentOnto:Graph ; graphContentOnto:describe ?caption. }UNION{ base:hemoglobin rdf:type graphContentOnto:Token ; graphContentOnto:tokenize_caption ?caption. ?graph rdf:type graphContentOnto:Graph ; graphContentOnto:describe ?caption. }UNION{ base:Hemoglobin rdf:type graphContentOnto:Token ; graphContentOnto:tokenize_caption ?caption. ?graph rdf:type graphContentOnto:Graph ; graphContentOnto:describe ?caption. } } </pre>	<p>1 base:ijms-39-130.PMC3957012_figure1_1_bar 2 base:alm-32-17.PMC3255499_figure2_1_2Dchart 3 base:base:alm-32-17.PMC3255499_figure3_1_2Dchart</p> 

FIGURE 6.5: SPARQL query command and answers for Question 1

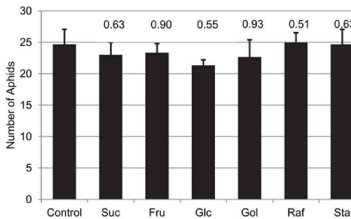
Query	Results
<pre> SELECT (SAMPLE(?cate) AS ?category) (SAMPLE(?number) AS ?value) WHERE { ?graph graphContentOnto:contain ?data_part ; graphContentOnto:describe ?caption ; graphContentOnto:is_referred ?para . ?data_part graphContentOnto:is_high ?bar_height . ?bar_height graphContentOnto:value ?number ; graphContentOnto:result_of ?xtitle . ?xtitle graphContentOnto:is_x ?cate . base:aphid rdf:type graphContentOnto:Token ; graphContentOnto:amod base:populations . {base:aphid graphContentOnto:tokenize_caption ?caption .} UNION {base:aphid graphContentOnto:tokenize_para ?para .} }GROUP BY ?cate ?number </pre>	<p>1 "Gut" 28.2 2 "Glc" 28.4 3 "Control" 27.733333333333334 4 "Fru" 27.133333333333333 5 "Suc" 26.866666666666667 6 "Raf" 28.2 7 "Suc" 26.466666666666665</p> 

FIGURE 6.6: SPARQL query command and answers for Question 2

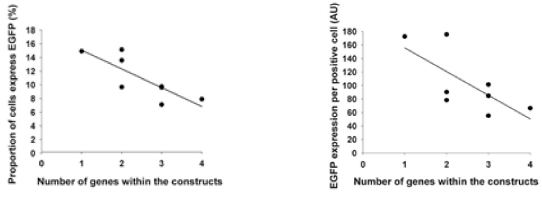
Query	Results
<pre> SELECT ?graph ?slope WHERE { base:EGFP graphContentOnto:tokenize_caption ?caption . base:genes graphContentOnto:tokenize_caption ?caption ; graphContentOnto:nmod base:number . ?graph graphContentOnto:describe ?caption ; graphContentOnto:contain ?data_part . ?data_part graphContentOnto:global ?slope. } </pre>	<p>1 base:pone.0048668.PMC3490874b_figure3_b_2Dchart base:decrease 2 base:pone.0048668.PMC3490874c_figure3_c_2Dchart base:decrease</p> 

FIGURE 6.7: SPARQL query command and answers for Question 3

Query	Results
<pre>SELECT ?graph ?xtitle ?number WHERE { base:LPS graphContentOnto:tokenize_component ?component . ?graph graphContentOnto:has ?component. ?component graphContentOnto:is_x ?xtitle. ?bar_heightgraphContentOnto:result_of ?component. ?bar_height graphContentOnto:value ?number . FILTER regex(?xtitle, "^LPS\$", "i") }</pre>	<p>1 base:1476-9255-11-16.PMC4046032_figure4_5_bar "LPS" 79.5 2 base:1476-9255-11-16.PMC4046032_figure5_4_bar "LPS" 68.5</p>

FIGURE 6.8: SPARQL query command and answers for Question 4

Query	Results
<pre>SELECT (SAMPLE(?token) AS ?Subject) (SAMPLE(?token2) AS ?Predicate) (SAMPLE(?token3) AS ?Object) WHERE { base:hemoglobin graphContentOnto:tokenize_caption ?caption . base:HbA1c graphContentOnto:tokenize_caption ?caption . base:A1c graphContentOnto:tokenize_caption ?caption . ?token graphContentOnto:tokenize_caption ?caption. {?token ?rel ?token2 ; graphContentOnto:tag base:noun; ?a base:hemoglobin. }UNION{ ?token ?rel ?token2 ; graphContentOnto:tag base:noun; ?a base:HbA1c. }UNION{ ?token ?rel ?token2 ; graphContentOnto:tag base:noun; ?a base:A1c. } ?token2 ?rel2 ?token3 . ?token3 graphContentOnto:tag base:noun. ?rel rdfs:subPropertyOf graphContentOnto:depend_on. ?rel2 rdfs:subPropertyOf graphContentOnto:depend_on. }GROUP BY ?token ?token2 ?token3</pre>	<p>1 base:hemoglobin base:levels base:plasma 2 base:HbA1c base:levels base:correlation 3 base:hemoglobin base:levels base:correlation 4 base:HbA1c base:levels base:plasma 5 base:HbA1c base:hemoglobin base:A1c 6 base:HbA1c base:hemoglobin base:levels 7 base:HbA1c base:levels base:Correlation 8 base:hemoglobin base:levels base:Correlation 9 base:HbA1c base:A1c base:Abbreviation 10 base:hemoglobin base:A1c base:Abbreviation 11 base:hemoglobin base:levels base:Quantification 12 base:HbA1c base:levels base:Quantification</p> <p>The caption is "Correlation between hemoglobin and HbA1c levels in patients after 1 month of treatment. Abbreviation: HbA1c, hemoglobin A1c."</p>

FIGURE 6.9: SPARQL query command and answers for Question 5

how my SPARQL query interrogated the ontology to retrieve graphs by Kim et al. [55] pertaining to LPS and includes its values. Figure 6.9 shows the results for the correlation between Hemoglobin and HbA1c in a graph presented by Sinha et al. [79]. This displayed all tokens that had at least one relation with the specified tokens. For quantitative evaluation, I analyzed the precision and observed errors that arose in the course of the simulation. For the aforementioned five questions, I obtained relevant answers by using five queries. However, errors arose due to OCR misrecognition. These were ignored because they were not related to the validity of

the ontology.

6.4 Discussion

In this study, I presented a new method of extracting information from a graph based on the use of ontology. I extracted the graph components and data located in the data section of the graph. A dependency parser was applied to analyze the captions of the graph and related paragraphs. The category to which each token belonged was acquired from DBpedia. The method was then applied to a graph-based search engine with user queries in the field of biology. The goal was to use the ontology to extract both implicit and explicit information from the graphs. Five inquiries were run, and the answers returned were, in the main, correct. Unfortunately, in some cases (e.g., the second question), failures in OCR introduced errors. The accuracy of the results provided evidence that my proposed method was able to precisely extract information from the graphs. For the fifth question, answers were found from the captions of the retrieved graphs and several triples representing tokens that were connected by dependencies were obtained. Interestingly, I retrieved tokens that were not available from the captions of the graphs, but were instead taken from other graphs sharing the same concepts such as quantification and plasma. Based on this result, my ontology is suitable for use in inquiries involving information pertaining to a graph. However, a limitation of the study was that I focused only on a limited set of graphs: line, plot, and bar. my system does not yet support analysis of other graph types that require a different method of interpretation. Moreover, the system currently cannot deal with multiple data.

Here, I will discuss possibilities of this study. According to errors from bar height measurement, I realized that a main cause of the error was OCR misrecognition. To address this problem, I should use my proposed OCR-error correction to suggest corrected words to the errors. Remind that if it will be used to solve this problem, it should be modified to work with numbers instead. Obtained knowledge will be expanded if I map an interpretation of quantitative data extracted from the data section of the graph to ontology. For example, a graph is interpreted its trend

that is also described in a context of the description. Moreover, this information is found on other graphs. Based on this example, I will acquire extended information according to shared concepts. Furthermore, if I merge my ontology with other ontologies, I possibly obtain unexpected knowledge come from different study domains. For example, I integrate medicine ontology with biology ontology. If I inquiry about some protein names, I may obtain knowledge of medicine relating to the protein names. Moreover, my system could identify a regression type of data; thus, I possibly predict unseen data by applying statistic analysis, such as linear or non-linear regression.

6.5 Conclusions

I developed an effective method for extracting graph information, and an ontology to support the dependency parsing of English sentences. Several techniques were combined to achieve this: OCR, NLP, and ontology. I evaluated the method by using the constructed ontology to address five questions. Accurate answers were obtained and significant results were achieved by the shared concepts used in my ontology, thereby demonstrating the effectiveness of the method. In future studies, I will develop the system further by building a simple user interface and extending the dataset to allow quantitative evaluations. I may also extend the domain of search data to other fields such as engineering.

Chapter 7

A Prototype of Ontology-Based Search Engine System

In the previous chapter, I proposed a method of extracting significant information from graph's data section, including creating an ontology for my search system. Under my consideration, only the ontology established is inapplicable for real use because users cannot understand and utilize the ontology without an amicable application directly. To achieve this obstacle, a prototype of ontology-based search engine system has been introduced in this chapter.

The main focuses of this chapter are to utilize the ontology containing information extracted from graph images and present them on a web-based application with comprehensible user interfaces. My system differs from a traditional search engine system because it can be queried by some questions, while the traditional one cannot. I attempt to list several questions that users need to know from the graphs, such as the tendency of data and a comparison between particular data categories along with their data labels. Due to this avail, the users obtain extended information from my system outperforming the tradition one which is applicable only keyword-based inquiry.

Background of the study will be presented at the first section of this chapter. The next section is the methodology which describes overall system design and implementation. Regard experiment and evaluation parts, I will explain in the next chapter.

7.1 Background

An ontology specifies the representation of a conceptualization. Recently, ontologies have been recognized as an important feature of information retrieval systems owing to their ability to link knowledge in different areas of the ontology. Ontology-based search engine systems can acquire more useful information than traditional search engine systems because users can find not only particular concepts obtained by a given query but also other related concepts. However, for the practical usage of ontologies, an application must be developed to present the results from an ontology query because the realization of ontologies alone is extremely difficult for the average end user. Typically, to input a query via ontology, it is necessary for the user to be skilled in a query language; in addition, a specific ontology realization will be required. This creates problems for a general user who would then avoid using the complex system. Therefore, via constructing a web-based application with user interfaces, I introduce a handy and capable search system that does not require any computer skills.

In recent years, there has been a substantial amount of research on information retrieval. There are many types of data that are regular targets for search systems, including text [85] and image [13] data. In particular, image-based information retrieval is a growing topic in several study fields (e.g., computer vision and knowledge-based information retrieval) because methods for extracting data from images are more complicated than those for extracting data from text. Hence, researchers require particular methods for extracting information from an image. For example, a system of image-content extraction could analyze an image's low-level features [64] such as colors.

In the past several decades, there have been several studies that proposed semantic search engine systems. Li et al. [62] developed a fuzzy search by allowing mismatches between query keywords and answers. To freely explore data, this fuzzy search accepted keywords even in the presence of minor errors. Based on this existing study, I realized that results or knowledge provided by a conventional system are limited because they are solely dependent on given keywords. In the most recent decades, studies on information retrieval have advanced to semantic systems utilizing ontology concepts that enhance and extend the obtained knowledge based on user specifications. Jayalakshmi et al. [44] proposed a semantic search engine system that depended on inverse document frequency (IDF) and text mining. The proposed search engine system created the search indexing using the contents of the files to retrieve the relevant document from a computer. Another existing study constructed a scalable semantic search for geospatial data [14] in which an application layer and a search service that provided a specific search functionality inspired by resource description frameworks (RDF) was introduced.

The objectives of this proposed system are (1) to propose a new method that can provide extended results outperforming the traditional system; (2) to support user convenience by presenting inquiry results with understandable user interfaces; and (3) to evaluate the validity of the system by systems performance comparison.

The data collection used in this system is graph images, as I used in my previous systems. Moreover, I obtain cleaned data extracted from graph components, including essential information from graphs' data sections.

7.2 Methodology

A methodology presented in this chapter utilizes the methods proposed in the previous chapters, i.e., OCR-error correction, graph component extraction, and graph information extraction. The entire systems are integrated into one main system. Note that the size of graph image dataset is 636 images.

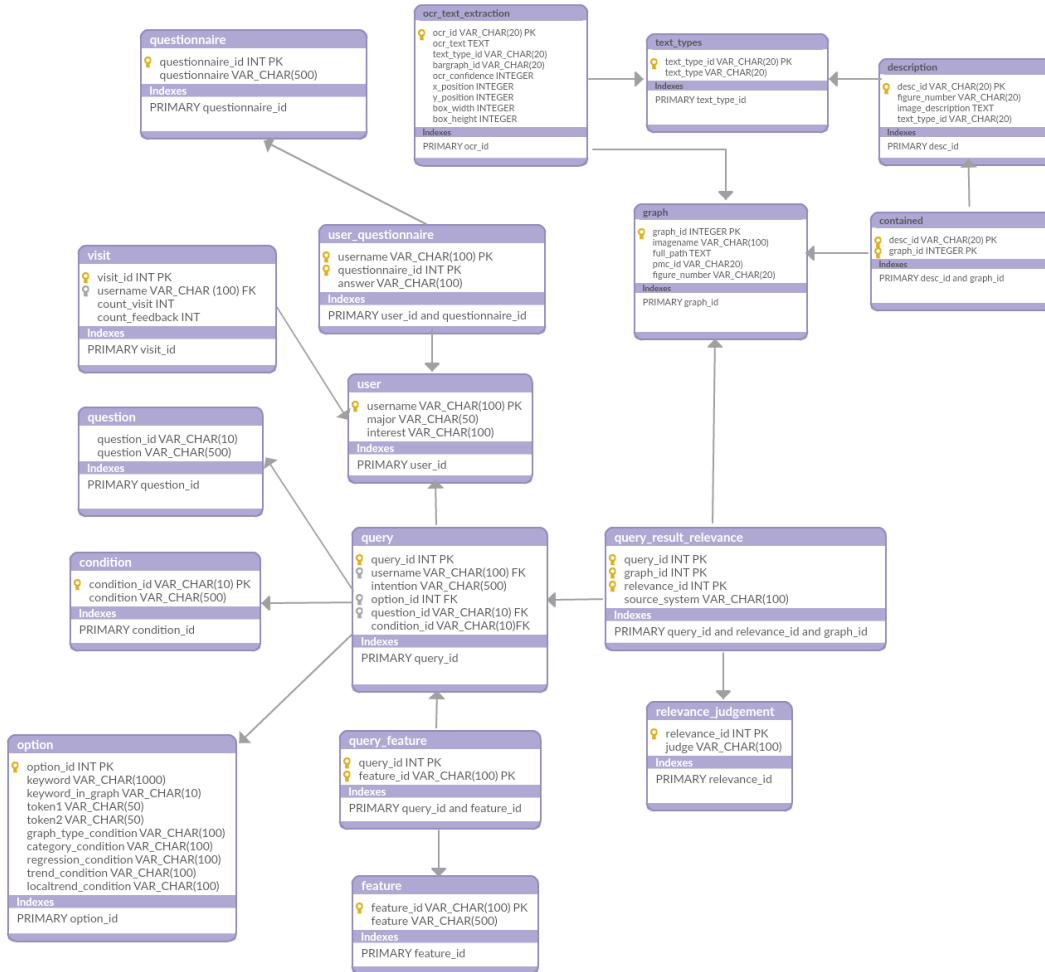


FIGURE 7.1: Illustration of relational database used in this system

7.2.1 Database design

For this main system, I used a relational database to store data that was related to my target data (i.e., a collection of graph images, including their captions and text paragraphs) and other necessary information such as captions, text paragraphs, and graph profiles. It was constructed due to two major purposes: to store the graph information (e.g., captions, paragraphs, and graph profiles) and to record user evaluation feedbacks. The graph information was an important data used to create my ontology. Moreover, due to an evaluation purpose, the system allowed

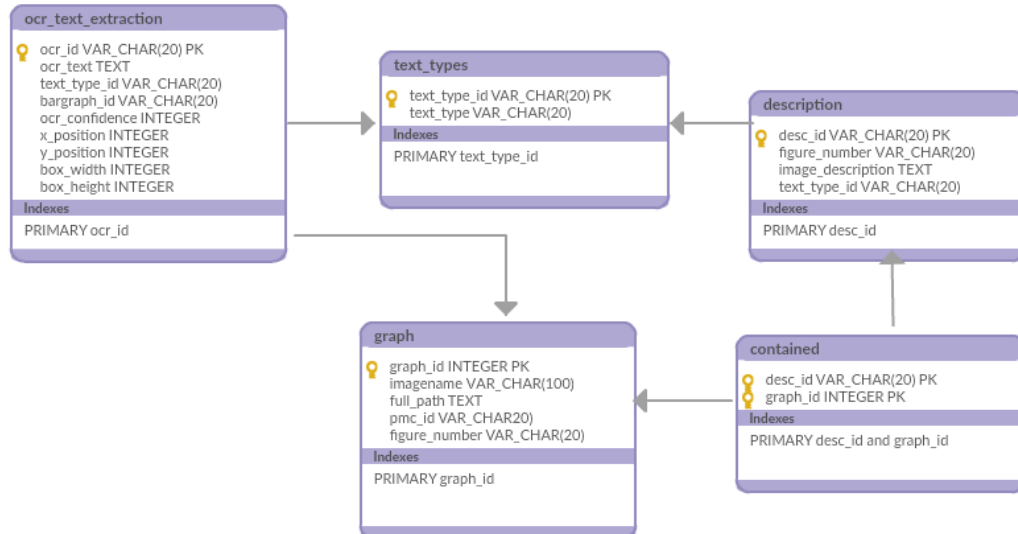


FIGURE 7.2: A part of database storing generic data related to the graph images

the users to comment and validate results obtained from queries accessing to both search engine systems. Those feedbacks were recorded for statistical analysis. The database design is presented in Figure 7.1.

Five tables were used to record the following information from each graph: graph, contained, description, text_types, and ocr_text_extraction (Figure 7.2). The “graph” table collected the profiles of graph images, such as the graph name. The “description” table contained the graph’s captions and the paragraphs that referenced the graphs. The “text_types” table contained the different types of the graph’s descriptions (e.g., caption, X-title, and legend). To acquire the graph components, I used OCR to first recognize and convert them into digital data. These data were stored in the “ocr_text_extraction” table.

Regarding the rest of tables, they were utilized to record user feedbacks (Figure 7.3). I collected not only the user evaluation feedbacks but also the results from queries. User table collected the user information, such as name and major. Question, Condition, Feature tables primarily kept the inquiry details, for example, questions used for a query. Query and Option tables stored such information of user’s query on each iteration. After the users inquired queries to the system,

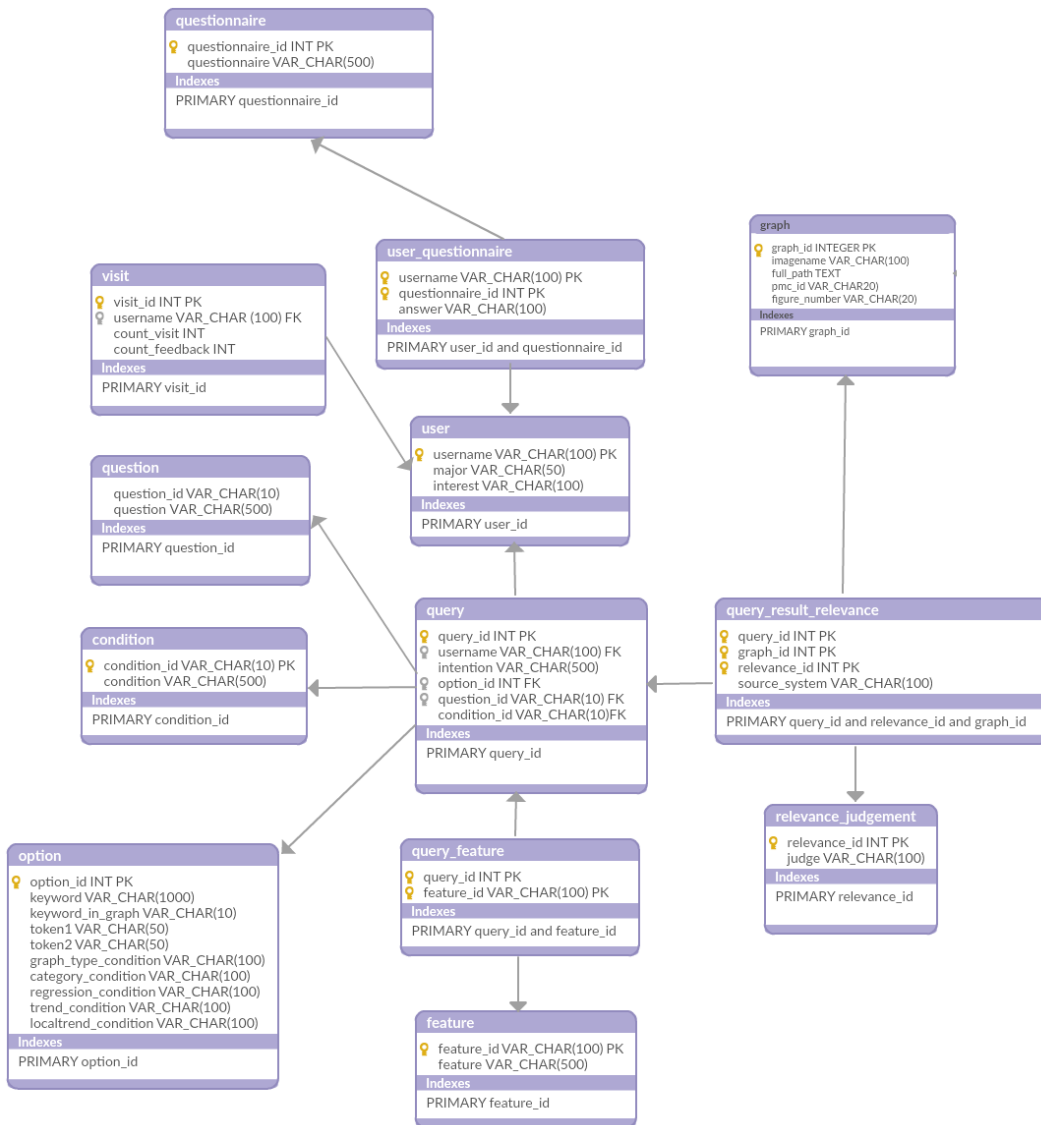


FIGURE 7.3: A part of database storing feedback data for the search engine system (a final system) in Feedback mode

some relevant results should be returned to the search engine systems, and those were evaluated by the users. Then, the obtained results and user evaluation were collected into Query_result_relevance table.

7.2.2 Ontology design

The ontology used in this study was based on the structure design in Chapter 6, but I redesigned it to be more applicable and to meet the requirements of the search engine system analyzed in this study. (Figure 7.4).

Herein, I describe the updated parts that differ from the previous version of the ontology. Note that the previous ontology was simply applicable to singular data in plot, line, and bar graphs but could not handle multiple data. Thus, I have since added a few relations that allow the ontology to be applicable to bar graphs containing multiple data labeled in a legend. A relation named “show” now connects the Legend and Slope classes because it describes the data tendency of each data label that belongs to different categories in the X-title class. A “represent” relation indicates the height of each data.

7.2.3 System implementation

An ontology-based search engine is a search engine application that utilizes ontologies to fulfill user inquiries. Through the use of ontologies, it is possible to obtain relevant results to a query as well as to obtain new extended knowledge. Such a search engine system helps users to retrieve images of graphs that contain information they require, such as a relation between the X-and Y-axis labels and a comparison of each bar in a bar graph. The users can easily comprehend the graph and its descriptive details because the search engine system precisely provides detailed information such as main ideas and tendencies. The system allows the users to select specific questions for inquiring; moreover, some settings must be accepted to restrict the amount of obtained results.

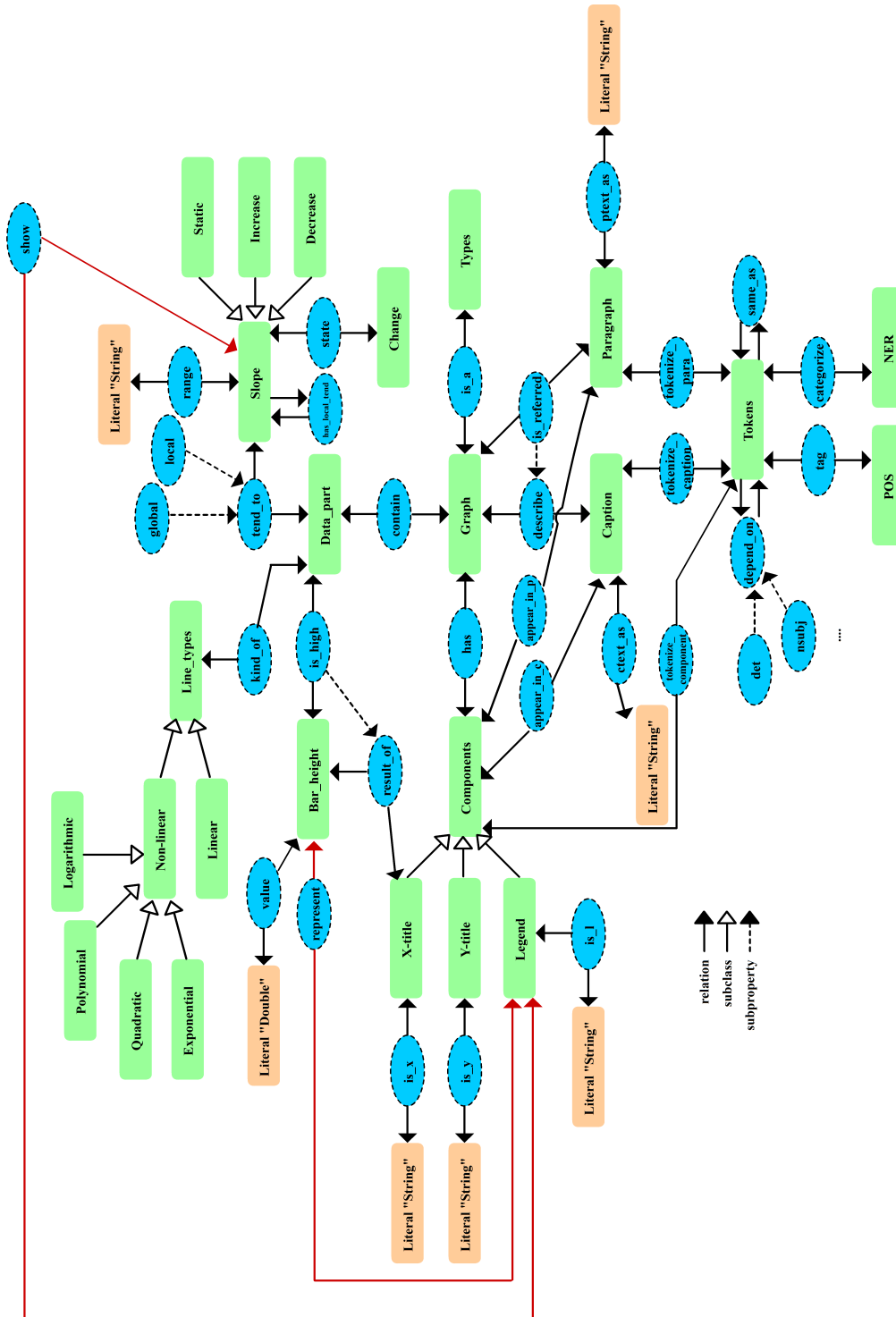


FIGURE 7.4: Illustration of the updated ontology, i.e., observing at read arrows, used in this system

I implemented the system containing two different modes: search mode and feedback mode. The search mode was used to search and inquire to the system by inserting some keywords and specific questions; then, relevant results were returned that were displayed in a search page. Whilst, the feedback mode contained a particular feature used for acquiring the user's feedbacks. This mode was proposed for an evaluation purpose. After the results were retrieved, the user would analyze and decide them as either relevance or irrelevance based on their intention. They should evaluate every result presenting on both my system and a traditional search engine which is called Elasticsearch (ES). In addition, in the feedback mode, there were two more web pages that required user profiles (i.e., a user page) and evaluation opinions (i.e., a questionnaire page). However, I will explain in the next chapter at Section 8.3.

I implemented the described search engine system in a search application. The developed application can query the search engine system by specifying some keywords and specific questions. The relevant results are then returned and displayed in a search page. This system was designed to support simplicity and immediate availability. To that end, only necessary functions such as the query settings are shown on the web page. Three sections such as menu section, inquiry section, and results section are presented on the main search page. There were three sections presenting on the page as shown in Figure 7.5: menu section, inquiry section, and result section.

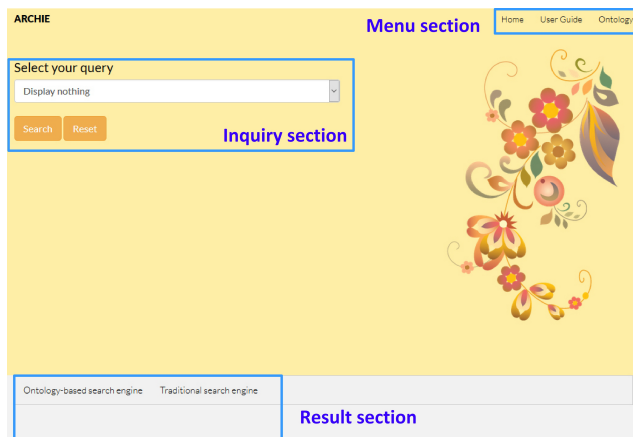


FIGURE 7.5: A user interface of search page with three sections

The menu section contains three tabs, namely home, user guide, and ontology. The home tab is the default screen when the system is launched in the search mode. The user guide is a page that briefly explains the system and its components, including a guideline of the system process and examples of system simulations. The ontology tab displays the ontology schema used in this system.

In the inquiry section, the users can select questions and input some required settings. The acquired relevant results are displayed on the results section. In addition, the question option can be selected by the users. I offer a few options that can help the users to filter unnecessary results, (i.e., conditions, as shown in (Fig. 7.6,) and features, as shown in (Fig. 7.7).

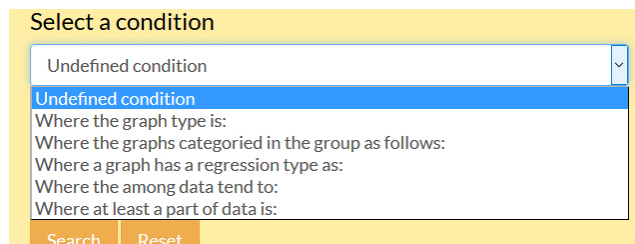


FIGURE 7.6: Selectable conditions for results filtering

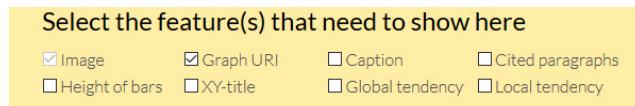


FIGURE 7.7: Selectable features that can be presented at the result section

The condition box contains five conditions. First, the users can restrict the graph type. I distinguish the graph types into two types such as bar graphs and 2Dchart. Second, the users can select results that belong to a specific group. Third, the results shown in the results section can be filtered based on a specific regression type. (For example, a user might need only graphs that have linear regression.) Fourth, the users can select the results with a specific tendency such as increasing or decreasing. Finally, for the line and plot graphs, a local tendency is also a significant option, because changes in the graph might identify essential information. Thus, users can filter the results based on the data variation. The feature box was created

to cover the needs of all users because additional information such as the graph caption or X- and Y-labels might be required by a user.

In the results section, results from my ontology-based and ES-based search engine systems are presented, a user can independently choose a tab to examine the results depended on the individual systems.

Herein, I discuss the questions that are included in the system and the settings that must be entered by the user. There are six queried questions, described as follows:

- Question 1: Display the graphs involving this following keywords.
The first question is the most basic because it is similar to a keyword-based search engine system (e.g., Google search). There are a few settings that are required and must be completed by the user. For example, a user may need graphs that feature a specific inputted token in the graph for deep discussion on a particular topic. An example query form for Question 1 is illustrated in Fig. 7.8. The user simply inputs at least one keyword to the text box separated by commas (for example, the string “data, test, accuracy” can be inputted to the text box). Moreover, the user can specify whether the keyword(s) must appear in the graph’s components (e.g., X-label, Y-label, or legend) by choosing either the “yes” or the “no” radio button above the text box. There is an optional text box that asks the user’s intention for the query; however, to complete the evaluation, the user should describe their intention for their query. For example, a user may input keywords such as “neural network, accuracy, image,” and the intention would be to obtain graph images relating the accuracy of neural networks when dealing with images. Figure 7.9 presents an example of result launched by Question 1 with additional settings, i.e., caption and global trend features.
- Question 2: Display the graphs involving following keywords and their main idea of captions.
The second question (Figure 7.10) requires only keyword(s) from the users to produce the relevant results. Moreover, the question asks for the main

Select your query

Display the graphs involving this following keywords:

Must the keyword appear in graphs? No Yes

Ex. A,B,C

Describe your query intention here.:

Select a condition

Undefined condition

Select the feature(s) that need to show here

Image
 Graph URI
 Caption
 Cited paragraphs
 Height of bars
 XY-title
 Global tendency
 Local tendency

FIGURE 7.8: Question 1 and its settings

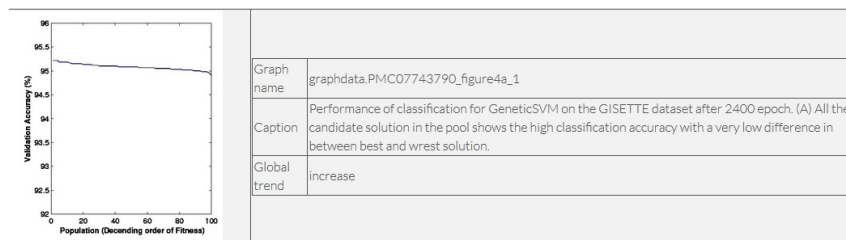


FIGURE 7.9: Example of result performed by Question 1

idea of the graph descriptions (e.g., the caption). Therefore, an extra feature has been added to the results section (Fig. 7.11), that presents the main idea. Sentences containing the main idea are selected by analyzing the appearance of keywords and the first sentence of the paragraph. A user can use this question to summarize information to realize the underlying concept of a graph.

- Question 3: Display the graphs involving following keywords and their maximum and minimum values of graphs.

The third question (Figure 7.12) is similar to the second question, and it requires keyword(s) to be set. The bar height and local trend features are initially selected and displayed in the results section, including the highest and

FIGURE 7.10: Question 2 and its settings

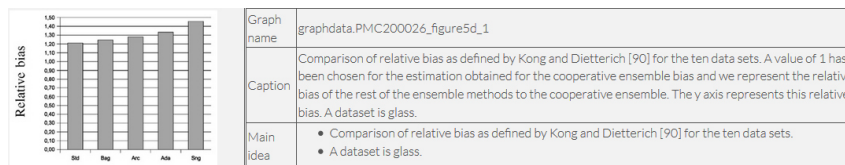


FIGURE 7.11: Example of result performed by Question 2

lowest values identified (Fig. 7.13). However, for a bar graph containing multiple data, it is difficult to identify which the highest and lowest values; thus, a comparison between each bar height and the legend are displayed instead. A user may use this question to analyze statistical data to compare to their results.

- Question 4: Display the graphs relationship extracted from axis titles. In general, there are significant relations that are established in any given graph. The fourth question is used to indicate which tokens are a part of a graph's relationships. For this question, the user inputs keyword(s), just as in the previous questions, and the relevant results are presented, including some tokens related to the relation between the X- and Y-labels (Figure 7.15). Then, the users must interpret the graph relations and expressions.
- Question 5: Display the relationships between two different tokens. The graphs used in this system were collected from a number of publications,

Select your query

Display the graphs involving following keywords and their maximum an

Fill up the keywords:

Describe your query intention here:

Select a condition

Undefined condition

Select the feature(s) that need to show here

Image Graph URI Caption Cited paragraphs
 Height of bars XY-title Global tendency Local tendency

FIGURE 7.12: Question 3 and its settings

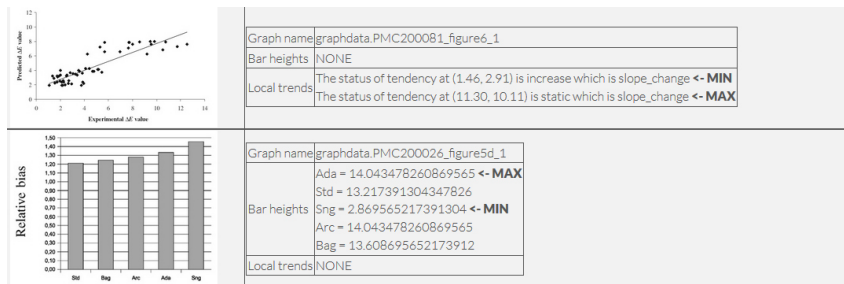


FIGURE 7.13: Example of result performed by Question 3

Select your query

Display the graphs relationship extracted from axis titles

Fill up the keywords:

Describe your query intention here:

Select a condition

Undefined condition

Select the feature(s) that need to show here

Image Graph URI Caption Cited paragraphs
 Height of bars XY-title Global tendency Local tendency

FIGURE 7.14: Question 4 and its settings

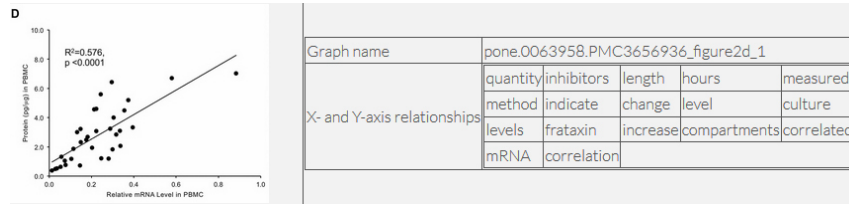


FIGURE 7.15: Example of result performed by Question 4

and they are always described by captions and cited paragraphs. Sentences comprise several tokens that are dependent on one another. The fifth question is similar to Question 4, but the question investigates the relationships between two different keywords. A user may use this question to understand any implicit relations between two tokens hidden in the descriptions. Figure 7.17 displays the results obtained by performing Question 5.

Select your query

Display the relationships between two differnt tokens ▼

The first keyword:

The second keyword:

* Please input one token to each textbox

Describe your query intention here:

Select a condition

Undefined condition ▼

Select the feature(s) that need to show here

Image
 Graph URI
 Caption
 Cited paragraphs
 Height of bars
 XY-title
 Global tendency
 Local tendency

FIGURE 7.16: Question 5 and its settings

- Question 6: Display the comparison of bar values on different X-categories but same data label.

The sixth question presents information in bar graphs that feature multiple data labels. The question presents a comparison based on bar heights and

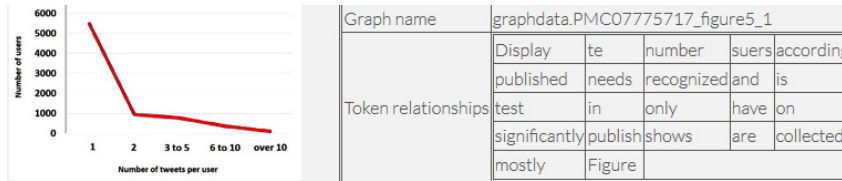


FIGURE 7.17: Example of result performed by Question 5

FIGURE 7.18: Question 6 and its settings

legends in the bar graphs. The comparisons can be achieved with respect to one of two items: with respect to bar categories (e.g., X-label) or with respect to the legend (or data label). A user may use this question for data comparison and analysis. Fig. 7.18 and 7.19 shows a form of Question 6 and an example of results generated using Question 6.

7.3 Conclusions

This system is the final system introduced in this dissertation. I implemented a prototype of ontology-based search engine system. I assembled all proposed systems in order to obtain knowledge extracted from the graphs based on searching, which was recorded into my new ontology. The ontology and database schemes were designed for the system. The main goal was to address the problem of the

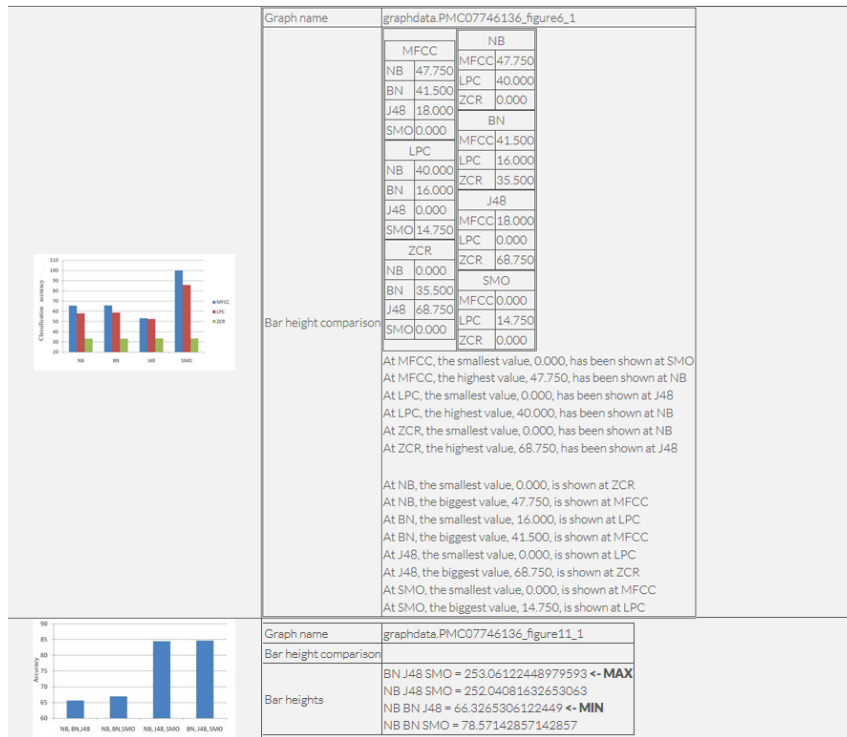


FIGURE 7.19: Example of result performed by Question 6

semantic gap by using ontology. For general use, my search engine system had a page for querying by using keywords. For an evaluation purpose, I added more two web pages assigned for the feedback mode, i.e., user and questionnaire pages, to collect necessary information. Ten participants had been required to do several tests through the system. They used some keywords and evaluated the obtained results by determining them as either relevance or irrelevance. With regard to experiments of this system, I will present it in the next chapter.

Chapter 8

Experiment and Evaluation

In the previous chapter, I presented a methodology of ontology-based graph search engine system literately. I designed and constructed my own ontology that supported the dependency parsing from English sentences and graph information. As mentioned in the prior chapter, I focused on extracting explicit and implicit information (e.g., axis titles and the relationship between titles) from graph images collected from the scientific literature. This method confidently provided researchers new solutions to access essential knowledge hidden behind the images, including extending information utilizing ontology. Moreover, a web application with friendly user interface had been implemented that responded to a purpose of user convenience. Users could use the system to query questions depended on their intention with specific keywords.

To measure system performance, I asked ten participants who experience to computer and biology to use the system and evaluate the relevancy of obtained results. Moreover, they required responding a few questions in order to cover the experiment purpose, such as how users satisfy the system. In this chapter, the experiments and results will be shown; moreover, a detail about experiment procedure will be included. The experiments show a robustness of the evaluation.

Initially, I will mention about system evaluation background describing several existing traditional search engine systems and reasons to select one of them used

for comparing to my ontology-based search engine system. Next, I will present experiment configuration describing software and settings used to accomplish the experiments. Furthermore, experiment procedure will be described that presents about the feedback mode of my system. It briefly described total steps of this dissertation's experiments. Then, user feedback evaluation, where is a core of this chapter, will be introduced. It presents about a target group of participants and their actions corresponding to the experiments. Finally, I will show the experimental results and the system performance measured by calculating performance models based on responsive feedbacks.

8.1 System Evaluation Background

This dissertation aims to create the ontology-based graph search engine system, as described in the previous chapter. Regarding system validation, I considered what the best way to evaluate the system, and I acknowledged that it would be great to have another common search engine to compare with. Therefore, to respond this idea, I decided to examine other search engine software commonly used nowadays. It should be noted that the classic search engines used for comparison must support full-text search and indexability.

After I investigated several search system software, it seemed to be difficult to pick the best one, if I did not know their functionalities and indexing processes. I determined to select five software containing different functionalities, i.e., ES, Solr, Sphinx, DB2 text search extender, and a state-of-art work (see Appendix B).

Among the classic search engine candidates, ES was the winner that had been used for comparing with my ontology-based graph search engine. ES, which is a full-text search engine, provides retrieved results based on keywords. In experiments, the user independently inputted keywords, selected questions, and some settings to the evaluated systems. Retrieved results from both systems were sometimes different because the users could obtain extended information from my system. For example, users would like to know about a number of publications published in this year compared to last year. They inquire my ontology-based search system by selecting

a question asking about tendency and input some relevant keywords, such as paper and number. In this case, it was possible to obtain results from my system. On the other hand, there was a limitation on ES. It could not deal with other specifications besides the inputted keywords. Therefore, this was a huge difference between my system and the traditional one.

8.2 Experiment configuration

The model of configuration for the experiments was described in this section.

The database software used for the system was PostgreSQL, which is sophisticated open-source database management system (DBMS). I used it because of its simplicity and supporting almost all SQL constructs, such as sub-queries. This database used to record the experiment results and the user evaluation.

The web application was implemented by PHP and javascript. It connected to the database to fetch graph information for the traditional system and store the results of experiments. In addition, for my search system, the web application must use the ontology that can be launched by Apache Fuseki, which is a SPARQL server providing provides REST-style SPARQL HTTP Update, SPARQL Query, and SPARQL Update using the SPARQL protocol over HTTP.

The users should specify keywords that were in the domains of biology and computer engineering. The data collected from publications were published in those areas; therefore, to obtain available results from the systems, the keywords related to those study areas unavoidably.

These were possible keywords in biology domain. Most images gathered from publications relating protein, DNA, and diseases. The examples are displayed below.

- comparison, miRNA, prediction, protein
- correlation, PDB chain
- amino, alanine

- HbA1c, patients, correlation
- cancer

Keywords in a computer domain related to data mining and machine learning domains, particularly about algorithms and their performance such as ANNs and SVMs. The examples are displayed below.

- intrusions, classifiers, performance
- image, classification
- sensitivity, specificity
- SVMs, accuracy
- Neural Networks, performance

8.3 Experiment procedures

As described in Chapter 7, my system contained two different modes: search mode and feedback mode. The feedback mode will be described in this chapter. In the feedback mode, some additional processes had been required to be inputted by the users. They, who participated the system evaluation, should follow a flow of the feedback mode as shown in Figure 8.1.

In the first step, the users should access to a user page (Figure 8.2) to fill own personal information, such as a name, a major, and interest(s). The name represents a display name or username, which does not necessitate to be a real name. The users must select a major that they belong to. The interests represent topics that they experience with, such as data mining, computer vision, and programming.

Next, after a submit button was pressed, the process moved to the search page. The users should test the search engine system on the search page. Due to validate the system, I required them to repeatedly test the system totally three times in a

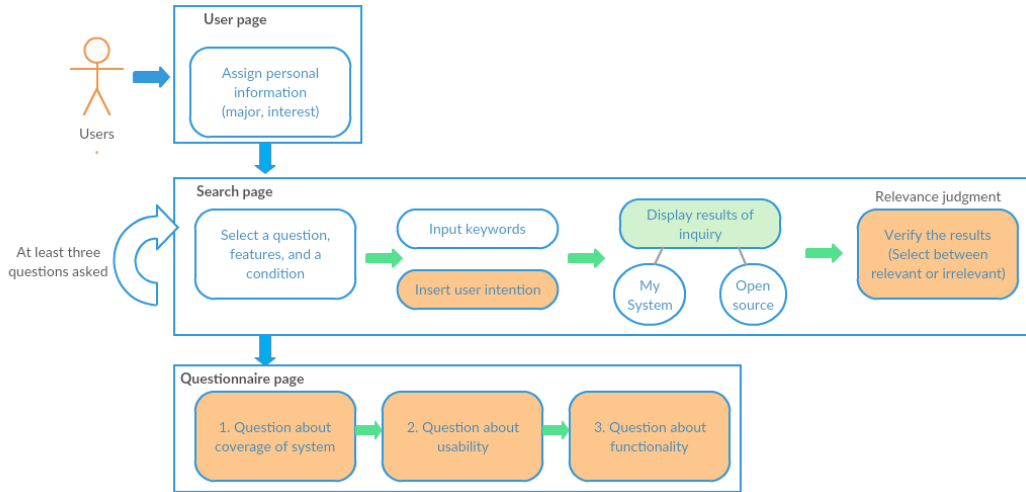


FIGURE 8.1: System flow in feedback mode

FIGURE 8.2: User page

row by using different keywords, questions, and other settings because I needed to collect a number of user evaluations for reliable results analysis. Remind that the users should evaluate the obtained results by selecting a decision between relevance or irrelevance as presented in Figure 8.3. Optionally, I asked the users to input an intention to describe a reason why they used the keywords and what were their intention. This information should be used for the system’s discussion. Moreover, on this page, not only my search system was proposed, but the traditional search engine system was also presented here. Therefore the users must validate the results acquired by both systems. The user responses were collected in the database as well as were used to analyze and compare performance between them.

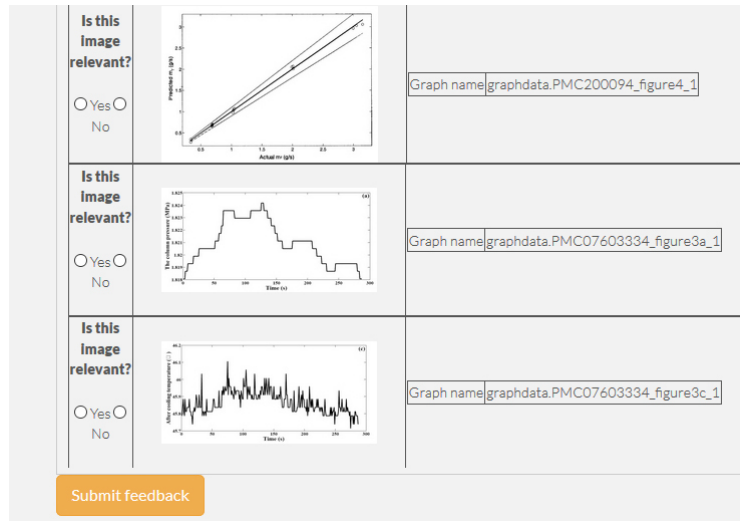


FIGURE 8.3: Illustrating the results of query in feedback mode

After the users completed their evaluation in each iteration, the users must submit a button locating at the bottom of the page in order to send the feedbacks to store into the database. Note that, to count as one time queried, the number of results should not be zero; otherwise, the users must select other keywords.

Questionnaire Form

System coverage

1. Do the provided questions cover your need for inquiry the system?
Score: 50
No coverage Most coverage

2. Do the provided conditions cover your need for inquiry the system?
Score: 50
No coverage Most coverage

3. Do the provided features cover your need for inquiry the system?
Score: 50
No coverage Most coverage

FIGURE 8.4: Questionnaire page

Finally, the feedback process moved to the final page, a questionnaire page (Figure 8.4). The users could give a score or left comments to questions. I used a technique called visual analysis scale (VAS) to scale scores to each question. The

submit button locating at the bottom of the page should be clicked after all questions were completely fulfilled.

8.4 User Feedback Evaluation

In this section, I mainly described the experiment results, evaluation, and analysis.

8.4.1 Participants

Ten participants attended the evaluation process. According to the limitation of the domains of the dataset, the participants should study or experience about either biology or computer, because they needed to consider the obtained results whether relevance or irrelevance based on their keywords and intention. One participant has an experience about biology, and the rest have studied about computer. Their nationality is Thai, and their age is between 25-35 years old. An average of ages is 31 years old. A standard deviation is 3.03. Two participants are male, and the rest are female. However, gender is a trivial factor for my evaluation.

8.4.2 Results and Analysis

My dataset was gathered from academic literatures whose sources were IEEE Xplore Digital Library, PubMed, and Google scholar. Note that publications obtained from Google scholar related to keywords such as machine learning, data mining, and artificial intelligent. The total number of data were 636 images separated to biology 138 images and computer 498 images. There were two types of graphs: 170 images for bar graph and 466 images for 2Dcharts. The images were collected from several publications. Figure 8.5 presents the keywords that each participant selected for each experiment iteration. Also, the data applicable to the system was only graph images whose types were bar graph and 2Dchart.

Participant	Keywords
1	Protein,antibody
	PBMCs
	cancer
2	sensitivity,specificity
	image,training
	performance,prediction
3	accuracy,comparison
	temperature,comparison
	image
4	emotions
	accuracy,sensitivity,specificity
	heart rate
5	ANN,error
	accuracy,precision
	outlier,ANN
6	predict,regression,coefficient
	predict,back propagation
	linear,regression,predict,ANN
7	image,classification
	comparison,accuracy
	test,training,SVM
8	cluster
	test,training
	comparison,performance
9	intrusions,classifiers
	comparison,SVM,ANN
	error rates,prediction
10	SVM,ANN,comparison
	decision tree,accuracy
	image,classification,accuracy

FIGURE 8.5: Selected keywords for each participant and experiment iteration

After 10 participants completely tested and evaluated on both systems. I statistically analyzed the results by computing three performance measurements: precision, recall, and F-measure, as demonstrated in Figure 8.6. Precision is a ratio of retrieved instances identified as relevance. The recall is a ratio of relevant instances that are retrieved. Both performance models are good measurements to deal an imbalance dataset. For example, the data with relevant class is very rare compared to the irrelevant ones; in another word, the precision and recall depends on how rare is the positive class existed in the dataset, and they are mostly used when the positive class is more interesting than the negative one. Moreover, F-measure is a mean between precision and recall representing an accuracy of the test and how the quality of the system.

Participant	Elasticsearch			Ontology		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.067	1.000	0.125	1.000	1.000	1.000
	1.000	1.000	1.000	1.000	1.000	1.000
	0.700	0.700	0.700	0.900	0.900	0.900
2	0.500	0.214	0.300	1.000	0.143	0.250
	0.125	1.000	0.222	1.000	1.000	1.000
	0.583	0.389	0.467	1.000	1.000	1.000
3	1.000	0.667	0.800	0.667	0.381	0.485
	0.706	1.000	0.828	0.800	0.571	0.667
	0.000	0.000	0.000	1.000	1.000	1.000
4	1.000	0.500	0.667	1.000	0.500	0.667
	0.143	0.333	0.200	1.000	0.667	0.800
	0.000	0.000	0.000	0.000	0.000	0.000
5	0.813	0.481	0.605	0.920	0.852	0.885
	0.000	0.000	0.000	1.000	1.000	1.000
	0.167	1.000	0.286	1.000	1.000	1.000
6	0.333	1.000	0.500	1.000	1.000	1.000
	0.120	1.000	0.214	1.000	0.667	0.800
	0.389	1.000	0.560	1.000	0.571	0.727
7	0.286	0.400	0.333	0.750	0.600	0.667
	0.900	0.692	0.783	1.000	0.385	0.556
	0.526	1.000	0.690	1.000	0.600	0.750
8	0.800	0.500	0.615	0.667	0.500	0.571
	0.500	0.667	0.571	1.000	1.000	1.000
	0.875	0.875	0.875	1.000	1.000	1.000
9	0.300	1.000	0.462	1.000	0.333	0.500
	0.455	1.000	0.625	1.000	0.200	0.333
	0.667	1.000	0.800	1.000	0.143	0.250
10	0.278	1.000	0.435	1.000	0.200	0.333
	0.455	0.833	0.588	1.000	0.167	0.286
	0.200	1.000	0.353	1.000	0.500	0.667

FIGURE 8.6: Statistical results analyzed by three performance models: precision, recall, and F-measure

Hereafter, I showed the results, including their critical viewpoints.

Based on my observation in Figure 8.5 and 8.6, I noticed that even the participants selected the same keywords, the performance models might not be equal. For example, the keywords from Participant 3 at the first iteration and Participant 7 at the second iteration were defined as “accuracy, comparison”. This situation was happened because of two reasons. First, particular settings had been performed. Illustrating that a participant might choose a condition that allowed only results typed as a bar graph; whilst another participant did not set any condition. Then, the obtained results might differ due to the different settings. Second, the participants were determiners to decide the results whether relevance or irrelevance; thus, the decision might be different depended on their consideration.

Before I proceeded the experiments, I defined a hypothesis that such results retrieved from the ontology-based search system should be outperformed than the traditional one, i.e., ES. Most results (Figure 8.6) agreed my hypothesis, but a few results did not. The ontology-based search engine could acquire the relevant results by using AND operator; since they certainly matched to an intention of participants. Unfortunately, a number of retrieved results were sometimes too small because only exact matches had been obtained by the systems that caused a small amount of recall. As the recall from Participant 9 and 10, the participants selected some specific keywords, and only one result was acquired on each iteration. They decided it as relevance; hence, the precision was high, as opposed to recall. In my dataset, there were some results relevant to the keywords, but they could not show on a screen. They could not find the certain keywords, but their synonyms or related words had been discovered. For example, the “decision tree, accuracy” keywords were selected by Participant 10. She needed to examine the accuracy of decision tree algorithm. Note that several documents collected in the dataset indirectly mentioned about decision tree algorithm. They used the decision tree algorithm name instead, e.g., J48. My system could return the result containing both keywords but could not for J48. ES could obtain an amount of results that related to “accuracy” which accidentally matched to J48. Therefore, the recall of ES was higher, but the precision was lower than my system. In a particular situation, my proposed system provided low recall because of a small size of dataset. However, high precision was obtained by my proposed system because most retrieved results were relevant. If the size of the dataset was extended, the problem of the low recall should be solved, and the high precision might be served ideally.

The precision and recall from Participant 4 in the last iteration were zero because no relevant results were returned from both systems. She used a keyword “heart rate”; unfortunately, there was not any data in my dataset relating to the “heart rate” Based on my inspection, most returned results were regarded as heart disease and did not mention about heart rate. Moreover, this issue also happened with Participant 3 in the last iteration of the ES-based system. She selected a very simple keyword “image” to find graphs that related about image. However, the number of results retrieved from ES was zero because of her query setting. She required

only the image that the “image” keyword existed inside the graph, for instance, in X- or Y-titles. The ES-based system could not support this requirement. This was an evidence that my system could handle this specification, and the participants could literately obtain the relevant results.

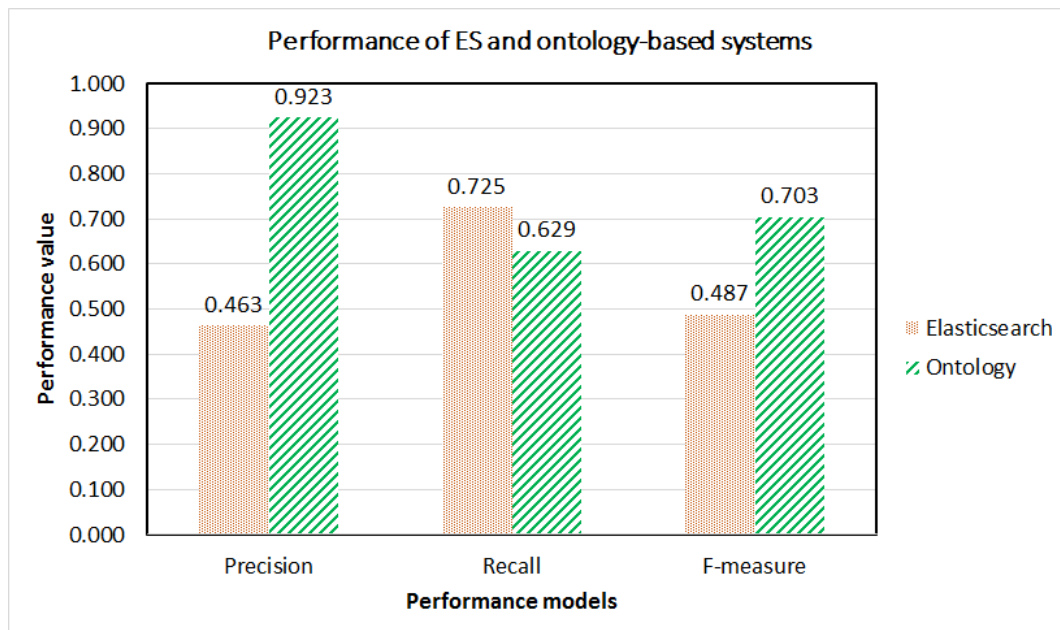


FIGURE 8.7: Average precision, recall, and F-measure from ES-based and ontology-based search engine systems

Regarding performance of both systems, I briefly analyzed and computed the results with the three performance models as average values, as presented in Figure 8.7. Obviously, the precision of my system was much higher than the ES because the participants considered that my system could mostly provide relevant results by using the specific questions, condition, and features; meanwhile, the ES-based search engine system provided the results based on only given keywords. However, the recall of my system was lower compared to the ES-based system, because currently, the ontology-based system did not support synonym or related words. Fortunately, this problem could be simply solved by connecting to other ontologies, such as DBpedia, to inquire about other related words and use them as extra keywords.

To compare the performance between both systems, this was difficult to use

either precision or recall to consider the system performance. Therefore, I computed the F-measure, which is the harmonic mean of precision and recall. After I analyzed it, the F-measure from my system was clearly much higher than ES one. In general, the high F-measure represents the better system performance. In addition, in the questionnaire page, the participants gave scores to questions asking about system coverage, usability, and functionality. An average score of Question 9 was one of supportive evidence to evaluate the system performance (Figure 8.8). Hence, my system was confidently outperformed than the ES-based system.

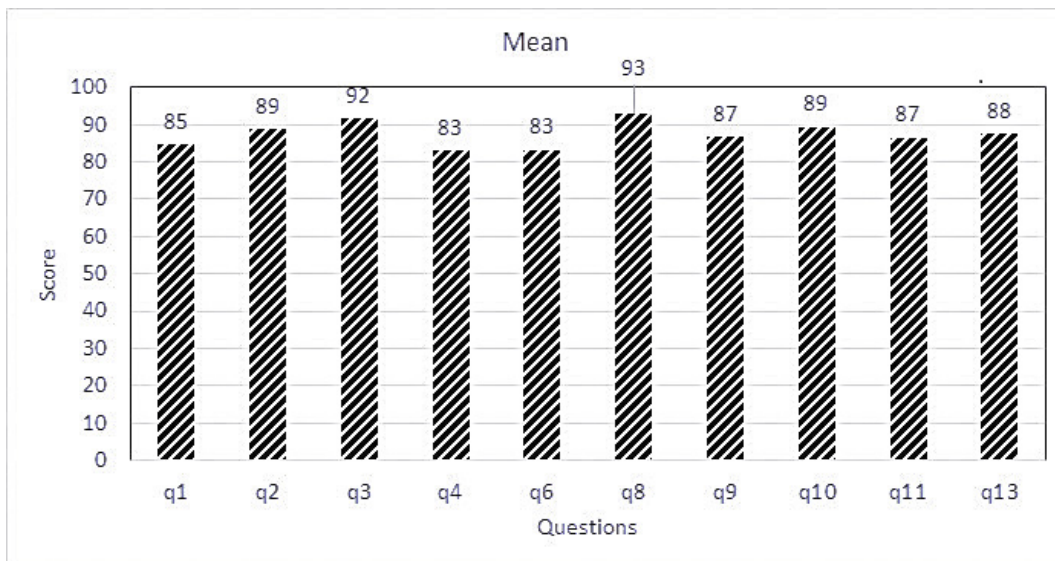


FIGURE 8.8: Mean of scores

Question numbers	Questions
q1	Do the provided questions cover your need for inquiry the system?
q2	Do the provided conditions cover your need for inquiry the system?
q3	Do the provided features cover your need for inquiry the system?
q4	Is the system easy to use?
q6	Ask about system presentation, is the system suitable to use for information inquiry and its results representation?
q8	How do you think about the successive seeking time of the system?
q9	Do you think that the system provides accurate results from your queries?
q10	To prove the system validation, do the system provide wider information due to using ontology when comparing the traditional search engine?
q11	Is the system applicable or useful for your study?
q13	Is the system able to handle errors, unexpected situations, or capture any anomalies?

FIGURE 8.9: List of questions

Figure 8.9 shows a list of questions. Question 1 to 3 represent system coverage. For example, “do the provided questions cover your need for inquiry the system?” Question 4 to 11 ask about system usability, such as how suitable a layout of the user interface, how speed, how accuracy, and how useful to a study. The rest,

i.e., Question 13, asks about functionality, such as error handling. Note that some question numbers are skipped because they are comments.

I considered the obtained scores of each question. I focused on the Question 1, 2, 3, 9, and 11 because they were very important questions to validate the system. I assumed a range of satisfaction as showing follows:

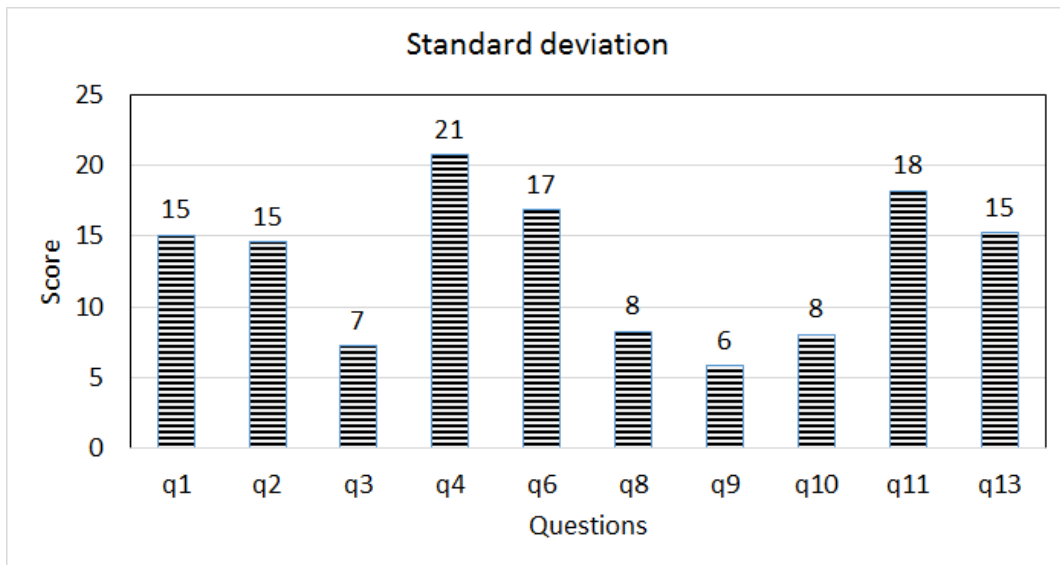


FIGURE 8.10: Standard deviation of scores

Participant	Score									
	q1	q2	q3	q4	q6	q8	q9	q10	q11	q13
1	91	94	94	90	80	91	85	90	95	100
2	90	95	95	95	90	100	95	91	95	89
3	50	58	80	50	50	100	75	80	40	50
4	100	100	100	100	100	89	90	100	100	100
5	90	100	90	90	100	100	85	85	90	100
6	91	99	99	100	81	100	91	100	99	82
7	85	81	83	90	87	80	86	87	77	85
8	80	70	85	40	60	90	80	75	80	80
9	70	90	90	85	100	80	90	90	100	95
10	100	100	100	91	85	100	90	95	90	95

FIGURE 8.11: Scores of each question in Questionnaire page provided by 10 participants

- 100-80 = Very satisfied
- 79-60 = Satisfied

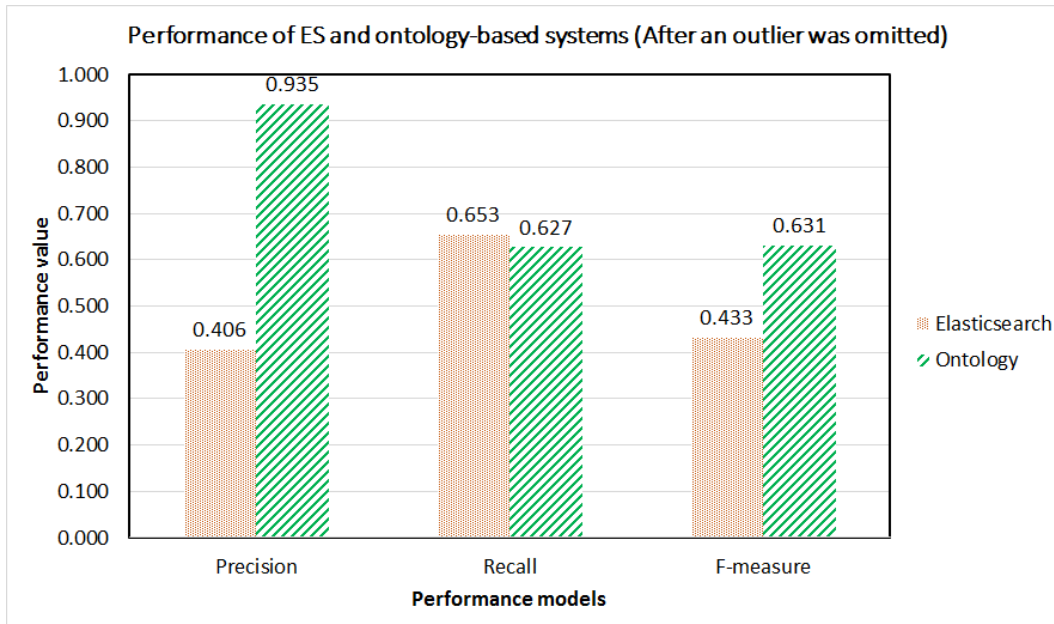


FIGURE 8.12: True performance of the search engine system without outliers

- 59-40 = Neural
- 39-20 = Poor
- 19-0 = Bad

The average scores from those questions were classified as Very satisfied. This could be concluded that my system was suitable to open up a new way for a novel technique of information retrieval because not only the high performance was presented as described above, but the participants also felt comfortable to use the system because it could support the research studies of the participants as displayed in Figure 8.8 at Question 11.

Here, I analyzed scores representing the satisfaction values provided by the participants. Figure 8.11 depicts the assigned scores of questions for each participant. As my observation, a participant gave some comments because she thought that the system should improve somehow due to less score of Question 1, 2, and 11 obtained by Participant 3. Her scores were not in a normal range of standard deviation (Figure

8.10). Her opinions were about a small volume of the dataset. As described my data collection, the size of the dataset was 636 graph images; since she possibly did not obtain any result from the system if she used too specific keywords. Moreover, she is interested in video comparison and temporal comparison; unfortunately, my computer dataset domain was only about data mining and machine learning.

To show the true performance of the system, I tried to omit outliers from the results; hence, the results from Participant 3 should be removed. Then, the precision of my system slightly rose from 0.923 to 0.935. The recall of ES-based system reduced after the outlier was omitted that caused the similar recall value to the ontology-based search engine system. However, F-measure of both systems trivially decreased, but the difference of the value was not changed. Figure 8.12 depicts the true performance of both search engine systems that already omitted outliers from the results.

Chapter 9

Discussion

In Chapter 8, I presented about the experiments and evaluations comparing between ES-based and ontology-based search engine systems. The experiments had been carried out by 10 participants, who have experienced in biology or computer domains. They used the systems to inquire three questions with specific keywords that related to biology or computer domains. They analyzed obtained results and decided which either relevance or irrelevance class should be selected. As the results in the previous chapter, my system was evidently outperformed than the traditional system (ES) because the F-measure of my system was higher than the other one.

This chapter is devoted to the discussion of this dissertation. The significant findings from the proposed methods and experiments should be described here. Moreover, I will introduce an extended idea and possibilities based on facts and the findings.

In the first section, I will summarize what I have done for this dissertation, including the core findings found in each study. Next, I will discuss them and introduce possible ways to improve the studies.

9.1 Findings of this dissertation

Nowadays, information retrieval plays a vital role in daily life due to a huge volume of data available on the Internet. This is surely difficult to people to seek what they need on the Internet manually. The search engine is a solution to this issue. Until now, research on information retrieval has been further advanced by utilizing semantic concepts, such as data mining and ontology. I attempted to address critical problems that could be solved by my proposed methods. In this dissertation, I proposed the methods to construct a novel search engine system applicable to an ontology storing graph information. I completed many works, designed possible solutions, conducted experiments, and discussed new findings discovered from evaluation results.

In Table 9.1, I summarized and listed the works done in my dissertation, including the core findings of each, as follows:

TABLE 9.1: List of studies and their core findings

Methods	Findings
Graph-type classification	ANNSVM is outperformed than CNNs
Graph components extraction and identification	Adjustable Epsilon is suitable to locate the position of graph components
Graph-based OCR correction	The results suggested by ontology is more accurate than the ones suggested by edit distance
Graph information extraction	More relevant and extended information are retrieved after querying on ontology
Prototype of graph-based search engine system	F-measure of the proposed system is higher than the traditional system

In Chapter 3, I introduced the graph classification method. It could distinguish between the bar graph and the 2Dchart by using discrete Fourier transform, Hough transformation, and wavelet transformation. I conducted experiments by comparing the performance of systems dealing with several different datasets. Based on the experimental results, I found that CNN was an unsuitable method to deal with the

graph image due to low accuracy. The wavelet coefficients could identify the dominant characteristics from the graphs outperforming than the Hough transformation, especially Coiflet 1. However, to make my dataset more separable, I decided to assemble the result data from wavelet coefficients and Hough transformation together. ANNSVM provided a remarkable accuracy, up to 0.91. Note that this system can classify the types of graphs, i.e. bar graph and 2Dchart, effectively. During I implemented the search engine system (the final system), I could distinguish the types manually because a number of data were small. Moreover, to show a concept of the classification system, I omitted this process to increase a precision of the overall system. However, if the size of data volume was increased, this system was surely necessary.

I presented the graph components extraction and identification system presented in Chapter 4. To accomplish the problem, I adapted DBSCAN to automatically estimate a proper Epsilon value for each data. I evaluated the system by benchmarking the performance between the proposed system with flexible Epsilon and another with fixed Epsilon. The proposed method provided very high identification rate, 0.93; whereas the other was only 0.31. This was because it returned the results as original images that contained many irrelevant parts. In other words, DBSCAN with fixed Epsilon could not detect the legend properly.

The OCR-error correction had been proposed in Chapter 5. I used a constructed ontology and other online ontology to suggest correct words to error ones. The results of correction were validated by comparing the performance between the proposed method, which used an ontology, and a traditional one, which used edit distance. The proposed method provided the highest F-measure and accuracy, 0.86 and 0.84 respectively.

Not only the graph components are important features to create the ontology for my semantic search engine, but also graph information resided in a data section of the graph is necessary. I proposed a method of graph information extraction in Chapter 6. It used to extract tendency of data plots, bar height, and significant relationship of graphs. I made some simulations to prove the validation of the system. I found that I could obtain a number of tokens unavailable in the captions of the

graphs but were instead taken from other graphs sharing the same concepts, such as quantification and plasma. This proved that my proposed method could introduce new knowledge by utilizing ontology concept.

I integrated entire proposed systems to a semantic search engine system in Chapter 7. I implemented a web-based application connected to my constructed ontology. The ontology recorded all information extracted from the previous methods. The F-measure from my system was much higher than the ES-based system, which represented a better quality of the search engine system. This summary also agreeable to the result from questionnaires gathered from the participants.

From the results analysis of the whole study, essential findings in this dissertation had been found based on the facts of the study. They were listed as below.

- CNN was unsuitable for classifying images if the image edge was only a trivial characteristics for classification.
- The most suitable wavelet coefficient applied to my dataset was Coiflet 1.
- The wavelet coefficients could identify the dominant characteristics from the graphs outperforming than the Hough transformation.
- An order of algorithms was a trivial matter; in other words, SVMANN and ANNSVM were similar.
- The 5-layer ANNs is applicable to my target data.
- Based on my observation, the graph component extraction system can excellently handle the graph image who legend quite separates from other surroundings.
- The graph component extraction can support the OCR-error correction system because it helps to reduce the noise ratio from the input images.
- The correct word suggested from the ontology is more accurate than edit distance.

- In my opinion, a scope of the question to query the information from graphs was more expanded than a traditional search engine that uses keywords and obtains document relevant to the keywords.
- Ontology offers concise knowledge that cannot be done by the traditional search engine.
- Ontology-based search engine provides extended information outperforming than the traditional one.
- Users are satisfied to use the ontology-based search system because most users feel comfortable to use it.

During the studies, not only essential findings, which are corresponding to the facts from the experiments, but the byproduct findings should also be described here. They are the findings derived from the experiments but do not relate to the core of the studies. As the following list, I showed the byproduct findings found in this research.

- A low image quality highly affects to the performance to the systems.
- Irrelevant parts included in the extractable legend also negatively affects classification performance.
- A query time in SPARQL was longer than querying in a database.
- The input image should be cleaned as much as possible to prevent OCR misunderstanding because they interfere the recognition process.
- To realize the characteristics of the frequency domain in an image containing texts, the high-frequency area is denser than other areas in the DFT images. Similar to the image not containing texts, a difference is that its density in the high-frequency area is smaller than the image containing texts. To sum up, the image with texts has a higher frequency than the one without texts.
- Based on my ontology structure, it is possible to indicate a category that a graph belongs to by using NER class.

- Regards the OCR process, a space between alphabet characters in a word also provides a confusion to the OCR because the OCR may recognize the word separately even it should be one word.

I discovered several interesting findings that I had never expected before. I used the ontology-based search engine system and selected a question about graph relationships. I inputted some keywords and obtained the results that I required for. I obtained several graphs corresponding to the keywords. After I collected the results of graph relationships, I found some similarities. Surprisingly, the obtained graph axis relationships provided around 10% to 20% similarity when compared to the graphs corresponding to the keywords. For example, the keywords “accuracy, performance” were inputted to my search engine system and obtain five graphs relating to the keywords, including their graph axis relationships. I examined the similarity among the graphs. The proportion of the graph axis relationship’s similarity was 20%. This represented that the graphs related to the same keywords should have corresponding relationships partially. This finding can be used to discover other graphs that contain the similar relationships in other documents.

Moreover, I examined the results of token relationships by inquiring the system. The ontology-based search system can cope a variant expression of words. I simulated some keywords that represented a similar expression and inputted them to the system to find their token relationships. The results showed that some same relationships were acquired. The rate of identical relationships appeared was around 30% to 40%. Therefore, the similar keyword expression can obtain the identical relationships. For example, I inputted two set of keywords: “method, performance” for the first iteration and “algorithm, performance” for the second iteration. I collected the relationships from both iteration and compared them to find the identical relationships. I obtained an identical ratio about 36%. With this finding, it is possible to find other relevant description contents, e.g., paragraphs, in multiple documents. For example, I obtain a token relationship in a sentence, and I find other paragraphs that contain the same token relationship. Since, the relevant description contents should be expanded.

9.2 Discussion of this dissertation

This dissertation aimed to address many problems that can be solved by each system. The problem of the semantic gap, which was the core problem, could be addressed by my main system. The results obtained from the final system represented the end results of the whole system because I integrated the systems to create the data for the final system and used it to interpret and express the output to users.

The critical point needed to be discussed here was how the whole system can achieve the problem of the semantic gap. The main idea of this research was to propose methods extracting information from graphical and linguistic representations as well as utilize them to express explicit and implicit knowledge. I used the method of graph component extraction and graph information extraction to extract information from the graphical data, including graphs' linguistic data, e.g. caption. Then, they were recorded into ontology; since the final system was an application to show the results from the ontology.

As regards to the ES-based system, it was a keyword-based search system; since, it could not provide the information located in graphical images, only image descriptions available.

I measured the true performance of the whole systems corresponding to the results obtained from the final system that already omitted outliers. As the results, the precision was greatly high comparing to the ES-based system, 0.935 and 0.406 respectively. However, the recall of my system was slightly lower. To analyze the recall of both systems, my system offered a lower value, but it was almost similar to the recall of the ES-based system. In another word, the difference between those recall values, which was equal to 0.02, was very trivial. Thus, if I analyzed the recall only, the performances from both systems were very similar. Regards the F-measure, my system provided higher F-measure than the other. This means that my system represented the better performance. To sum up, based on the evidence described here, my ontology-based search engine system can overcome the addressed problem, because it offers the higher precision and F-measure comparing to the ES-based search engine system.

At the current state, I attempted to propose several methods as well as designed database and ontology schemes to achieve the main objectives of the research. The performance of entire systems had been evaluated as I presented in the previous chapters. However, there are many ways to improve the performance of the system.

9.3 Limitations and possibilities of the study

I will introduce their limitations and improvement ways for every system including an explanation of the target data. Moreover, based on the findings, I extend ideas to introduce possibilities that are the ways to utilize the findings to other study directions. Table 9.2 demonstrates a summarization of limitations for each proposed systems.

Regarding the target data of this study, a collection of graph images had been used for the experiments of each system. The data were gathered from the scientific literature. This system had covered computer science and biology domains. The graphs from different domains provide a diversity of data expression. In a viewpoint of the data domain, a possible solution to handle the variant data expression is to integrate ontologies from other's domains, such as Physics or Biology. In contrast, about different graph structure aspect, my system can extract precise information from graphs with the general structure, not including tree graph or network. To deal with other kinds of graph structures, it is necessary to propose particular methods to extract information from them because of a diversity of graph expression existed. There are many ontologies publishing on the Internet. To extend my ontology, I need to merge mine and other ontologies together. A coverage of the system should depend on what domain of the ontologies is integrated with. However, to integrate them, this is necessary to take into account to ontology alignment. Kinds of interoperability are limited because a minimal change has been required for ontology schemes in order to merge inter-ontologies. Thus, it is important to standardize my ontology scheme compatible to the merged ontologies. To do so, before creating the ontology, I should examine the schemes of merged ontologies in advance and attempt to seek what concept can be connected. Moreover, a merging process can be performed

in many particular ways, such as manually, semi-automatically, or automatically. Manual ontology merging is highly labor-intensive; hence, semi or entirely automated techniques are definitely preferable. To do this, a similarity of concept relationships should be examined. A merging system traces along relationships through ontologies and observes which parts contain similar concepts and relationships. Also, they may realize similarity of concepts through textual string metrics, e.g., edit distance, including semantic knowledge and relationships. There are many kinds of graphs available in the literature. In this study, I limited to a kind of graph presented as a general structure, such as bar graph and plot graph, because they have been often used in the scientific literature rather than other graph types. They are suitable to convey the statistical data or compare results. As mentioned in Chapter 1, only two types of graphs have been used in this research: bar graph and 2Dchart. The system is highly applicable to these data supportive by obtaining high accuracy as shown in the experiments. However, if I deal with either bar graph or 2Dchart, system performance may increase somehow because of no classification errors.

Methods	Limitations
Graph-type classification	<ul style="list-style-type: none">• The system could deal only simple object's, such as a circle and a rectangle, but could not deal complex shapes because a part of my system, i.e., Hough transformation, is not applicable to detect them.• Suitable wavelet families would vary to input data.• I assigned 5-layer ANNs because the input data are non-linear separable. However, with other data, the number of layers might be changed. The 5-layer ANNs does not valid to every data.

<p>Graph components extraction and identification</p>	<ul style="list-style-type: none"> • This system would not accurate if the graph contained sparse data due to the DBSCAN's concept. • Also, the DBSCAN was not suitable to handle very high density data because it would be not able separate too dense data to clusters. • It consumed much time processing because the automatic Epsilon estimation would iteratively check data several times to acquire the most possible Epsilon. • If a graph legend did not locate at the top or right side of graphs, this system could not detect it accurately.
<p>Graph-based OCR correction</p>	<ul style="list-style-type: none"> • An overall performance of the system was highly dependent to online ontologies, such as DBpedia and WordNets. If their endpoints are not available, the system will be not able to service users. • The system could deal with only English letters.
<p>Graph information extraction</p>	<ul style="list-style-type: none"> • Only single label of bar graph and 2Dchart as well as a multiple data labels in bar graph were applicable to the system. • The system could provide precise bar heights if the Y-scale started with zero.

A prototype of graph-based search engine system	<ul style="list-style-type: none">• The system would not rank results by their relevance but order of appearance in my ontology.• It still required keywords to specify the results.• It could not filter unnecessary results before providing to users.• It could not provide literature information such as an author name and a source.
---	---

TABLE 9.2: List of limitations of all proposed systems.

For the graph-type classification, I used a number of techniques to prepare the dataset, including proposing a new classification, called ANNSVM. Experimental results of this system were achievable. However, for the future benefit, the process of the system should be simpler than the current one. I received some comments from an expert about its complexity because there were too many steps for preparing the dataset and classifying the types. A possible solution is to reduce the number of processes by removing unnecessary steps. For example, I skip a process of Hough transformation because the finding confirmed that wavelet coefficients could identify the dominant characteristics better than Hough transformation. However, it is important to maintain the system performance in spite of omitting the Hough transformation process. To do so, I will priorly clean irrelevant features, e.g., image background, from the images before the classification process in order to emit the graph characteristics and omit the unrelated parts. The algorithms, i.e., SVMs and ANNs require predefined parameters. To advance this system, a system of parameter estimation may need to be integrated. Based on the finding, CNNs was unsuitable to cope the graph images but effectively classified the photo images. To extend classifiable graph types, I should use CNNs to classify the graph types whose dominant characteristics was color, such as pie charts, area chart, and 3-dimensional

bar graph. In this research, I did not take into account to 3-dimensional graphs. If I use the system to analyze the 3-dimensional graphs, it may result in misclassification that leads a failure to the graph information extraction system. In the classification process, ANNs cooperated to SVMs. The number of hidden layers had been fixed to five layers. If the number of layer increases, the classification results may not be much different from the extant five-layers ANNs, because my data is nonlinearly separable due to a contribution of wavelet coefficient. Wavelet coefficients calculate at every possible scale and along every time instant and represent the similarity extent comparing the examined section of signals to the scaled and shifted wavelets. Generally, the high value of coefficient provides the greater the similarity between the wavelets and the original signal and via versa. This is the main reason why the wavelet coefficient expresses the dominant characteristics of the graphs. Based on this fact, I realize that the results possibly apply to other algorithms, not limit to only classification, for example, clustering. For deep discussion, I will use a clustering algorithm to analyze the graphs in the same group and identify correlated characteristics; moreover, I may realize exceptional characteristics from analyzing outliers.

To detect the graph component, this can be handled by using the graph component identification and extraction system. I used DBSCAN to cluster the data plots that stay close to each other based on data density. To locate the high-density data, Epsilon is an important factor, but it is a user-defined value. Therefore, this system can automatically define Epsilon by analyzing the density of data. To enhance the system quality, I should improve the speed of the system because it needed extra time to analyze the data for defining Epsilon. To solve this, the processed data should be reduced to decrease time-cost for processing, for example, using sub-sampling. During the processes, an image preprocessing step should be assembled to the system because the clustering results were sometimes incorrect due to image noise and irrelevant data. Moreover, the graphs do not always contain legends due to single data representation. In this case, an existence of legend needs to be identified and confirmed beforehand by a system. It should analyze descriptive information of the graphs and decide the graphs containing whether single or multiple data. MinPts

is another parameter priorly defined by users. If MinPts is assigned with a suitable value, DBSCAN may offer good clustering results. For example, during the experiments, I observed that I obtained only one cluster from the proposed system because the system could not separate data into independent groups. If I obtain a suitable MinPts, this problem may be solved. Moreover, I reviewed several documents about DBSCAN, I perceived a clustering technique, named OPTICS. Its basic idea is similar to DBSCAN, but it addresses a problem of detecting meaningful clusters in data of varying density that occurs in DBSCAN. It also requires two parameters same as DBSCAN. If I decide to use OPTICS instead of DBSCAN, the obtained results may be nearly indistinguishable because both algorithms use the same Epsilon value provided by my system for clustering due to same inputted data. Regarding a contribution of this system, it is used in many kinds of data, not limit to images, because this idea is proposed based on a natural of an algorithm that clusters data by analyzing density.

For OCR-error correction, the ontology was constructed by using descriptive contents and other ontologies. It suggested correct words to errors effectively. The system needs a support when performing to vocabularies come from other specific domains, such as mathematics and biology, because there are untranslatable vocabularies which are rarely found in a general dictionary, such as a scientific name. Additionally, this system hardly deals with words containing non-English alphabet, such as Greek alphabet, that usually found on mathematical documents. At this state, I used an English language pack for OCR. Basically, my ontology had been supported globalization. However, some localized tools should be changed, such as dependency parser and OCR language pack, because they should be compatible with a target language for preventing any errors. Moreover, a system analyzable the context of sentences may be necessary to accurately select corrected suggestions. For example, in a sentence describing the weather, there are two words suggested by ontology in order to correct OCR errors. The system should select the one that highly relates to the sentence context. This idea may be able to adapt to a generic thesaurus, e.g., WordNet, to find a word candidate in graph structures of vocabulary. Another idea is to use Google word suggestion system to support my OCR-error correction system to select corrected candidates. Moreover, a genetic algorithm may be

a proper solution to improve an efficiency of words suggestion of this system because this algorithm is used for optimization which helps to offer the most suitable word to the system. As described in Section 5.2.2.3 of Chapter 5, I introduced a dictionary named DepDic used to records the chain dependencies of the tokens. To support this process, n-gram should be another technique to create a vocabulary storage. It decomposes each string in sentences into letters. In my idea, it may be used to find word candidates.

The graph information extraction was proposed to extract the graph information established in the data section as well as to construct the database and ontology based on the designs. During the extraction, I found some errors from a process of bar height measurement. A cause of the error was OCR misrecognition. For this step, I isolated a scale part of Y-axis from a graph and used OCR to recognize scale numbers to calculate a scale ratio. To solve this, my proposed OCR error correction is a suitable option. To do so, the OCR error correction should be adapted to be workable with numbers, not words. To obtain the bar height, this system works well with the images that have a standardized layout. For example, Y-axis contains a scale started by zero. Also, it is effectively applicable to simple graph images. If the graph contains too complex information such as noise, the current state of this system may provide inaccurate extracted information that leads incorrect interpretation. To enhance knowledge, I may interpret the quantitative data extracted from the data section of the graph (e.g., bar heights and tendency) and map them to ontology. For example, I interpret the bar graph and obtain tendency. In a context of its description, it describes a trend of data with some explanation, as similar to another graph which contains a related description. Based on this example, these relevant graphs acquire extended information according to shared concepts. Moreover, I may obtain unexpected information from other knowledge domains. This will happen if I merge other ontologies with my ontology. For example, the data stored to my ontology related to information technology. If I integrate biology ontology to my existing ontology, I may acquire interesting knowledge relating to not only biology but also information technology. Based on this example, if I attempt to query towards biology ontology about a protein name, I should acquire data relating to information technology, such as statistical data about the protein, intelligent algorithms relating

the protein, and relationships between the protein and other measurements. My system could identify a regression type of data; thus, I possibly predict unseen data by applying statistic analysis, such as linear or non-linear regression.

The final system is a prototype of ontology-based search engine system. This system utilizes entire systems proposed in this dissertation. Regarding limitations, this system does not support a lemma technique yet. Note that the lemma is a technique to change a word to its root. There are some libraries available on the Internet. If I integrate a lemma process to my system, the obtained results should be enlarged, and new knowledge is also delivered. Moreover, it cannot separate between stop words or rare words. This problem will be solved if I use text mining technique. The size of the dataset was limited and specified only two domains. To cover the users' needs, the data volume should be expanded. My ontology should be integrated with other ontologies to enlarge data source. In Question 2 of this system presented in Chapter 7, I investigated the main idea based on sentences containing keywords and the first sentence of the paragraph. However, to precisely obtain the main idea, I should utilize text summarization, which is a text mining technique, to summarize the whole paragraphs and show only a core part of paragraphs. Additionally, I obtained the unexpected findings by observing the results of relationships. The partial relationships should be useful if I input them into ontology because I may discover new knowledge by tracking other relations on the ontology. In another aspect, I may cluster the graph relationships based on their shared relationships by using a graph or network clustering and find some similarities on the graphs belonging to the same group. Moreover, if I utilize deep learning to the system, it is possible to develop a question answering system based on my ontology. This function surely facilitates users to speedily obtain desired answers. Further, the deep learning is used for matching between text and image. They represent as vectors and using a deep learning technique, e.g., CNNs, analyzes and matches the two vectors. If this is used to my system, the obtained results will be unlimited to only graphs but included other kinds of images. For example, a user needs to query the system by using a keyword "compiler", my system will provide graphs showing statistical data about "compiler", including other images, such as compiler pictures. Currently, this system did not have a ranking feature. To order relevant graphs or documents, I

will use the deep learning to rank the results by analyzing user interactions, such as a click. Furthermore, based on the system's ability, it is possible to develop a new function integrated to my system to suggest or recommend publications to readers. When they use my existed system to query relevant graphs corresponding to their keywords, some relationships have been discovered in the graphs. The new function recommends the publications corresponding to the relationships; since the readers can decide which documents are worth to read. Regarding the ontology creation, the ontology scheme maybe able to deduce by data itself. If there is a system that can analyze the data and result some existed concepts and relations, this is possible to create the ontology scheme automatically. In the present, existing technologies may be not suffice to handle all information in variant graphs. The most difficult issue should be a method to realize the data in the graphs perfectly because there are many image noises and unnecessary information that need to be omitted beforehand.

During the experiments and evaluations, I received many useful feedbacks and comments from the participants in order to improve the system usability as follows:

- At the result section of the search page, I should include sources of documents, such as a publication URL and a paper's title.
- I should enlarge the size of the dataset, including expanding data domains to cover all needs.
- I should redesign option selections in the search page to be simpler.
- The layout of the prototype should be organized to prevent confusion.

As the comments above, they required a interface improvement to support user convenient. The participants did not deny the idea of method and my assumption supported by results from questionnaire and evaluation.

Here, I present practical contributions acquiring from this dissertation.

- New method of the graph-type classification system.

- New method of graph component extraction.
- Adaptive DBSCAN with automatic Epsilon estimation.
- New method of OCR-error correction using ontologies.
- New method of graph-content extraction to obtain knowledge from a data section of the graph.
- New prototype of ontology-based search engine system.
- New design of the relational database to collect typical graph information and user feedback evaluation.
- New ontology design supporting OCR-error correction and search system by storing extractable graph information and graph's descriptions.

In conclusion, this dissertation proposed several systems relating to extracting essential information from the graphs and also contributed many benefits to academic researchers. In this dissertation, I clarify that the ontology-based search engine system provides precise and concise graph information outperforming than the ES-based search engine system. It had been proved by the user feedback evaluation. The F-measure of the ontology-based search engine system was higher than another that represented a better performance. Moreover, based on the user questionnaire responded, the participants are satisfactory to use the system. The main contribution of this dissertation is the novel ontology-based search engine system together with the new design of ontology that is applicable to graph information.

Chapter 10

Conclusions and Future Works

In this final chapter, I summarize my dissertation. Moreover, I will imagine the possible future work.

10.1 Conclusions

This dissertation presented a novel search engine system that utilized ontology and a relational database, including proposing several methods for graph image information extraction. My main objectives of this dissertation were presented as follows:

- To narrow the problem of semantic gap.
- To distinguish the graph types and propose a new graph-type classification system.
- To extract and locate the graph components.
- To suggest a new solution of Epsilon estimation for DBSCAN.
- To design ontology for a semantic-based OCR-error correction system and search engine.

- To extract extended information from the data section of the graph.
- To create a prototype of ontology-based graph search engine system.
- To evaluate the ontology-based search engine system with a traditional search engine system.

I principally addressed the problem of semantic gap. I conducted several experiments and evaluated the obtained results. It clearly showed that the system can identify and extract information from graph image. Moreover, the information were included into ontology integrated to the search system. As the results, my system can provide the information to users via ontology. Since, I guarantee that the problem of semantic gap was already solved by this research.

To achieve the objective of the system presented in Chapter 3, I introduced a new graph type classification system using several independent techniques to prepare and classify data, such as DFT, Hough transformation, and wavelet transformation. This system contributed benefits and support to my search engine system. To effectively seek specific results, this was necessary to divide the graph types beforehand and extract significant information based on particular types. It supported three graph types: bar graph, line and plot graphs, and pie chart. However, the pie chart was uninvolved to the search engine system. Summing up the results, the accuracy from the proposed method reached to 0.91. This was an evidence of high performance system.

For the graph component extraction system presented in Chapter 4, I proposed a method to identify and extract the graph components from the graphs, such as X-title, Y-title, and optionally legend. To obtain X- and Y-titles, the method was quite straightforward because they usually locate at the bottom and left sides of the graphs, as opposed to legend. It may or may not locate in the graph. To detect it, I used DBSCAN to capture and group the data based on data density. DBSCAN needs Epsilon and MinPts parameters that must be set before clustering. My system could provide suitable Epsilon automatically by analyzing data position. Moreover, after many graph components were retrieved, this system can identify which class

the image outputs belong to. Based on the results, it can be concluded that the research has been very successful because the accuracy rate for classification was up to 0.93.

To overcome the goal of the system presented in Chapter 5, I designed an ontology and construct a novel OCR-error correction system. After I obtained the graph components from the previous system, I used OCR to recognize and convert texts to digitalized data. However, the misrecognition might occur. This system coped this problematic by using a suggestion from the ontology. In experiments, I compared performance between ontology-based and edit distance-based OCR error corrections. As the results obtained from the experiments, I acquired high accuracy and F-measure: 0.84% and 0.86% respectively. Moreover, I considered about image noise that might be the critical factor to reduce the performance of the system. I used the proposed graph component extraction to obtain cleaned outputs. The results showed that the noise ratio was decreased comparing to a tradition image partition around 0.19%.

To fulfill my target of the research in Chapter 6, I must extract graph information located in the data section. I proposed a new graph information extraction to examine how high of bars, how trends of data, and significant relationships. Moreover, I designed an ontology and database that support both OCR-error correction and search engine system. To evaluate the system, I set up some simulations based on possible questions asked by users. I observed the obtained results from each simulation.

I integrated all implemented systems into one main system to extract the total information I needed from the graphs as well as constructed ontology and database. I programmed a web-based application applicable to search and query thought my constructed ontology created by all extractable graph information. Ten participants helped me to evaluate the systems. They could select specific questions, settings, and input some keywords. They considered the returned results as either relevance or irrelevance. I validated the performance between my ontology-based and ES-based search engine systems. As the results, I concluded that my ontology-based search engine systems provided better performance than the traditional one due to higher

F-measure obtained. Moreover, the result from a questionnaire was supported my conclusion.

Regarding the limitations of the study, this system had covered the data from computer science and biology domains. However, it was also applicable to other domains if we expanded the target data. Due to graph types and a kind of graph limited, it could express the information extracted only from bar graph and 2Dchart which were in a general graph structure.

In conclusion, I proposed the systems to extract the graphical and linguistic information from the graph image itself and its descriptions. The system provided the great performance measurements; since it proved that it can mitigate the semantic gap problem and achieve entire objectives. It clarified that the ontology-based search engine system provides precise and concise graph information outperforming than traditional search engine systems. The major contribution is not only the new method of ontology-based search engine system but also an ontology design supporting graph information and descriptions.

10.2 Future works

Further study of the issue would be of interest. To outline the directions of future work, the size of data should increase, and the domain of data should expand to other study areas. As mentioned about this research's contributions, the system will provide many benefits to researchers. Therefore, to extend the data, it should extensively cover users' needs. To enhance the system usability, a keyword or spelling suggestions may be necessary for the users, who do not know how to spell the keywords or do not have ideas to select keywords. It will utilize an intelligent technique, e.g., deep learning and data mining, to analyze user behaviors and suggest them the keywords. Another idea is to analyze description context to predict the user intention and offer some possible keywords. Currently, the system sometimes provides massive information to users. If the system can be improved by using advanced data mining techniques to make a decision which information is usable or unusable. Further, a question answering system will be introduced by deep learning

in the future. If these functions will be proposed, the ontology-based search engine is surely much more powerful. Moreover, this will be great if the ontology of this system is applicable to other domains of ontology, such as gene ontology or electronic ontology. Currently, the ontology had been designed and constructed manually. However, in the future, it is possible that there is a system that will generate the ontology automatically by referring some existing structures and relationships, such as dependency parsing in sentences. The usefulness of this system will be extended if it works with not only graph images but also other images. If that is possible, a system used for image interpretation will be feasible.

Appendix A

List of Publications

A.1 International Journal Papers

[J.1] Kanjanawattana, S. & Kimura, M. (2017). Extraction and identification of bar graph components by automatic epsilon estimation. *International Journal of Computer Theory and Engineering*, 9(4), 256-261.

[J.2] Kanjanawattana, S., & Kimura, M. (2017). Novel Ontologies-based Optical Character Recognition-error Correction Cooperating with Graph Component Extraction. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 7(4), 69-83.

[J.3] Kanjanawattana, S., & Kimura, M. (2017). ANNSVM: A Novel Method for Graph-Type Classification by Utilization of Fourier Transformation, Wavelet Transformation, and Hough Transformation. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 8(2), 5-25.

A.2 International Conference Papers (Peer-reviewed)

[C.1] Kanjanawattana, S. (2012, May). An extended K-means++ with mixed attributes. In *The 12th WSEAS International Conference on Applied Computer Science (ACS12)*, (pp. 131-135).

[C.2] Kanjanawattana, S. & Kimura, M. (2015, November). A proposal for a method of graph ontology by automatically extracting relationships between captions and x- and y-axis titles. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, (pp. 231-238).

[C.3] Kanjanawattana, S., & Kimura, M. (2015, December). Graph-type classification based on artificial neural networks and wavelet coefficients. In *Proceedings of Second International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC2015)*.

[C.4] Kanjanawattana, S., & Kimura, M. (2016, October) Ontologies-based Optical Character Recognition-error Correction Method for Bar Graphs. In *The Tenth International Conference on Advances in Semantic Processing (SEMAPRO 2016)*, (pp. 1-8).

[C.5] Kanjanawattana, S., & Kimura, M. (2016, December). Extraction of Graph Information Based on Image Contents and the Use of Ontology. *International Association for Development of the Information Society*, (pp. 19-26).

[C.6] Kanjanawattana, S., & Kimura, M. (2017, September). Semantic-based Search Engine System for Graph Images in Academic Literature. *1st EAI International Conference on Technology, Innovation, Entrepreneurship and Education*. In press.

A.3 Workshop

[W.1] Kanjanawattana, S., & Kimura, M. (2016, February). Graph-type classification with neural networks using wavelet coefficients and discrete fourier transformation. In *SEATUC2016 Intensive Workshop*.

Appendix B

Background of generic search engines

B.1 System Evaluation Background (Appendix part)

ES is an open-source distributed repository built on the top of Lucene and efficiently copes a huge amount of data. Moreover, it provides a simple and potent application programming interface (API) that allows many applications created from various programming languages access to the repository at ease. The most powerful feature in ES is a mapping process. It is similar to a schema definition in relational databases; in another word, it is a schema of every index in ES. Because of this feature, ES can comfortably deal with complex or multiple index. Moreover, ES accesses to databases only when it creates or update index. It stores the index and data into its repository that helps to increase the speed of adding new documents. ES offers the robust query domain specific language (query DSL) included in its search APIs [80], for example, *filter*, *bool*, and *match_all*. To manipulate the document's relevancy score, ES can provide the impression of precision and recall regardless of any retrieval strategy has been used for querying. With some DSLs, I can obtain the perfect recall. For example, *Fuzziness* is a DSL query type to indicate how many edits can be adapted to a keyword term, which I need to find its

match. Simulating that I input a keyword "month", the relevant documents should be returned. Then, with *Fuzziness*, another edited keyword is still valid for query the relevant documents, such as "mouth". This is obvious that recall probably increases, as opposed to precision. In the other hand, I admitted that, for information retrieval, the considerable performance score should be precision. There are several DSLs that helps to enhance the precision score, such as *bool*, *filters*, and *min_score*. For example, I use *min_score* to define a minimum score threshold to exclude documents that match less than the defined minimum value. By this query, I can only interest in the most relevant results from entire retrieved data; thus, the very high precision should be obtained. Based on the evidences above, ES is a high potential search engine software that provides many flexible query APIs. To state how the precision and recall are, it depends on diverse factors such as user strategies and techniques, a way to structure queries, and index definition.

Solr is another open-source search engine software which is also built on Lucene for indexing, searching and analyzing as same as ES. It also provides a fast response based on restful APIs that have a request method on HTTP. It contains many features, such as facets navigation, query language supports structured, and search results clustering. Both ES and Solr are proper solutions for general information retrieval needs, because they are mature, stable and having a great support from big community. However, some difference does exist. The first difference between both is about the distributions [63]. Solr consume more space in hard drive than ES around five times because Solr distribution includes optional functionality installed simultaneously with the base program, such as a testing framework and a monitoring tool. Meanwhile, ES installs only the base code, but users can manually add plugins. For example, in preprocessing, ES use another separate module, called logslash, to read data from a database; whilst, Solr integrates preprocessing, indexing, and searching modules together. The second difference is found in the cluster management. ES can form a cluster with only one node, which represents a single server, while Solr uses three nodes; thus, ES is a litter simpler than Solr to manage clustering. I regarded the performance between both systems and realized that they showed similar results on the dataset [3, 63], because they use Lucene for indexing

and searching. Therefore, in term of relevancy, users can choose either ES or Solr, but ES is a little better than Solr in term of speed and resource management.

Sphinx is an open-source search engine for searching in data from different sources, e.g. relational databases. It is a stand alone and light-weight program. Its indexing process is quite fast comparing to others because of directly connecting to database, but the index cannot be updated new documents after created. From this reason, the indexing process between ES and Sphinx seem to be different. Moreover, the indexing in ES is much flexible to deal with plural tables and can properly cope to multiple indexes easier than Sphinxs due to effective mapping process in ES. To analyze the speed of indexing and searching, search engine systems based on Lucene are faster than Sphinx [58]. However, regarding the relevance performance, I realized that ES should provide better results than Sphinx because ES offers several DSL that can advance precision and recall. Unlike Sphinx, the index can be constructed based on structured query language (SQL) query that may limit some functions, for example, SQL query cannot cope a fuzzy case, but ES can by using *Fuzziness*. As noted above, this *Fuzziness* improves the relevance score.

DB2 text search extender is a part of IBM database, named DB2. It is a commercial software from IBM. This part is an extension of database used for searching and indexing text. Based on information in official website [41], the indexing process is fast even on a large size of dataset, and multiple document formats have been supported. However, for its indexing process, it uses SQL command to create an index, which is similar to Sphinx; since, after I realized the limitation of SQL command, as presented above, this program may unmatched to this dissertation's purpose, because I attempted to benchmark several search engine systems to find which provides the best relevance score.

The state-of-art work was presented in [62]. Li et al. attempted to address a problem for user knowledge limited or do not know an actual keyword to search to their search engine; thus, a user possibly tried to input some keywords as correct as possible in order to obtain needed information. They proposed a powerful search engine composing incremental-search algorithms and type-ahead search. Their focus was to increase querying speed by using cache on client's side as well as to improve

recall by allowing to retrieve more results from one specific keywords, for example, the user inputs a person name "Jone" and retrieve some related results. With the type-ahead search technique, another name which is similar to the keyword also allow to be used "Jane". Unfortunately, its functionality is limited because this application may be difficult to deal with a complex index, and it is also uncomfortable to create a search engine based on this application.

I describe the background of selected search engines comprising ES, Solr, Sphinx, DB2 text search extender, and the state-of-art work from [62]. I chose them because these were effective, stable, great supports, and famous. They had been recommended from many communities. The one I selected to use for creating the common search engine compared to my ontology-based graph search engine is ES because ES takes an advance than others in viewpoint of indexing, searching, and speed as mentioned above. Table B.1 shows a summary of the classic search engines and their features.

Program name	Particular features
ES	<ul style="list-style-type: none"> • Indexing is based on Lucene. • It is fast and stable. • Restful API is available. • It can deal with a large size of data with a complex index. • It allows to create multiple indexes. • Search APIs is officially provided and easy to use. • Query DSLs is a main factor to enhance system performance. • It is an open source program.
Solr	<ul style="list-style-type: none"> • Indexing is based on Lucene. • It is fast and stable. • In a viewpoint of relevance, it is not greatly different to ES. • In a viewpoint of speed and management, it is great, but ES is slightly better. • Restful API is available. • It is an open source program.

Sphinx	<ul style="list-style-type: none">• It is stand alone and light-weight program.• It can deal with simple search perfectly.• It still need to be developed to cope a large data with a complex index.• For indexing and searching, it uses SQL command, and no particular API is available.• It is an open source program.
DB2 text search extender	<ul style="list-style-type: none">• It is a commercial software.• It is an extension of IBM database named DB2• It is fast and good to handle a large data size.• For indexing and searching, it uses SQL command, and no particular API is available.

The system from [62]	<ul style="list-style-type: none">• It is a new method to create an efficient search engine.• It consists of incremental-search algorithms and type-ahead search.• It can cope a case of fuzzy keyword.• It can search documents within milliseconds by using cache.• Further development is needed, particularly functionality.
----------------------	--

TABLE B.1: Summary of classic search engines and their features.

Bibliography

- [1] Aggarwal, C. C. & Yu, P. S. (2001). Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, (pp. 37–46). ACM.
- [2] Agrawal, S., Verma, N. K., Tamrakar, P., & Sircar, P. (2011). Content based color image classification using svm. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*, (pp. 1090–1094). IEEE.
- [3] AKCA, M. A., Aydođan, T., & İlkuçar, M. (2016). An analysis on the comparison of the performance and configuration features of big data tools solr and elasticsearch. *International Journal of Intelligent Systems and Applications in Engineering*, 74, 7.
- [4] Alvarez, J. M., Gevers, T., LeCun, Y., & Lopez, A. M. (2012). Road scene segmentation from a single image. In *European Conference on Computer Vision*, (pp. 376–389). Springer.
- [5] Anick, P. (2003). Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 88–95). ACM.
- [6] Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., & Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE journal of biomedical and health informatics*, 18(4), 1261–1271.
- [7] Antoniou, G. & Van Harmelen, F. (2004). *A semantic web primer*. MIT press.

- [8] Arivazhagan, S. & Ganesan, L. (2003). Texture classification using wavelet transform. *Pattern Recognition Letters*, 24(910), 1513 – 1521.
- [9] Arivazhagan, S., Ganesan, L., & Priyal, S. P. (2006). Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters*, 27(16), 1976 – 1982.
- [10] Avci, E. (2008). Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system. *Applied Soft Computing*, 8(1), 225–231.
- [11] Barla, A., Odone, F., & Verri, A. (2003). Histogram intersection kernel for image classification. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, (pp. III–513). IEEE.
- [12] Bassil, Y. & Alwani, M. (2012). Ocr post-processing error correction algorithm using google online spelling suggestion. *arXiv preprint arXiv:1204.0191*.
- [13] Batko, M., Falchi, F., Lucchese, C., Novak, D., Perego, R., Rabitti, F., Sedmidubsky, J., & Zezula, P. (2010). Building a web-scale image similarity search system. *Multimedia Tools and Applications*, 47(3), 599–629.
- [14] Both, A., Ngomo, A.-C. N., Usbeck, R., Lukovnikov, D., Lemke, C., & Speicher, M. (2014). A service-oriented search framework for full text, geospatial and semantic search. In *Proceedings of the 10th International Conference on Semantic Systems*, (pp. 65–72). ACM.
- [15] Brophy, J. & Bawden, D. (2005). Is google enough? comparison of an internet search engine with academic library resources. In *Aslib Proceedings*, volume 57, (pp. 498–512). Emerald Group Publishing Limited.
- [16] Brusa, G., Caliusco, M. L., & Chiotti, O. (2006). A process for building a domain ontology: an experience in developing a government budgetary ontology. In *Proceedings of the second Australasian workshop on Advances in ontologies-Volume 72*, (pp. 7–15). Australian Computer Society, Inc.

- [17] Cai, C., Wang, W., Sun, L., & Chen, Y. (2003). Protein function classification via support vector machine approach. *Mathematical Biosciences*, 185(2), 111 – 122.
- [18] Cao, T., Lahiri, I., Singh, V., Louis, J., Shah, J., & Ayre, B. G. (2013). Metabolic engineering of raffinose-family oligosaccharides in the phloem reveals alterations in carbon partitioning and enhances resistance to green peach aphid. *Frontiers in plant science*, 4, 263.
- [19] Chang, S.-F., Smith, J. R., Beigi, M., & Benitez, A. (1997). Visual information retrieval from large distributed online repositories. *Communications of the ACM*, 40(12), 63–71.
- [20] Chapelle, O., Haffner, P., & Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5), 1055–1064.
- [21] Chen, D., Odobez, J.-M., & Bourlard, H. (2004). Text detection and recognition in images and video frames. *Pattern recognition*, 37(3), 595–608.
- [22] Chen, Y., Sampathkumar, H., Luo, B., & Chen, X.-w. (2013). ilike: Bridging the semantic gap in vertical image search by integrating text and visual features. *IEEE Transactions on Knowledge and Data Engineering*, 25(10), 2257–2270.
- [23] Cheng, C., Koschan, A., Chen, C.-H., Page, D. L., & Abidi, M. A. (2012). Outdoor scene image segmentation based on background recognition and perceptual organization. *IEEE Transactions on Image Processing*, 21(3), 1007–1019.
- [24] Cheng, Y.-C. & Chen, S.-Y. (2003). Image classification using color, texture and regions. *Image and Vision Computing*, 21(9), 759 – 776.
- [25] Cusano, C., Ciocca, G., & Schettini, R. (2003). Image annotation using svm. In *Electronic Imaging 2004*, (pp. 330–338). International Society for Optics and Photonics.
- [26] Deserno, T. M., Antani, S., & Long, R. (2009). Ontology of gaps in content-based image retrieval. *Journal of digital imaging*, 22(2), 202–215.

- [27] Duda, R. O. & Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11–15.
- [28] Esmaelnejad, J., Habibi, J., & Yeganeh, S. H. (2010). A novel method to find appropriate ε for dbscan. In *Asian Conference on Intelligent Information and Database Systems*, (pp. 93–102). Springer.
- [29] Fan, Y., Shen, D., & Davatzikos, C. (2005). Classification of structural images via high-dimensional image warping, robust feature extraction, and svm. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005* (pp. 1–8). Springer.
- [30] Frate, F. D., Pacifici, F., Schiavon, G., & Solimini, C. (2007). Use of neural networks for automatic classification from high-resolution images. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(4), 800–809.
- [31] Gao, S. Y., Jack, M. M., & O'Neill, C. (2012). Towards optimising the production of and expression from polycistronic vectors in embryonic stem cells. *PloS one*, 7(11), e48668.
- [32] Gauch, S., Chaffee, J., & Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems: An international Journal*, 1(3, 4), 219–234.
- [33] Gruber, T. R. et al. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199–220.
- [34] Guarino, N. et al. (1998). Formal ontology and information systems. In *Proceedings of FOIS*, volume 98, (pp. 81–97).
- [35] Hare, J. S., Sinclair, P. A., Lewis, P. H., Martinez, K., Enser, P. G., & Sandom, C. J. (2006). Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches.
- [36] Hearst, M. A., Divoli, A., Guturu, H., Ksikes, A., Nakov, P., Wooldridge, M. A., & Ye, J. (2007). Biotext search engine: beyond abstract search. *Bioinformatics*, 23(16), 2196–2197.

- [37] Hiran, K. K. & Doshi, R. (2013). An artificial neural network approach for brain tumor detection using digital image segmentation. *Brain*, 2(5).
- [38] Huang, W., Tan, C. L., & Leow, W. K. (2005). Associating text and graphics for scientific chart understanding. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, (pp. 580–584). IEEE.
- [39] Hyvönen, E., Saarela, S., Styrman, A., & Viljanen, K. (2003). Ontology-based image retrieval. In *WWW (Posters)*.
- [40] Hyvönen, E., Saarela, S., & Viljanen, K. (2003). Ontogator: combining view- and ontology-based search with semantic browsing. *information retrieval*, 16, 17.
- [41] IBM (2016). Db2 text search.
- [42] Ibrahim, W. H., Osman, A. A. A., & Mohamed, Y. I. (2013). Mri brain image classification using neural networks. In *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, (pp. 253–258). IEEE.
- [43] Islam, M. R., Bulbul, F., & Shanta, S. S. (2012). Performance analysis of coiflet-type wavelets for a fingerprint image compression by using wavelet and wavelet packet transform. *International Journal of Computer Science and Engineering Survey*, 3(2), 79.
- [44] Jayalakshmi, T. & Chethana, C. (2016). A semantic search engine for indexing and retrieval of relevant text documents. *International Journal*, 4(5).
- [45] Jiang, L., Yu, S.-I., Meng, D., Mitamura, T., & Hauptmann, A. G. (2015). Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, (pp. 27–34). ACM.
- [46] Jobbins, A., Raza, G., Evett, L., & Sherkat, N. (1996). Postprocessing for ocr: Correcting errors using semantic relations. In *LEDAR. Language Engineering for Document Analysis and Recognition, AISB 1996 Workshop, Sussex, England*.

- [47] Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014). Convolutional neural networks for document image classification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, (pp. 3168–3172). IEEE.
- [48] Kanjanawattana, S. & Kimura, M. (2015a). Graph-type classification based on artificial neural networks and wavelet coefficients. In *Proceedings of Second International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC2015)*, (pp. 77–85).
- [49] Kanjanawattana, S. & Kimura, M. (2015b). A proposal for a method of graph ontology by automatically extracting relationships between captions and x- and y-axis titles. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, (pp. 231–238).
- [50] Kanjanawattana, S. & Kimura, M. (2016a). Extraction of graph information based on image contents and the use of ontology. In *International Conferences ITS, ICEduTech and STE 2016*, (pp. 19–26).
- [51] Kanjanawattana, S. & Kimura, M. (2016b). Ontologies-based optical character recognition-error correction method for bar graphs. In *The Tenth International Conference on Advances in Semantic Processing (SEMAPRO 2016)*, (pp. 1–8). IARIA.
- [52] Kanjanawattana, S. & Kimura, M. (2017). Extraction and identification of bar graph components by automatic epsilon estimation. *International Journal of Computer Theory and Engineering*, 9(4).
- [53] Karami, A. & Johansson, R. (2014). Choosing dbscan parameters automatically using differential evolution. *International Journal of Computer Applications*, 91(7).
- [54] Kataria, S., Browner, W., Mitra, P., & Giles, C. L. (2008). Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *AAAI*, volume 8, (pp. 1169–1174).

- [55] Kim, Y.-H., Yoon, D.-W., Kim, J.-H., Lee, J.-H., & Lim, C.-H. (2014). Effect of remote ischemic post-conditioning on systemic inflammatory response and survival rate in lipopolysaccharide-induced systemic inflammation model. *Journal of Inflammation*, 11(1), 1.
- [56] Klöppel, S., Stonnington, C. M., Chu, C., Draganski, B., Scahill, R. I., Rohrer, J. D., Fox, N. C., Jack, C. R., Ashburner, J., & Frackowiak, R. S. J. (2008). Automatic classification of mr scans in alzheimer's disease. *Brain*, 131(3), 681–689.
- [57] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc.
- [58] Kumar, J. (2016). Benchmarking results of mysql, lucene and sphinx.
- [59] Kwon, Y. H. et al. (1994). Age classification from facial images. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, (pp. 762–767). IEEE.
- [60] Lasko, T. A. & Hauser, S. E. (2000). Approximate string matching algorithms for limited-vocabulary ocr output correction. In *Photonics West 2001-Electronic Imaging*, (pp. 232–240). International Society for Optics and Photonics.
- [61] Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98–113.
- [62] Li, G., Ji, S., Li, C., & Feng, J. (2011). Efficient fuzzy full-text type-ahead search. *The VLDB JournalThe International Journal on Very Large Data Bases*, 20(4), 617–640.
- [63] Luburić, N. & Ivanović, D. (2016). Comparing apache solr and elasticsearch search servers. *6th International Conference on Information Society and Technology ICIST 2016*.

- [64] Ma, J.-q. (2009). Content-based image retrieval with hsv color space and texture features. In *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*, (pp. 61–63). IEEE.
- [65] McDonald, R., Pereira, F., Ribarov, K., & Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (pp. 523–530). Association for Computational Linguistics.
- [66] Mehrotra, R., Namuduri, K., & Ranganathan, N. (1992). Gabor filter-based edge detection. *Pattern Recognition*, 25(12), 1479 – 1494.
- [67] Mezaris, V., Kompatsiaris, I., & Strintzis, M. G. (2003). An ontology approach to object-based image retrieval. In *Image Processing, 2003. IICIP 2003. Proceedings. 2003 International Conference on*, volume 2, (pp. II–511). IEEE.
- [68] Müller, H.-M., Kenny, E. E., & Sternberg, P. W. (2004). Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol*, 2(11), e309.
- [69] Nagata, M. (1998). Japanese ocr error correction using character shape similarity and statistical language model. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, (pp. 922–928). Association for Computational Linguistics.
- [70] Nekooeian, A. A., Eftekhari, M. H., Adibi, S., & Rajaeifard, A. (2014). Effects of pomegranate seed oil on insulin release in rats with type 2 diabetes. *Iranian journal of medical sciences*, 39(2), 130–135.
- [71] Niles, I. & Pease, A. (2003). Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In *PROCEEDINGS OF THE 2003 INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE ENGINEERING (IKE 03), LAS VEGAS*. Citeseer.
- [72] Park, S. B., Lee, J. W., & Kim, S. K. (2004). Content-based image classification using a neural network. *Pattern Recognition Letters*, 25(3), 287 – 300.

- [73] PyWavelets discussion group (2008). Wavelet properties browser. Accessed 12 February 2016.
- [74] Salatino, A. A. (2014). Grid search svm. Accessed 3 February 2016.
- [75] Sánchez, J. & Perronnin, F. (2011). High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, (pp. 1665–1672). IEEE.
- [76] Sarlashkar, M. N., Bodruzzaman, M., & Malkani, M. (1998). Feature extraction using wavelet transform for neural network based image classification. In *System Theory, 1998. Proceedings of the Thirtieth Southeastern Symposium on*, (pp. 412–416). IEEE.
- [77] Sergyán, S. (2008). Color histogram features based image classification in content-based image retrieval systems. In *Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium on*, (pp. 221–224). IEEE.
- [78] Sharma, H. & Jansen, B. J. (2005). Automated evaluation of search engine performance via implicit user feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 649–650). ACM.
- [79] Sinha, N., Mishra, T., Singh, T., & Gupta, N. (2012). Effect of iron deficiency anemia on hemoglobin a1c levels. *Annals of laboratory medicine*, 32(1), 17–22.
- [80] Smith, L. (2016). Elasticsearch query-time strategies and techniques for relevance: Part i.
- [81] Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE transactions on knowledge and data engineering*, 25(1), 1–14.
- [82] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.

- [83] Strigl, D., Kofler, K., & Podlipnig, S. (2010). Performance and scalability of gpu-based convolutional neural networks. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, (pp. 317–324). IEEE.
- [84] Tong, X. & Evans, D. A. (1996). A statistical approach to automatic ocr error correction in context. In *Proceedings of the fourth workshop on very large corpora*, (pp. 88–100).
- [85] Tsuruoka, Y., Tsujii, J., & Ananiadou, S. (2008). Facta: a text search engine for finding associated biomedical concepts. *Bioinformatics*, *24*(21), 2559–2560.
- [86] Vailaya, A., Figueiredo, M., Jain, A., & Zhang, H. (1999). Content-based hierarchical classification of vacation images. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*, volume 1, (pp. 518–523). IEEE.
- [87] Veluchamy, M., Perumal, K., & Ponuchamy, T. (2012). Feature extraction and classification of blood cells using artificial neural network. *American Journal of Applied Sciences*, *9*(5), 615.
- [88] Wang, C., Zhang, L., & Zhang, H.-J. (2008). Learning to reduce the semantic gap in web image retrieval and annotation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 355–362). ACM.
- [89] Wick, M., Ross, M., & Learned-Miller, E. (2007). Context-sensitive error correction: Using topic models to improve ocr. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, (pp. 1168–1172). IEEE.
- [90] Xia, J., Du, P., He, X., & Chanussot, J. (2014). Hyperspectral remote sensing image classification based on rotation forest. *IEEE Geoscience and Remote Sensing Letters*, *11*(1), 239–243.

- [91] Ye, Q., Gao, W., & Zeng, W. (2003). Color image segmentation using density-based clustering. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, (pp. III–345). IEEE.
- [92] Yu, J., Tao, D., Wang, M., & Rui, Y. (2015). Learning to rank using user clicks and visual features for image retrieval. *IEEE transactions on cybernetics*, 45(4), 767–779.
- [93] Zhao, R. & Grosky, W. I. (2002). Narrowing the semantic gap-improved text-based web document retrieval using visual features. *IEEE transactions on multimedia*, 4(2), 189–200.
- [94] Zhong, H. & Xia, L.-M. (2007). Ontology-based image retrieval. *Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications)*, 42(17), 37–40.
- [95] Zhuang, L. & Zhu, X. (2005). An ocr post-processing approach based on multi-knowledge. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, (pp. 346–352). Springer.