論文 / 著書情報
Article / Book Information

題目(和文)	ファジィ関係の可視化およびそのビッグデータ論文検索システムへの応用
Title(English)	Fuzzy Relationship Visualization and its Application to Bibliographic Big Data Retrieval System
著者(和文)	マスリナ ビンティ ゾルケプリ
Author(English)	Maslina Binti Zolkepli
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第9876号, 授与年月日:2015年3月26日, 学位の種別:課程博士, 審査員:廣田　薫,寺野　隆雄,柴田　崇徳,小野　功,董　芳艶
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9876号, Conferred date:2015/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Fuzzy Relationship Visualization and its Application to Bibliographic Big Data Retrieval System
(ファジィ関係の可視化およびそのビッグデータ論文検索システムへの応用)

Maslina Binti Zolkepli
マスリナ ビンティ ゾルケプリ

Academic Advisor : Prof. Kaoru Hirota

Doctoral Thesis

Tokyo Institute of Technology
東京工業大学
Interdisciplinary Graduate School of Science and Engineering
大学院総合理工学研究科
Department of Computational Intelligence and Systems science
知能システム科学専攻

ABSTRACT

A bibliographic data visualization method is proposed by incorporating i) combination of Newman-Girvan algorithm and fuzzy c-means to find fuzzy relationship among bibliographic data, ii) automatic switch to compare algorithms' performance, and iii) fuzzy ontology framework to find overlapping keywords.

A combination of Newman-Girvan clustering method and the self-adapted fuzzy c-means algorithm is utilized to find fuzzy relationship among the bibliographic big data. The combination of a crisp and a fuzzy clustering method is able to generated several highly related bibliographic data from more than 1.5 million data available in the database in less than 5 minutes. A fuzzy visualization of the clustered result is displayed in a network view by grouping objects with similar cluster membership. Users are able to interactively click on each data point to find more information about each data point. As current bibliographic visualization methods focus on crisp relationship among its data, fuzzy analysis and visualization may offer deeper insights to the bibliographic big data, leading to faster decision making by improving the precision of the displayed information.

To compare the individual Newman-Girvan algorithm and the self-adapted fuzzy c-means with their combination, an automatic switch between ensembles of clustering algorithms is proposed by utilizing a fuzzy inference engine as a decision support tool to select the fastest performing clustering algorithm between fuzzy c-means clustering, Newman-Girvan clustering, and the combination of both. It aims to realize the best clustering performance with

the reduction of computational complexity from $O(n^3)$ to O(n). The automatic switch is developed a fuzzy logic controller written in Java and the experimental results demonstrates that the combination of both clustering algorithms is selected as the best performing algorithm in 20 out of 27 cases with the highest percentage of 83.99%, completed in 161 seconds.

As for the retrieval method, a fuzzy ontology based knowledge reasoning framework that combines the fuzzy logic and descriptive logic is proposed to represent overlapping and imprecise keywords in bibliographic big data retrieval. Fuzzy ontologies for bibliographic big data is created using fuzzy ontology language and it aims to increase precision of retrieval result before the clustering process.

The combination of clustering algorithms, the automatic switch, and the fuzzy ontology based knowledge reasoning framework is to be incorporated into the Bibliographic Big Data Retrieval System that focuses on visualization of fuzzy relationship, planning to be released to the public through the Internet.

ACKNOWLEDGEMENT

I would like to convey my sincere appreciation and deepest gratitude to my academic advisor, Professor Kaoru Hirota for his valuable guidance and advice. It is for his patience, understanding, critical guidance and wise counsel that made the completion of this study possible.

Not to forget my co-academic advisor, Associate Professor Fangyan Dong for her help and guidance in many ways that I cannot explain. Sincere appreciation is also due to Ms. Harumi Hoshino for taking care of my academic needs in one way or another.

A very big thank you is also expressed to all of my labmates for their time and advice, and also for participating in part of my research.

Last but not least, a sincere appreciation for my family for their understanding and the sacrifice throughout the duration of my study. Not forgetting my parents for their endless support, encouragement, and love. Thank you all for your love and patience

TABLE OF CONTENT

ABSTRACT **i**

ACKNOWLEDGEMENT **iii**

TABLE OF CONTENT **iv**

CHAPTER 1 1

Introduction **1**

CHAPTER 2 9

Fuzzy Relationship Visualization for Bibliographic Big Data using Fuzzy C-Means and Newman-Girvan Algorithm **9**

2.1 Introduction 9

2.2 Hybrid Approach of Self-Adapted Fuzzy C-Means Clustering and Newman-Girvan Clustering Algorithm 11

2.3 Bibliographic Data DBLP 17

2.4 Fuzzy Visualization using Java Universal Network/Graph 21

2.5 Experiment on Journal Papers Retrieval from DBLP 24

2.6 Chapter Summary 44

CHAPTER 3 47

Automatic Switching of Clustering Methods based on Fuzzy Inference in Bibliographic Big Data Retrieval System **47**

3.1 Introduction 47

3.2 Clustering result of the self-adapted fuzzy c-means, Newman-Girvan algorithm, and ther combination 49

3.3 Automatic Switch between 3 clustering algorithms based on Fuzzy Inference 54

3.4 Experiment of Automatic Switching on Clustering Results 64

3.5 User-based Evaluation for the Bibliographic Big Data Retrieval System 68

3.6 Chapter Summary 75

CHAPTER 4 77

Fuzzy Ontological Approach in Keyword-Based Retrieval for Bibliographic Big Data Retrieval System 77

4.1 Introduction 77

4.2 Fuzzy Ontology for Bibliographic Big Data 80

4.3 A Semantic Tool of Keyword-Based Retrieval for Bibliographic Big Data Retrieval System 81

4.4 Fuzzy Ontology Knowledge Reasoning Framework 83

4.5 Experiment of Fuzzy Ontologies from User Queries on Bibliographic Big Data 87

4.6 Chapter Summary 90

CHAPTER 5 91

Conclusion 91

BIBLIOGRAPHY 94

RELATED PUBLICATIONS 100

LIST OF TABLES

Table 2.1 Experiment result for self-adapted fuzzy c-means clustering, the Newman-Girvan clustering algorithm and the proposed method on author search keyword “Andreas Neumann”. 30

Table 2.2 Experiment result for self-adapted fuzzy c-means clustering algorithm 31

Table 2.3 Experiment result for Newman-Girvan clustering algorithm 32

Table 2.4 Experiment result for combination of both clustering algorithms 32

Table 2.5 The response time comparison for the three clustering algorithms 33

Table 2.6 Precision, recall, and f-measure comparison between 3 clustering algorithms 34

Table 3.1 Clustering Result From Bibliographic Big Data Retrieval System 51

Table 3.2 AutomaticSwitch Result – Percentage Output 64

Table 3.3 Answer Option For Feedback Questionnaire And Corresponding Merits 71

Table 3.4 Usability points and corresponding usability levels 71

Table 3.5 System evaluation result by participants 72

Table 3.6 Usability level and usability points for each category 73

Table 4.1 Experimental result for fuzzy ontology based retrieval 88

LIST OF FIGURES

Figure 1.1 IVC Screenshot 3

Figure 1.2 MetaNetViz Screenshot 4

Figure 1.3 BibRelEx Screenshots 4

Figure 1.4 Citewiz screenshot 5

Figure 1.5 Biblioviz screenshot 5

Figure 1.6 Outline of thesis 8

Figure 2.1 Method to calculate vector centers in the fuzzy c-means clustering method. 13

Figure 2.2 Bibliographic big data visualization method architecture. 16

Figure 2.3 Raw form of DBLP dataset prepared by ArnetMiner. 18

Figure 2.4 Dataset format for fuzzy c-means clustering. 19

Figure 2.5 Dataset sample for fuzzy c-means clustering. 19

Figure 2.6 Dataset for Newman-Girvan clustering without weight information. 19

Figure 2.7 Dataset for Newman-Girvan algorithm with membership degrees as weight for each data point. 20

Figure 2.8 Example of JUNG visualization 22

Figure 2.9 Bibliographic big data search method's user interface 26

Figure 2.10 Dataset for the self-adapted fuzzy c-means clustering based on the search keyword "Andreas Neumann". 27

Figure 2.11 Membership matrix result from the fuzzy c-means method. 28

Figure 2.12 Newman-Girvan dataset with weight from self-adapted fuzzy c-means membership degrees. 28

Figure 2.13 Comparison chart of 3 clustering algorithms' performance. 34

Figure 2.14 Crisp relationship of the dataset based on the author keyword “Andreas Neumann”. 37
Figure 2.15 Visualization of Newman-Girvan clustering result for keyword “Andreas Neumann” 38
Figure 2.16 Visualization of self-adapted fuzzy c-means clustering result for keyword “Andreas Neumann” 39
Figure 2.17 Visualization of combination algorithm for keyword “Andreas Neumann" 41
Figure 2.18 Newman-Girvan clustering result for author search #2: “Edward Omiecinski”. 41
Figure 2.19 Self-adapted fuzzy c-means clustering result for author search #2: “Edward Omiecinski”. 42
Figure 2.20 Combination clustering result for author search #2: “Edward Omiecinski”. 42
Figure 2.21 Newman-Girvan clustering result for author search #3: “William Kent”. 43
Figure 2.22 Self-adapted fuzzy c-means clustering result for author search #3: “William Kent”. 43
Figure 2.23 Combination clustering result for author search #3: “William Kent”. 44
Figure 3.1 Bibliographic Big Data Retrieval System’s user interface 53
Figure 3.2 The automatic switch layout 55
Figure 3.3 Bibliographic Big Data Retrieval System architecture with automatic switch function. 56
Figure 3.4 Fuzzification of total number of clusters 57
Figure 3.5 Membership degrees graph for total number of clusters 57
Figure 3.6 Fuzzification of total number of cluster 58
Figure 3.7 Membership degrees graph for total number of vertices 58
Figure 3.8 Fuzzification of time in seconds 59

Figure 3.9 Membership degrees graph for time in seconds 59

Figure 3.10 Defuzzification for percentage output and defuzzification method specification 60

Figure 3.11 Membership degrees graph for percentage 60

Figure 3.12 Fuzzy associative matrix for percentage control 61

Figure 3.13 Rule block for inference rules of the automatic switch 63

Figure 3.14 Bar chart for percentage output result of the fuzzy inference 67

Figure 3.15 User feedback questionnaire 70

Figure 3.16 Bar chart for User Satisfaction Level 73

Figure 4.1 Computer science domain for fuzzy ontology 80

Figure 4.2 Screenshot of entities in Protégé 81

Figure 4.3 OWL 2 Ontology concept framework 82

Figure 4.4 Fuzzy ontology knowledge reasoning framework 84

Figure 4.5 Bibliographic Big Data Retrieval System architecture 85

Figure 4.6 Bibliographic Big Data Retrieval System prototype with Domain selection and Importance indicator 86

Chapter 1

Introduction

Bibliographic big data is data about references to published literature, including journal conference proceedings and books. It may be general in scope or cover a specific academic discipline. Bibliographic data usually includes titles, names, subjects, date, and information about the physical description of the published literature.

Since the first research paper was published in 1665, the academic world has seen more than 1.4 million research papers published in more than 23,000 journals by more than 2,000 publishers.

The large volume of research paper available today has created a demand for an effective retrieval method that can help research experts to get their desired papers in the fastest time. Due to that massive amount of literature, researchers have difficulties in identifying correct search terms[1]. Searches are often unsuccessful and the researchers need to spend a long time to get the materials they really needed. In academic fields, a practical tips on how to write a good quality paper is to find out what's hot, and find the trends of the subject area, such as publication venues of journals and conferences, authors and their publications per year[2].

Computers were able to make text digital since the 1960s. It was done to reduce the cost to publish several American journals. In late 1960s, digitized texts, known as bibliographic databases, becomes a new form of information resource[3].

Several databases of academic literature were first offered in 1970s where the public was able to perform online information retrieval over private networks. These databases contained bibliographic data about journal articles that were retrievable using keywords in author and title, and sometimes by journal name or subject heading. Unfortunately, the graphical user interface were rigid, difficult to access and the retrieval process was only accessible to expert users like librarians instead by general public[4].

Due to the demand to search for bibliographic data effectively, bibliographic visualization tool have been introduced to search and visualize connections between literatures, authors and researchers. The tool provides the ability do display connection between bibliographic entries, display complete bibliographic entries, categorize scientific papers by title, author, and keyword, display the chronological details and inspiration of papers, and provide visualizations of huge set of search result.

It is very useful in helping users like students, scientists, and researchers to find their desired scientific papers to conduct their research work. The tool is especially helpful in guiding users when the research domain is unfamiliar. These users are experiencing difficulties in collecting relevant documentation in their field of study because searching for scientific papers using the web can consume a lot of time.

The main issue in conducting an efficient research is how to control the quality of information that reaches our attention. A standard database cannot effectively generate and manipulate bibliographic data because it is characterized by strongly networked connection. A bibliographic visualization tool is able to visualize this networked connection effectively by using network graph.

There are several bibliographic visualization tools available to the public now. These tools vary in terms of capabilities, structures, and availability.

Infovis CyberInfrastructure(IVC)[5] offers visualization algorithms such as JUNG GraphML and Prefuse XML Graph to build highly interactive visualizations of structured and unstructured data. Data can be represented as a set of entities or nodes, possibly connected by any number of relation or edges. For example, hierarchical data, network data, and non-connected collections of data. Prefuse is designed to visualize interrelated information so it can be stored in a graph of a tree structure. Non-related data can also be visualized by storing it within a data table.

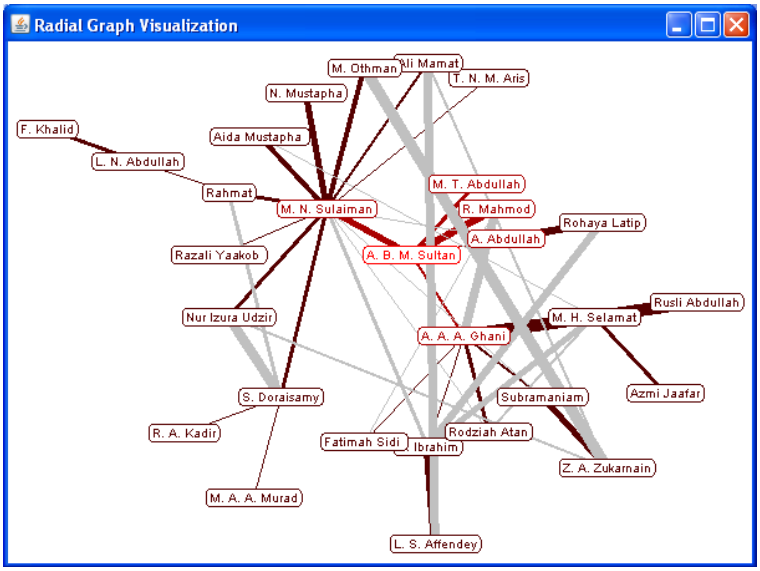


Figure 1.1 IVC Screenshot

Metadata Network Visualizer or MetaNetViz[6] is an application to automatically extract publication metadata from text document and visualize the relationship among the metadata. It visualizes the connection among the metadata through graph drawing representation of interconnected metadata. The interface represents the metadata as connected nodes in a graph.

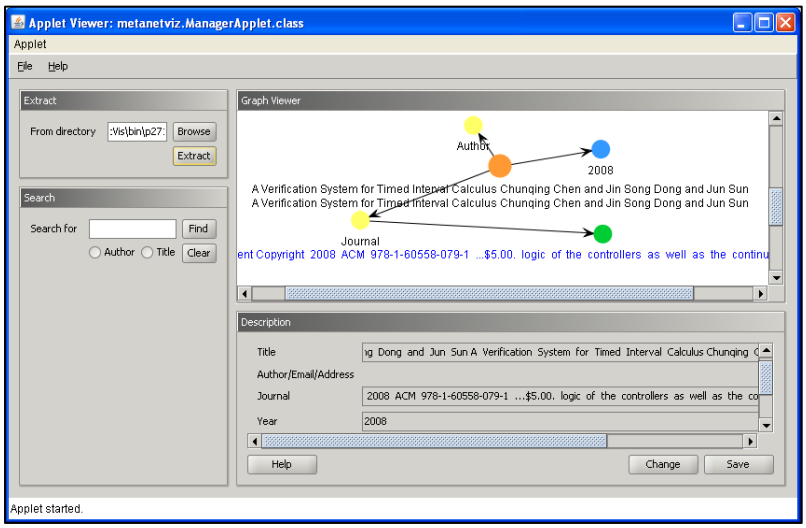


Figure 1.2 MetaNetViz Screenshot

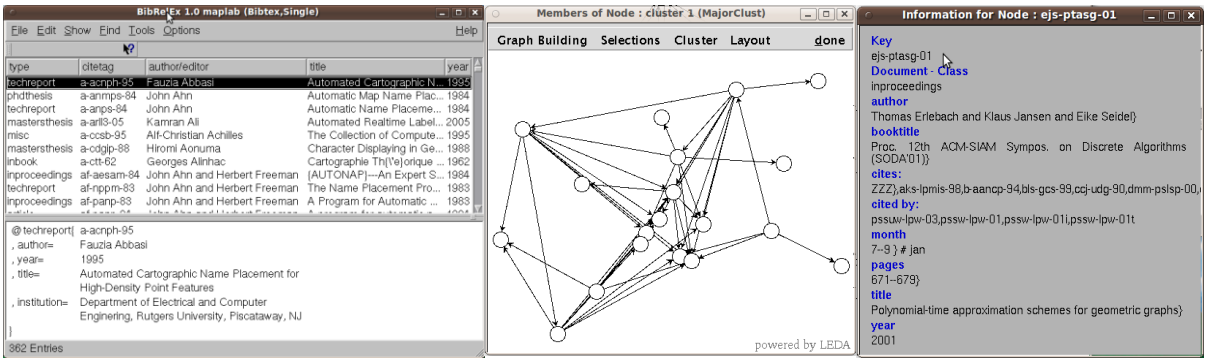


Figure 1.3 BibRelEx Screenshots

BibRelEx[7] combines expert knowledge on a scientific literature and makes it available to researchers who wish to explore the literature. To visualize the bibliographic information, BibRelEx collects expert annotations on publications and their semantic relationships to other publications. Then it lets researchers explore this semantic literature and knowledge through visualization that enables researchers to track relevant documents based on their colleagues' expertise

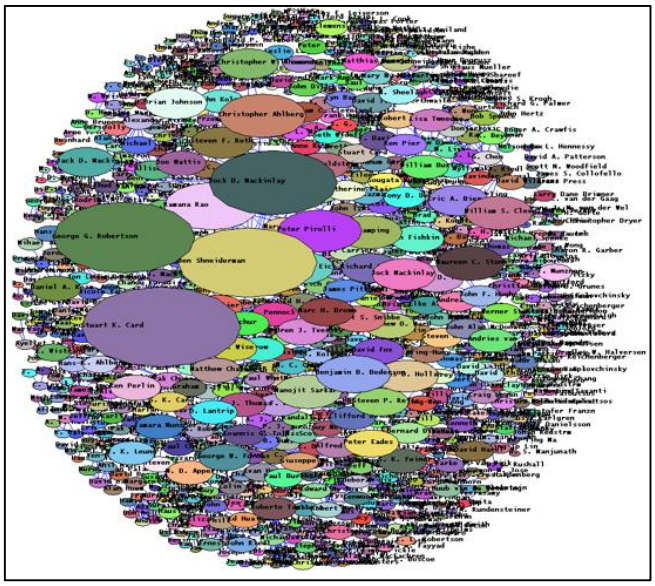


Figure 1.4 Citewiz screenshot

Citewiz[8] is a bibliographic visualization tool that visualizes the chronology and influences in networks of scientific articles. It offers three types of visualizations: a timeline visualization for overviews and navigation in a full citation database, an influence visualization for detailed views of a specific subset of the citation database, and an interactive concept map for exploring keywords and co-authorship in the database. Citewiz is based on a taxonomy of the usage of citation databases.

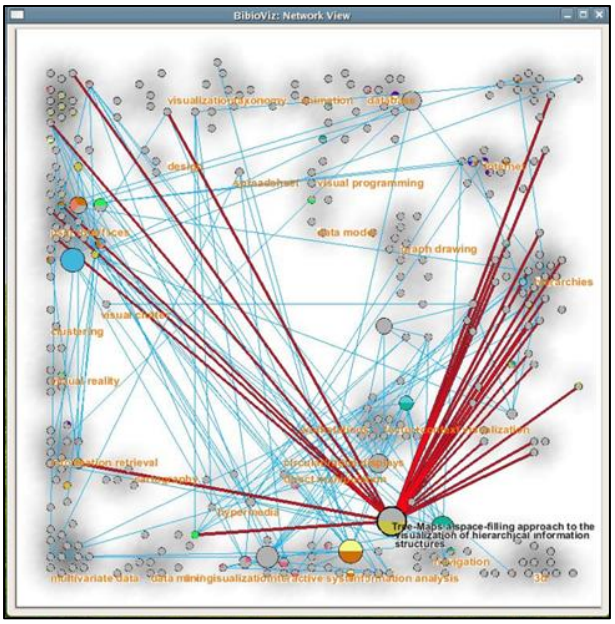


Figure 1.5 Biblioviz screenshot

Another bibliographic visualization tool, Biblioviz[9] is inspired by the InfoVis 2004 contest where many tools that were presented during the contest offers some unique views of the bibliography data, but there is no single best system offering all the desired views. The idea to combine several desired functionalities was realized by developing Biblioviz that gives the maximum number of views of the data using a minimum number of visualization constructs in a unified fashion.

Based on the existing bibliographic visualization tools that have been described above, it can be seen that different visualization tools are developed for different types of users. To date, there are no bibliographic visualization tools that focuses on the fuzzy relationship among its data.

In Chapter 2, the combination of Newman-Girvan algorithm and the self-adapted fuzzy c-means are introduced as the clustering mesh of the proposed method in 2.2 . The DBLP Citation Network dataset used in the proposed method is explained in 2.3. JUNG visualization algorithm is described as the visualization method in 2.4, and the clustering and visualization result are presented and discussed explicitly in 2.5.

In Chapter 3, an automatic switch is proposed to select among the Newman-Girvan algorithm, the self-adapted fuzzy c-means and the combination of both algorithms. The dataset used for the automatic switch is described in 3.2 . The fuzzy inference engine developed as the automatic switch is described in 3.3. The automatic switch result and user feedback evaluation is presented in 3.4.

In Chapter 4, a fuzzy ontological approach in keyword-based retrieval is proposed. A fuzzy ontology for bibliographic data in computer science domain is explained in 4.2. A

semantic tool of keyword-based retrieval is described in 4.3. and the proposed fuzzy ontology knowledge reasoning framework is described in 4.4. Experiment result of fuzzy ontologies from user queries is presented and discussed in 4.5.

In Chapter 5, the proposals and related works in the thesis are summarized, in addition, new extensions and applications are discussed for the future work. The road map visualizes the dependence among the individual chapters, and summarizes the thesis organization as shown in Figure 1.1.

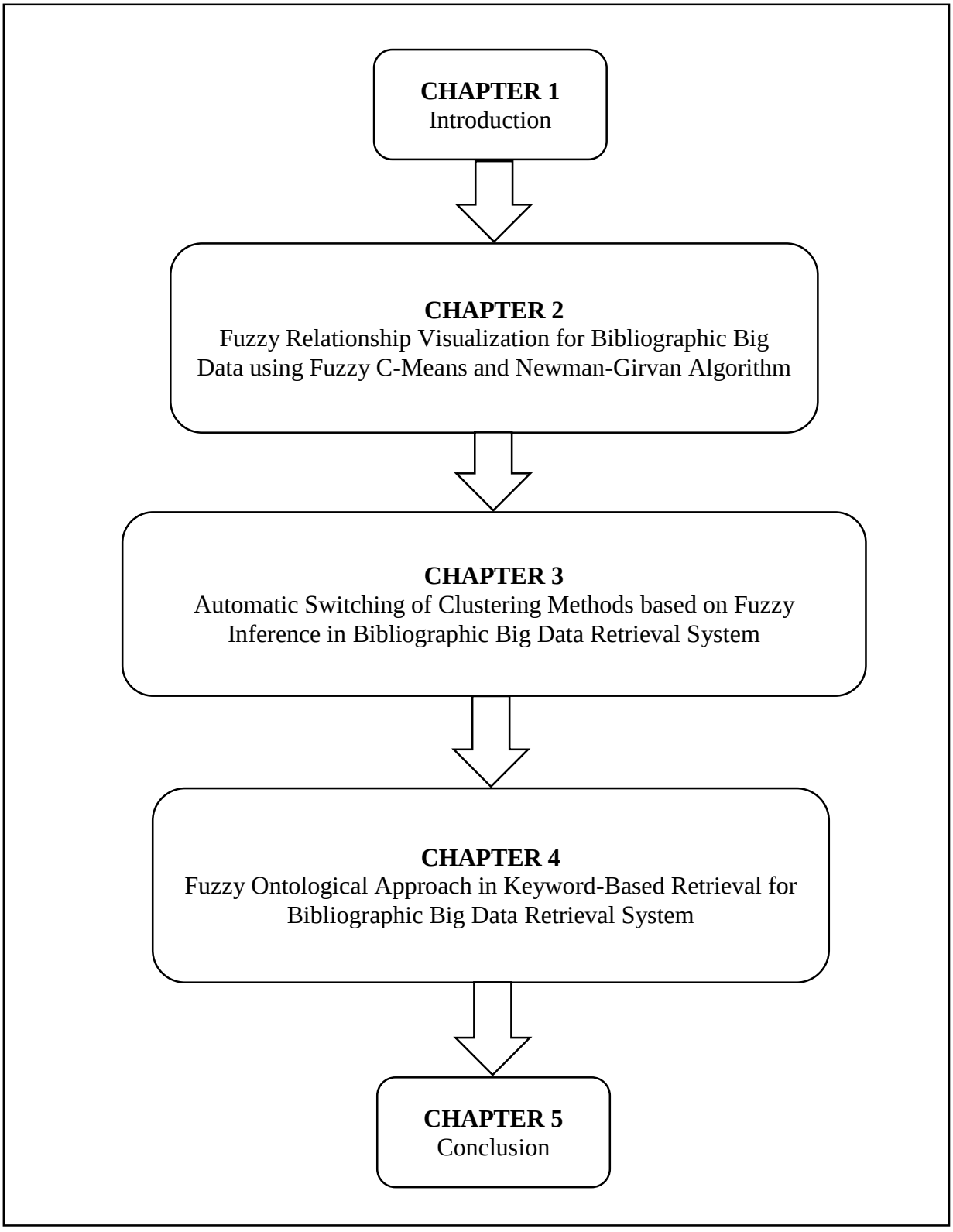


Figure 1.6 Outline of thesis

Chapter 2

Fuzzy Relationship Visualization for Bibliographic Big Data using Fuzzy C-Means and Newman-Girvan Algorithm

2.1 Introduction

Bibliographic visualization method is a method developed to visually capture the relationship among bibliographic big data consisting of journal/conference papers especially in the field of computer science. The method is accepted by users like researchers, educators, and learners to find appropriate scientific papers. Conventional bibliographic visualization methods include Infovis CyberInfrastructure(IVC)[5], MetaNetViz[6], BibRelEx[7], CiteWiz[8], and Biblioviz[9], where their main concern is to visually convey crisp relationships among bibliographic big data in various ways without paying much attention to the fuzzy relationship among the big data. Therefore, by introducing fuzzy analysis and visualization, it may offer deeper insights into the big data.

There are several hybrid fuzzy c-means clustering methods that have been introduced in recent times, such as the fuzzy c-means hybrid approach to the clustering of supply chain[10]. It integrates fuzzy c-means, genetic algorithms and tabu search to determine the optimal clustering parameters that individual fuzzy c-means cannot produce. Another hybrid fuzzy clustering algorithm that combines the fuzzy c-means and Multivariate Adaptive Regression

Splines(MARS)[11] is introduced in bankruptcy forecasting. The clusters are created using fuzzy c-means and classified into two groups. A MARS model is then created using the data from the two groups as part of the input information. Basically many hybrid approach of combining fuzzy c-means with other methods aims to overcome fuzzy c-means limitations and enhance the advantages of the integrated methods.

By incorporating a hybrid combination of self-adapted fuzzy c-means clustering[12] and the Newman-Girvan clustering algorithm[13], a method is presented to search for relationships in bibliographic big data by applying fuzzy concept. It accepts results from self-adapted fuzzy c-means clustering as part of input information for the Newman-Girvan clustering algorithm to produce a more in-depth result to provide quantitative information on how much a data belong each cluster.

The proposed method uses visualization techniques where the membership value of each dataset is displayed in a manner that retains the fuzziness to prevent loss of useful information. The visualization techniques provide an interactive network view by grouping objects with similar cluster membership, shows connections between objects in each cluster and the strength of relationship between each object by applying fuzzy concept. The proposed method is able to increase the level of details per retrieved result that conventional methods[7,8,9] without fuzzy logic do not focus on positively. It aims to assist users to make faster decisions by increasing the precision of the information displayed. The level of detail in visualization is critical to the users because they either requires information on specific items or just need to view the general characteristics of their search[14].

The dataset used in the proposed method is the DBLP Computer Science Bibliography citation network dataset, available at http://arnetminer.org/DBLP_Citation [15,16]. The two

clustering algorithms, i.e., the self-adapted fuzzy c-means and Newman-Girvan algorithm, on the dataset are performed in Java. The proposed method implements Java Universal Network/Graph Framework[17] for interactive visualization to provide functions that enable users to explore and manipulate the search result.

Two target clustering algorithms are presented in 2.2. The dataset DBLP is mentioned in 2.3., and the visualization method is proposed in 2.4.. The experiment results are shown in 2.5.

2.2 Hybrid Approach of Self-Adapted Fuzzy C-Means Clustering and Newman-Girvan Clustering Algorithm

There are a number of clustering algorithms suitable for bibliographic big data. In the proposed method, two clustering algorithms are chosen for clustering purposes, namely the self-adapted fuzzy c-means clustering and the Newman-Girvan clustering algorithms.

2.2.1 Self-adapted Fuzzy C-Means Clustering

Fuzzy c-means algorithm is a powerful unsupervised method for data clustering[18]. In fuzzy c-means, data points on the boundaries between several clusters are not forced to fully belong to one cluster. They assigned membership degrees between 0 and 1 to indicate their partial membership of each cluster. The membership degree is assigned to each data point corresponding to each cluster center on the basis of distance between the cluster and the data point. The closer the data to the cluster center, the higher membership degree it has towards the particular cluster center. Fuzzy c-means clustering is able to give results for overlapped dataset,

a feature that other crisp clustering method is not able to do. For example, in hard k-means, each data point must exclusively belong to one cluster center. In fuzzy c-means however, as each data point is assigned a membership degree to each cluster center, it is able to belong to more than one cluster center. Hence giving a more precise result than k-means clustering algorithm.

Compared with other fuzzy clustering methods such as Gustafsone-Kessel[19] and Gath-Geva[20] algorithms, fuzzy c-means performs better by creating better-separated and meaningful clusters with high compactness[21].

A big advantage that fuzzy c-means clustering has is that it does not decide the absolute membership of a data point to a given cluster. Since absolute membership is not calculated, it can also be extremely fast as the number of iterations required to achieve a specific clustering exercise corresponds to the required accuracy.

In the proposed method, the membership degrees information of each data point resulting from the fuzzy c-means clustering is suitable to be used as a weighing factor when it is combined with the next clustering method, the Newman-Girvan clustering algorithm. The sample of the programming code is shown in Figure 2.1.

One issue with fuzzy c-means is an apriori specification of the number of clusters needs to be determined. To overcome this issue, a self-adapted fuzzy c-means is introduced recently in which a new validity function is introduced where the inter-cluster distances should be as bigger as possible and the intra-cluster distances should be as smaller as possible.

```
public void calculate_centre_vectors(){
 System.out.println("calculate_centre_vectors()");
 int i,j,k;
 double numerator,denominator;
```

```
 double t[][] = new double[MAX_DATA_POINTS][MAX_CLUSTER];

 for (i=0; i<num_data_points;i++){
  for(j=0;j<num_clusters; j++){
   t[i][j]=Math.pow(degree_of_membership[i][j], fuzziness);
   }
  }

 for(j=0;j<num_clusters;j++){
  for(k=0;k<num_dimensions;k++){
   numerator=0.0;
   denominator=0.0;

   for(i=0;i<num_data_points;i++){
    numerator += t[i][j]*data_point[i][k];
    denominator += t[i][j];
   }
   cluster_centre[j][k] = numerator/denominator;
  }
 }
}
```

Figure 2.1 Method to calculate vector centers in the fuzzy c-means clustering method.

The advantage of the self-adapted feature is the initial number of clusters does not need to be determined prior to the clustering process. This feature is highly appropriate for the proposed method as the number of clusters generated by the clustering algorithm will be the stop criterion for edge removal process in the Newman-Girvan clustering algorithm.

2.2.2 Newman-Girvan Clustering Algorithm

The second algorithm used in the clustering combination is the Newman-Girvan clustering algorithm. It is hailed as an algorithm that marked the beginning of a new era in community detection field[22]. It works by selecting edges according to the values of measures of edge centrality and removes the edges with the highest betweenness, splitting the whole network into isolated sub-graphs until the network is broken into isolated single nodes. The steps of the algorithm are:

Step 1. Compute the centrality for all edges.

Step 2. Remove the edge with largest centrality. In case of ties with other edges, one is picked randomly.

Step 3. Recalculate the centrality on the running graph.

Step 4. Iteration of the cycle from step 2 until desired graphs are produced.

Newman-Girvan clustering predominates other clustering methods as numerical studies have shown that the recalculation phase in Step 3 of the algorithm is important to detect meaningful communities[23].

There have been countless applications of the Newman-Girvan algorithm and it is now integrated in many network analysis programs such as Java Universal Network/Graph Framework at http://jung.sourceforge.net/ and igraph at http://igraph.org/. The algorithm is chosen for the proposed method as it has succeeded in many applications in social and biological networks[24]. One issue with Newman-Girvan algorithm is how to decide when to stop removing edges when a suitable number of clusters have been found. To compensate this, the self-adapted fuzzy c-means is able to find the sum of clusters voluntarily, thus giving this sum information to the Newman-Girvan clustering method as the stopping criterion for the iteration process in Newman-Girvan that removes edges with the largest centrality.

2.2.3 Combination of both clustering algorithms

A combination of clustering algorithms is proposed since every single clustering method has its own strengths and weaknesses, and a combination of two or more clustering algorithm may produce a better result than the best individual clustering algorithm.

The algorithm of Newman and Girvan is deterministic, meaning that nodes that are lying at the boundary between communities may not be clearly classified. To overcome this issue, self-adapted fuzzy c-means is introduced to classify the data points using membership degrees. Membership degrees value is used as a weight criteria that gives each data point more values. The higher the membership degree of a data point, the more it belongs to a cluster. The automated cluster number generation by the self-adapted fuzzy c-means clustering will be the input for the criterion to stop removing edges in the Newman-Girvan clustering algorithm.

To get these two valuable input, the self-adapted fuzzy c-means algorithm is first applied to the dataset. The membership matrix values and cluster number output from the self-adapted fuzzy c-means clustering will be used as additional input for the Newman-Girvan clustering algorithm. Newman-Girvan algorithm will classify the data points into several clusters until the stop criterion is met, and the result will be visualized to the end user interface. Figure 2.2 describes the architecture of the bibliographic big data visualization method.

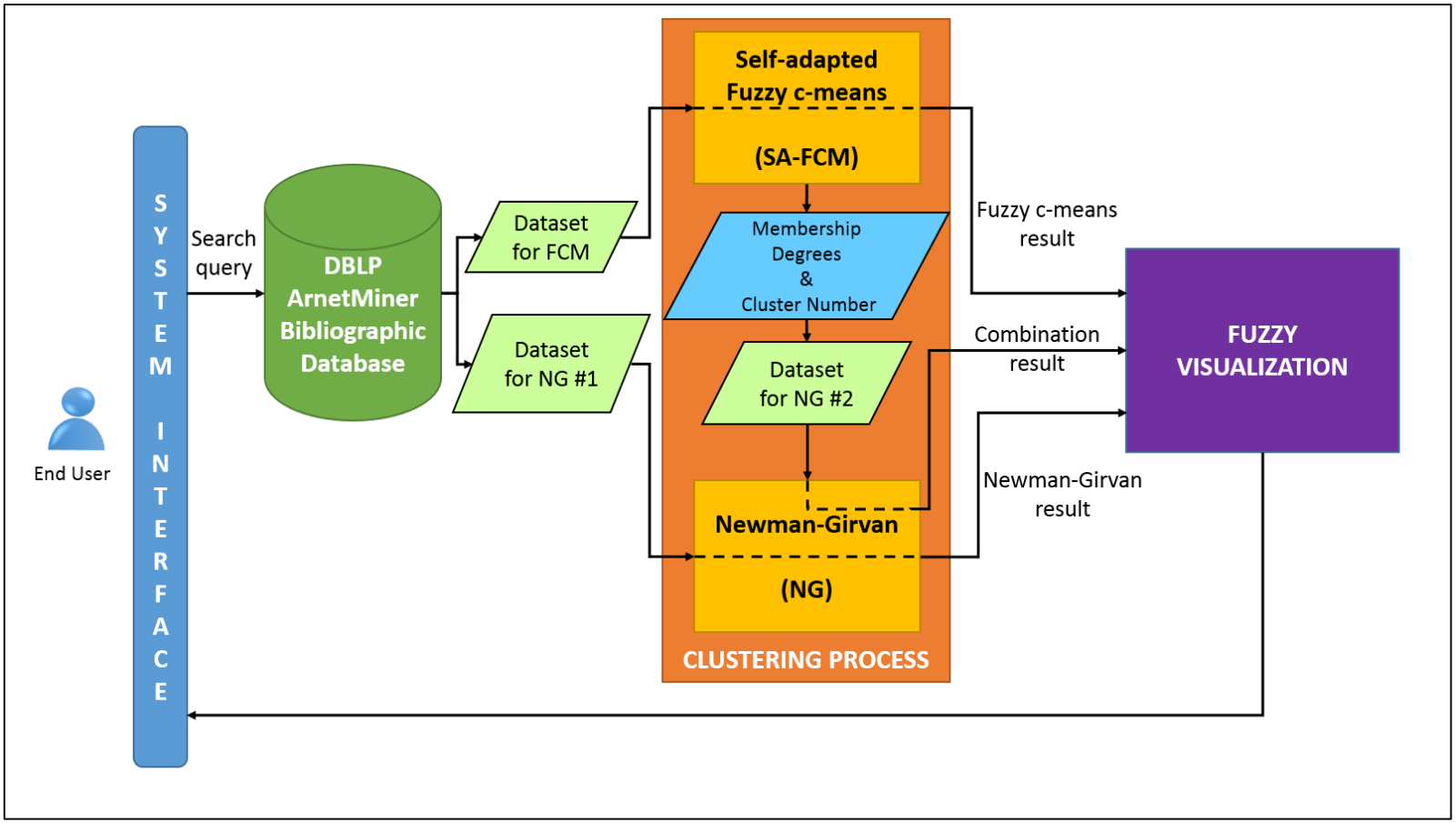


Figure 2.2 Bibliographic big data visualization method architecture.

2.3 Bibliographic Data DBLP

2.3.1 DBLP Bibliographic Data

Bibliographic big data used to test the proposed method is the DBLP dataset. DBLP is a computer science bibliography website hosted at Universität Trier, in Germany that provides bibliographic information on major computer science journals and proceedings. The DBLP dataset was selected to be used in the proposed method as it is freely available, it is one of the most recent dataset released for research purpose.

2.3.2 Citation Network Dataset by ArnetMiner

```
#*Fighting botnets with economic uncertainty.
#@Zhen Li,Qi Liao,Andrew Blaich,Aaron Striegel
#t2011
#cSecurity and Communication Networks
#index3133484
#%
#!

#*An efficient scheme to handle bursty behavior in secure group
communication using binomial key trees.
#@R. Aparna,B. B. Amberker
#t2011
#cSecurity and Communication Networks
#index3133489
#%
#!

#*Security enhancements for UDDI.
#@Alexander J. O'Ree,Mohammad S. Obaidat
#t2011
#cSecurity and Communication Networks
#index3133513
#%
#!

#*Cryptosystems based on continued fractions.
```

```
#@Ali Kanso
#t2011
#cSecurity and Communication Networks
#index3133504
#%
#!

#*Network specific false alarm reduction in intrusion detection system.
#@Neminath Hubballi,Santosh Biswas,Sukumar Nandi
#t2011
#cSecurity and Communication Networks
#index3133485
#%84141
#%486088
#%507406
#%516884
#%557534
#%659486
#%1098060
#%1132537
#!
```

Figure 2.3 Raw form of DBLP dataset prepared by ArnetMiner.

The dataset is provided by ArnetMiner, an online service used to index and search academic social networks[15,16], available at http://arnetminer.org/ DBLP_Citation . It contains citation relationships between DBLP papers where each node is a paper from DBLP, and is further associated with abstract and citation relationships. Figure 2.3 shows the DBLP dataset in its raw form.

From the total of 1,511,035 entries in the DBLP Citation Network Dataset prepared by ArnetMiner, 492,550 unique authors have been identified. Out of these unique authors, there exist at least 66,801 connections among them.

The DBLP dataset was accessed on 12 April 2013. This is the Version 5 of the dataset that is prepared by ArnetMiner for DBLP citations up until 21 February 2011. Therefore it does not store information of papers that are published after the release date.

Self-adapted fuzzy c-means methods expects an input data file in the format as shown in Figure 2.4. The example of a valid input file for fuzzy c-means is shown in Figure 2.5.

```
Line 1: <number of data-points> <number of dimensions>
Line 2: <fuzziness coefficient> <termination criterion>
Line 3: <data points> …
```

Figure 2.4 Dataset format for fuzzy c-means clustering.

```
11 2
2.0 5.0E-4
1118466 773276 1118466 42 599096
642966 42 599096 642966 773276
```

Figure 2.5 Dataset sample for fuzzy c-means clustering.

For Newman-Girvan clustering algorithm, the DBLP dataset needs to be transformed into Pajek NET format[25] shown in Figure 2.6. Pajek files are text files, where each line is an element, and the list of edges follows the list of nodes. It is supported by nearly most of graph softwares. In Pajek format, nodes have basically one unique identifier and a label. The definition of nodes starts with the chain *Nodes N where N is the number of nodes following.

```
*Nodes
1 "Jose A. Blakeley"
2 "Yuri Breitbart"
3 "Stavros Christodoulakis"
4 "Umeshwar Dayal"
5 "Angelika Kotz Dittrich"
…
*Edgeslist
281 95 567
320 80 1656 1873 2922
333 1656 68 84
339 2719 587
355 237240 2041
358 63504
372 1656
…
```

Figure 2.6 Dataset for Newman-Girvan clustering without weight information.

For edges, there are two ways of representing them. In the first way, the first identifier is the source node and all following are the neighbors. The dedicated marker is *Edgeslist. This way of representing edges is used in the individual Newman-Girvan clustering without weight information to make a comparison with the individual fuzzy c-means algorithm the combination of both clustering algorithms. The dataset in Figure 2.6 consists the entire DBLP dataset of 492550 nodes, where the labels are quoted directly after the nodes identifier.

The second way to represent the edges is where the edges are defined as pair of nodes identifier. The *Arcs marker goes before the pairs list. The weight is added by a third column, as shown in Figure 2.7. In the proposed method, this type of dataset is used in the combination of the clustering method. The membership degree result of the self-adapted fuzzy c-means clustering used as weight for each data point for the Newman-Girvan clustering algorithm.

```
*Nodes 42
124812
158591
…
*Arcs
1       13        0.004125401364886321
1       14        0.004125401364886321
1       15        0.004125401364886321
…
```

Figure 2.7 Dataset for Newman-Girvan algorithm with membership degrees as weight for each data point.

2.3.3 Bibliographic Big Data

Big data is useful in obtaining critical insights from the processing and the behavior of the data. The challenge when working with big data is to find a way to perform automated processing capability that is extremely fast and produces meaningful results to the users.

The large volume of DBLP citation network dataset is ideal for the proposed method as bibliographic data are naturally big and highly connected to one another. The proposed method first requires the user to enter a keyword and then quickly identifies which data matches the user queries. To ensure the clustering process is able to return fast results, only relevant data should be included in the clustering process. Irrelevant data is kept in the database for later use.

2.4 Fuzzy Visualization using Java Universal Network/Graph

2.4.1 JUNG

The visualization of this method is developed using JUNG (the Java Universal Network/Graph Framework)[17]. JUNG is a free, open-source software library written in Java that provides a common and extendible language for the manipulation, analysis, and visualization of data that can be represented as a graph or network.

JUNG is used in the development of the visualization part of the proposed method. It is selected because it offers a simple yet attractive way to construct tools for the interactive exploration of network data. It also enables visualization to be portrayed in an interactive network view. Nodes are clustered based on their membership degree and connections are shown as edges between related nodes.

There are two main classes in JUNG library that is used in the proposed method. First is the EdgeBetweennessClusterer[17] class that computes clusters for a graph based on the betweenness property of the edges. Second is the WeakComponentClusterer[17] class that finds all weak components in a graph, and the weak component is defined as a subgraph in which each pair of nodes is connected at least by one undirected graph.

The rendering of the clustering visualization uses node and edge color function to create the visual contrast between elements. JUNG also allows user to focus their attention on specific portions of the graph. By doing this users will be able to interactively explore and manipulate the search result to enhance their understanding of the target result.

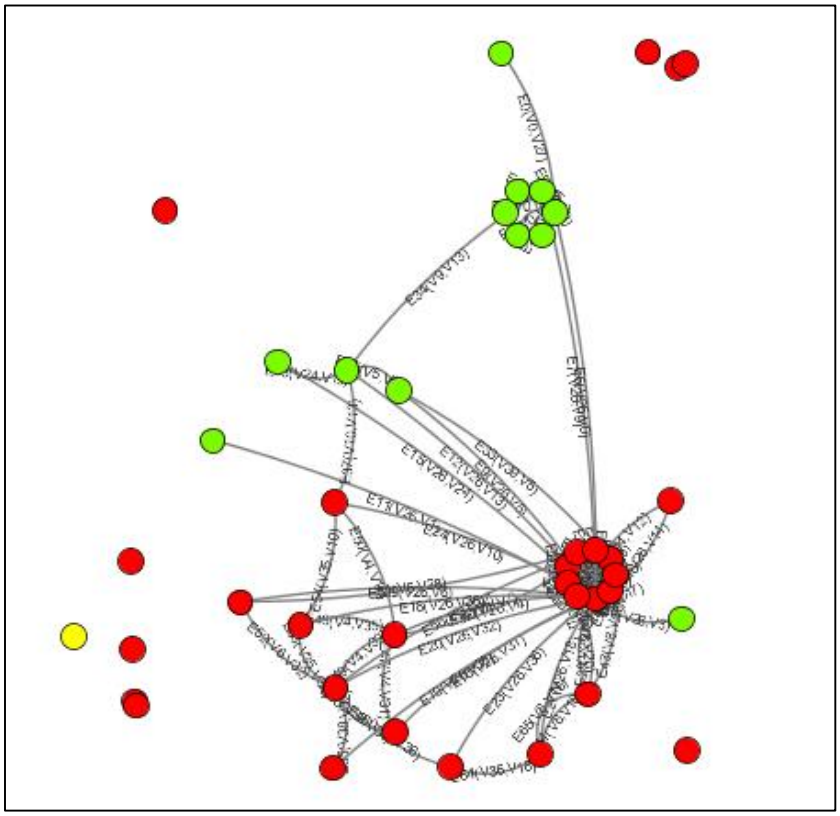


Figure 2.8 Example of JUNG visualization

2.4.2 Fuzzy Methods Visualization Requirements

Fuzzy-based clustering methods have different visualization requirements from crisp clustering methods as it brings more complexity to the visualization aspect. It is important for the user of such fuzzy system to effectively visualize and interpret the information and their propagation. The objective of applying fuzzy information visualization is to retain valuable and higher quality knowledge resulted from the fuzzy clustering method[26]. By showing the

membership degrees through visualization techniques, search results can be displayed in a manner that can help user focus on the target data and pay less attention to the others [27].

In the proposed method, the search result is displayed in graph view based on the Fruchterman-Reingold algorithm[28] as shown in Figure 2.8. It incorporates two principles for graph drawing, which is the nodes connected by an edge should be drawn near each other and the nodes should not be drawn too close to each other. It was chosen as the algorithm is good at distributing nodes evenly, making edge lengths uniform, and reflecting symmetry. The main factors of using this algorithm is due to its implementation speed and simplicity. Graphs drawn using Fruchterman-Reingold were drawn in less than a second, which is crucial in the proposed method as it needs to achieve the time limit of 5 minutes of less.

In the proposed method, the nodes represent the search category, such as author, paper title, year, or publication venue, and the edges represents the connections between each node.

One of the desirable characteristics of visualization for fuzzy clustering result is the seamless integration of the result and its fuzzy values. The visualization technique used in the proposed method to realize this characteristic is by using different colors and brightness as the intrinsic representations of the nodes. Multiple dark colors are used to differentiate nodes according to their cluster and light colors are used in nodes that do not belong to any cluster.

To further represent the relation of each node, different thickness of the edges will show the strength of their connection. This is determined by the membership degrees of each node.

To facilitate the decision making by the users, it is ideal to offer interactive functionalities that could help the users get more information. The proposed method allows user to click on each node to find more detailed information.

2.5 Experiment on Journal Papers Retrieval from DBLP

2.5.1 Investigating the performance of the combination of self-adapted fuzzy c-means and Newman-Girvan algorithm

An experiment is performed to confirm that the combination of self-adapted fuzzy c-means and Newman-Girvan clustering algorithm performs better than the individual clustering algorithm. To achieve this objective, the combination of clustering algorithms is applied to the DLBP citation network dataset prepared by ArnetMiner.

2.5.2 Comparison of fast decision making

The massive nature of bibliographic big data requires for the combination of the two clustering algorithms to be optimized in order to ensure that several matched result based on the user queries can be obtained in 5 minutes or less.

A method that aims at integrating fuzzy classification algorithms with an interface to visualize fuzzy results is introduced[29] where users are allowed to enter keyword(s) or browse from categories of pre-classified data. The classification algorithm is optimized for speedy return of result where the indexing is simplified by classifying each item upon entry to the database rather than when a query is input. The upper levels of a hierarchy are saved along with the item to speed matching at the time of query entry.

The proposed method deals with bibliographic big data. Therefore it needs a mechanism to reduce the number of nodes and edges to ensure the visualization of the result is not overwhelming for users to comprehend. The crucial task is to ensure that only relevant

information is displayed. Therefore, when the user enters a keyword, the method will first select matching data from the database. The data retrieval is focused only on data that matches the keyword and the retrieved data is transformed into a format suitable for clustering purposes. By using this mechanism the clustering algorithms will only be applied on the retrieved data instead of the whole dataset.

2.5.3 Experiment Settings

To conduct the experiment, a software program is developed to get keyword input information from users, their preferred categories, and to show the visualization of the clustering result to the users. The program is developed in Java language and the experiments are carried out in Eclipse IDE 4.2.2[30] using Dell Latitude E5430 laptop with Intel (R) Core (TM) i5-3210M at 2.50GHz. A prototype of the software program is shown in Figure 2.9.

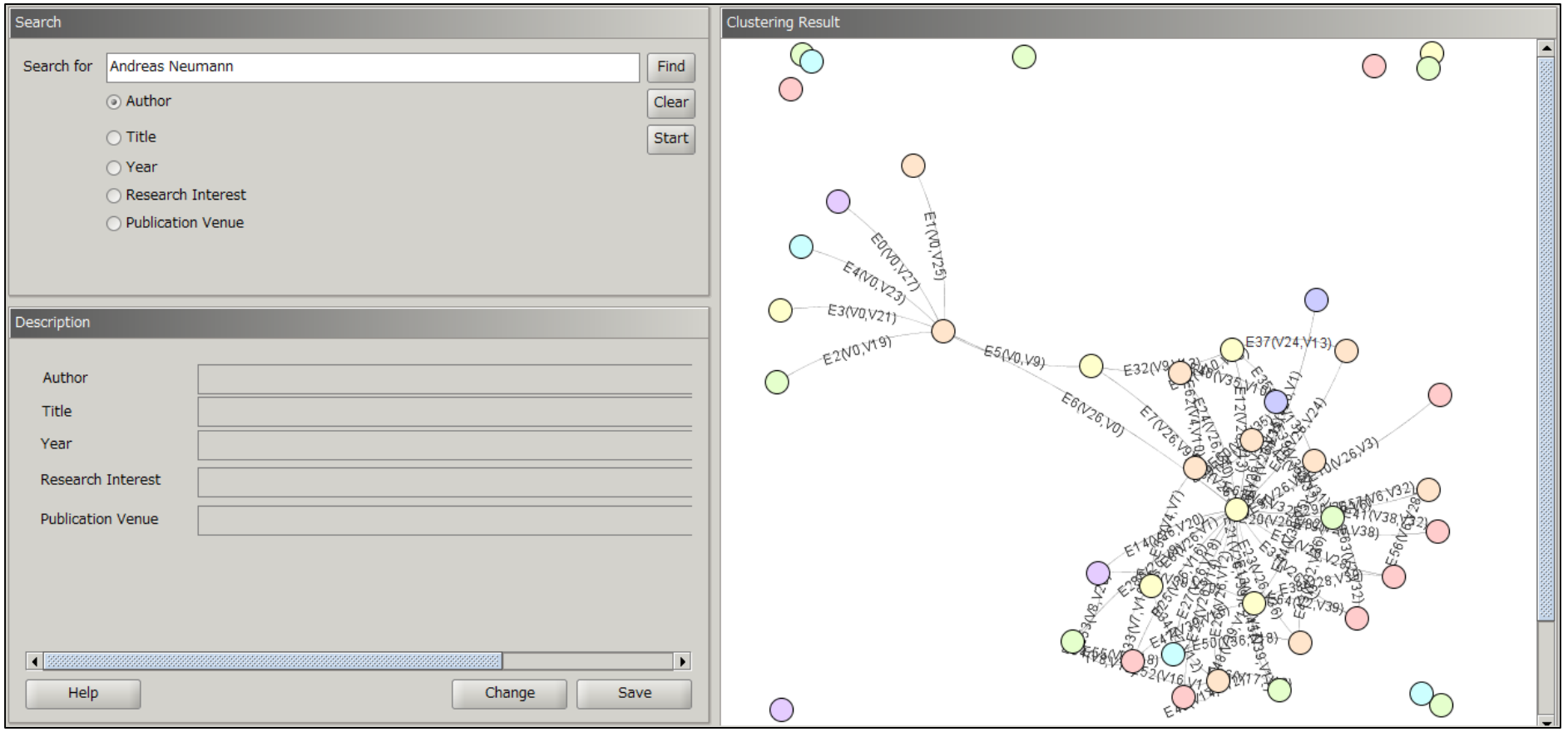


Figure 2.9 Bibliographic big data search method's user interface

2.5.4 Applying clustering algorithms on DBLP citation network dataset

To start the process, first users will enter a keyword of papers that they want to find in the bibliographic big data. They need to choose a category of the search, i.e., by the authors, title, publication year, or publication venue. Next, they click on the button 'Find' and the method will search for related papers that contain the keywords. After all related papers have been gathered from the database, users will select the button "Start Clustering" to begin the clustering process. The visualization result of the clustering process is shown in the right side of the user interface.

Steps of the clustering process are described as follows :

Step 1. Create dataset for self-adapted fuzzy c-means clustering

The keyword "Andreas Neumann" will generate the dataset as shown in Figure 2.10. The dataset contains 42 data found in the DBLP Citation Network Dataset that is related to the keyword "Andreas Neumann".

```
42 2
2.0 5.0E-4
1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 …
13 14 15 16 17 18 1 18 19 20 …
```

Figure 2.10 Dataset for the self-adapted fuzzy c-means clustering based on the search keyword "Andreas Neumann".

Step 2. Perform the self-adapted fuzzy c-means algorithm.

The self-adapted fuzzy c-means algorithm is performed on the dataset. A fuzzy c-means java program is used in the process with added self-adapted functions to ensure that the number

of clusters can be automatically set. The result from the program are displayed in a membership matrix as shown in Figure 2.11.

```
Membership matrix :
0.004125401364886321
0.004125401364886321
0.002428400884306878
…
```

Figure 2.11 Membership matrix result from the fuzzy c-means method.

The membership degree of each data point is used as a weight as added information in the dataset used in Newman Girvan clustering algorithm. With the added weight information, a more precise result is able to be obtained when the second algorithm is performed on the dataset.

Step 3. Create dataset for Newman-Girvan clustering

The dataset for the Newman-Girvan clustering algorithm is created after result have been produced by the self-adapted fuzzy c-means. The membership matrix is used as weight for each edges, and the number of clusters created is used to indicate when Newman-Girvan algorithm should stop dividing its graphs into smaller clusters.

```
*Nodes 42
1       124812
2       158591
…
*Arcs
1       13        0.0041254013648863
1       14        0.0041254013648863
…
```

Figure 2.12 Newman-Girvan dataset with weight from self-adapted fuzzy c-means membership degrees.

Step 4. Perform the Newman-Girvan algorithm

The dataset in Figure 2.12 is used for the Newman-Girvan algorithm procedure. A java program using EdgeBetweennessClusterer library in JUNG is used to perform the algorithm. It

computes clusters in graphs based on edge betweenness, where the betweenness of an edge measure the extent to which that edge lies along shortest paths between all pairs of nodes.

Edges which are least central to communities are progressively removed until the communities have been adequately separated. It works by iteratively following the 2 step process:

1. Compute edge betweenness for all edges in current graph.
2. Remove edge with highest betweenness.

The stop criterion for the iteration is the number of cluster generated by the self-adapted fuzzy c-means algorithm. When the number of cluster generated in the Newman-Girvan algorithm is smaller than the number of cluster generated by the self-adapted fuzzy c-means algorithm, the iteration will stop.

Step 5. Visualize clustering result using fuzzy visualization method.

After both clustering algorithms have been applied to the dataset, the visualization of the result are realized by implementing Jung libraries in Java using Frutcherman-Reingold algorithm, with added fuzzy visualization features that gives the user a more in-depth information of the search result as compared to the usual crisp visualization features of current bibliographic visualization methods.

2.5.5 Measures for effectiveness evaluation of the proposed method

Since the dataset used for each experiment depends on the keyword input information and category selected by users, it is ensured that the clustering result includes almost all data

that user wants. To measure the effectiveness of the combination of clustering method, i.e., precision, recall, and f-measure [31], are calculated using

$$\text{Precision(P)} = \frac{\#(\text{relevant papers retrieved})}{\#(\text{relevant papers})}, \quad (1)$$

$$\text{Recall (R)} = \frac{\#(\text{relevant papers retrieved})}{\#(\text{retrieved papers})}, \quad (2)$$

$$\text{F} - \text{measure} = \frac{2\text{PR}}{\text{P} + \text{R}}. \quad (3)$$

Precision is defined as the proposed method's ability to retrieve papers that are mostly relevant, while recall indicates the ability of the proposed method to find all of the relevant papers in the database. F-measure is a harmonic mean that trades off precision versus recall. F-measure is also used to measure the method's performance as it will give an even weight to both precision and recall.

2.5.6 Clustering result and visualization

2.5.6.1 Comparison of clustering result by using self-adapted fuzzy c-means clustering, the Newman-Girvan clustering algorithm and the combination of both algorithms.

Table 2.1 Experiment result for self-adapted fuzzy c-means clustering, the Newman-Girvan clustering algorithm and the proposed method on author search keyword "Andreas Neumann".

Algorithm	**Clusters**	**Nodes**	**Time**
Newman-Girvan	6	24	1m 50s
Self-adapted fuzzy c-means	2	9	3m 24s
Combination	1	5	4m 08s

Table 2.1 shows the number of clusters found using each of the individual clustering method and the combination of clustering method using the keyword “Andreas Neumann. When the Newman-Girvan algorithm is performed on the “Andreas Neumann” dataset, 6 clusters were found in 1 minute and 50 seconds. Self-adapted fuzzy c-means algorithm generates 2 clusters from the same dataset in 3 minutes and 24 seconds. When a combination of both algorithms is applied to the same dataset, it took 4 minutes and 8 seconds to find one cluster that highly matches the search keyword by the user. Table 2 shows the experiment result for self-adapted fuzzy c-means clustering based on 9 search cases from 3 search categories, by author, by title, and by publication venue.

Table 2.2 Experiment result for self-adapted fuzzy c-means clustering algorithm

Self-adapted fuzzy c-means algorithm			
Keywords	Time(s)	Clusters	Nodes
Author search			
#1: “Andreas Neumann”	204	2	9
#2: “Edward Omiecinski”	189	5	12
#3: “William Kent”	165	4	12
Title search			
#4: “Big Data”	61	3	11
#5: “Fuzzy Clustering”	259	8	36
#6: “Community Network”	94	3	18
Publication search			
#7: “Information Technology Management”	171	5	18
#8: “Wireless Communications Mobile Computing”	130	3	9
#9: “Scalable Computing: Practice and Experience”	156	2	9

Table 2.3 shows the experiment result for Newman-Girvan clustering algorithm and Table 2.4 shows the experiment result for the combination of both algorithms.

Table 2.3 Experiment result for Newman-Girvan clustering algorithm

Newman-Girvan clustering algorithm			
Keywords	Time(s)	Clusters	Nodes
Author search			
#1: “Andreas Neumann”	110	6	24
#2: “Edward Omiecinski”	200	8	25
#3: “William Kent”	152	4	8
Title search			
#4: “Big Data”	63	4	16
#5: “Fuzzy Clustering”	265	11	58
#6: “Community Network”	98	5	31
Publication search			
#7: “Information Technology Management”	173	7	26
#8: “Wireless Communications Mobile Computing”	132	3	10
#9: “Scalable Computing: Practice and Experience”	159	3	11

Table 2.4 Experiment result for combination of both clustering algorithms

Combination of both algorithms			
Keywords	Time(s)	Clusters	Nodes
Author search			
#1: “Andreas Neumann”	248	1	5
#2: “Edward Omiecinski”	209	2	6
#3: “William Kent”	178	3	7
Title search			
#4: “Big Data”	70	2	9
#5: “Fuzzy Clustering”	271	6	32
#6: “Community Network”	102	2	12
Publication search			

#7: “Information Technology Management”	176	4	14
#8: “Wireless Communications Mobile Computing”	136	2	7
#9: “Scalable Computing: Practice and Experience”	161	1	4

2.5.6.2 Time evaluation of clustering processes

Table 2.5 The response time comparison for the three clustering algorithms

Clustering Algorithm	**Mean**	**Standard Deviation**
Self-adapted fuzzy c-means algorithm	158.78	64.92
Newman-Girvan clustering algorithm	150.22	59.72
Combination of both algorithms	172.33	214.75

Table 2.5 shows the mean and standard deviation of response time for all three clustering algorithms. The response time of the hybrid combination of self-adapted fuzzy c-means and Newman-Girvan algorithm has the highest mean, 214.75 compared to 158.78 for self-adapted fuzzy c-means and 150.22 for Newman-Girvan algorithm. It means that the combination algorithm tend to take longer time to be completed than the individual clustering algorithm. This is due to the extra processes that the combination algorithm needs to perform to get the final result. However, the combination of the algorithm time also has a higher standard deviation of 214.72, as opposed to 64.92 of the fuzzy c-means and 59.72 of the Newman-Girvan algorithm. It means that the time required to complete the combination of algorithms varies from case to case. This is due to the size of dataset every time a search is done. The bigger the search result, the longer time it needs to gather all related data and perform the algorithms on them. For example, in title search, if the keyword input information is related to a well-known research area, such as “Fuzzy Clustering”, the keyword will yield a massive amount of data that requires a longer time to process them, compared to a less well-known research area.

Even though the mean time of combination of algorithms is the highest among the 3 algorithms, the important factor is that the combination of algorithms is able to focus on several nodes that really matches the keyword input information. The individual clustering algorithms yielded larger number of clusters and total number of nodes, thus requiring users more time and effort to examine each one of them. This point reaches the target of the proposed method that aims to converge a few target papers in average 5 minutes or less from more than 1.5 million papers stored in the DBLP.

2.5.6.3 Performance evaluation of clustering processes

Table 2.6 Precision, recall, and f-measure comparison between 3 clustering algorithms

Clustering Algorithm	**Precision**		**Recall**		**F-measure**	
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**
Newman-Girvan	0.592	0.242	0.530	0.266	0.530	0.236
SA-FCM	0.633	0.109	0.706	0.104	0.658	0.058
Combination	0.751	0.103	0.711	0.107	0.724	0.073

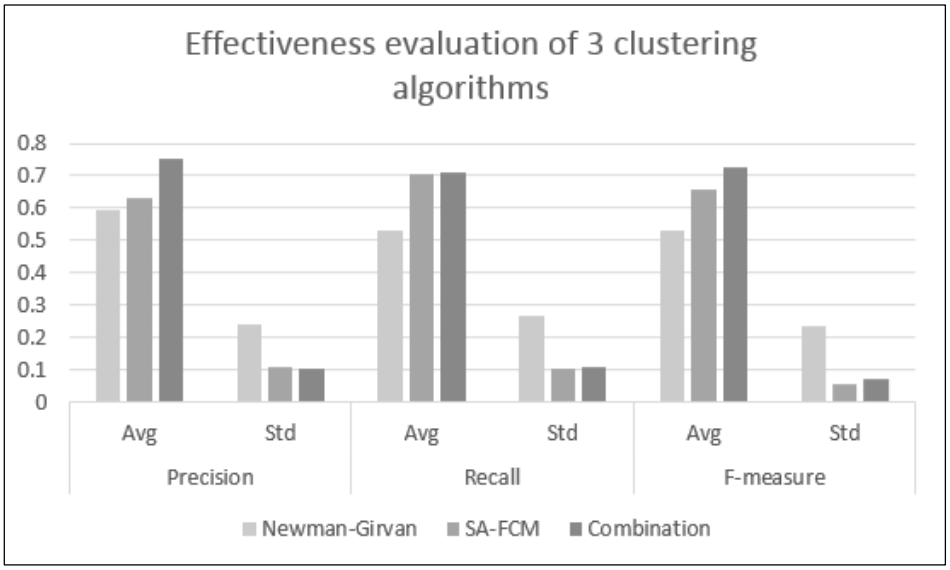


Figure 2.13 Comparison chart of 3 clustering algorithms' performance.

Table 2.6 shows the mean and standard deviation of precision, recall, and f-measure for all three clustering algorithms and Figure 2.13 shows the comparison chart for the evaluation.

It is clearly shown that the average precision of the combination of clustering algorithms gets the highest value of 0.751 compared to the other two individual algorithms. The combination also has the lowest standard deviation of precision that means the value of precision for each search case does not vary greatly.

For recall, the combination of clustering algorithms gets the highest average value of 0.711 with the self-adapted fuzzy c-means following closely with 0.706 average recall. The standard deviation of recall for combination of clustering algorithms is slightly higher than the self-adapted fuzzy c-means clustering. This means the recall value for the combination of algorithms vary slightly more than the self-adapted fuzzy c-means algorithm.

F-measure is used to measure the performance of the proposed method that takes into account both precision and recall. The result shows that the average f-measure for the combination of algorithms gets the highest value of 0.724 compared to 0.658 for self-adapted fuzzy c-means and 0.530 for Newman-Girvan algorithm.

The overall result show that there is a significant numerical improvement of the combination of self-adapt fuzzy c-means and Newman-Girvan algorithm when compared to each individual clustering algorithm.

From practical use point of view, the improvement is sufficient, since the combination is able to help users to focus on several highly related results of their search. The less precise result is not clustered, but they are still available in the visualization. The users is able to select the unclustered result and get the information that they need, if desired.

To ensure the clustering result appeals to the users, a prototype of the proposed method is planning to be tested using user-based evaluation method. In this evaluation method, the prototype is tested by selecting a number of participants to perform a set of pre-determined tasks on the prototype. A feedback questionnaire will be given to the participants after the tasks have been performed and the participants are requested to fill in the questionnaire based on their opinion of the prototype. The practical usability of the proposed method can be determined from the feedback result. This evaluation process is planning to be conducted when the prototype is completed.

2.5.7 Comparison of visualization technique by using non-fuzzy technique and fuzzy requirement technique

Figure 2.14 shows main view of the visualization. The view is divided into two sides. The left side contains search area and paper description area. The right side is the visualization of clustering result area. In search area, users are able to search from four categories, by author, by title, by publication year, or by publication venue. The paper description area on the bottom left side displays information of a node that is selected by the user. The visualization area on the right side displays the clustering result in network view.

In Figure 2.14, the visualization part is displaying the crisp relationship of papers related to the author "Andreas Neumann". The search yields a total of 42 papers related to the keyword input. This includes papers written by the author and papers that cites the paper written by the author.

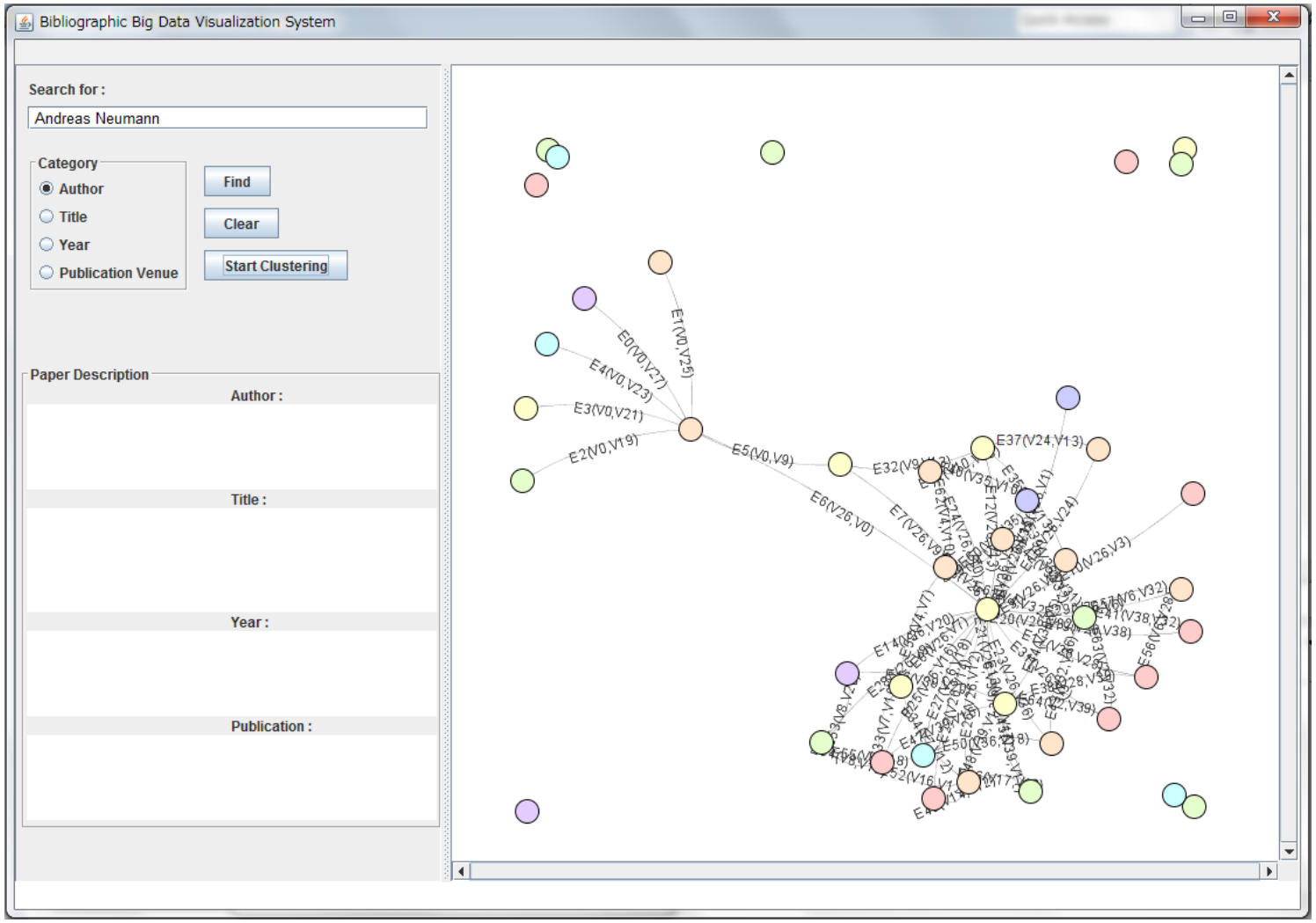


Figure 2.14 Crisp relationship of the dataset based on the author keyword “Andreas Neumann”.

Initially, the result will show the crisp connection among the papers that are relevant to the search keyword before clustering steps are performed. Since the nodes are not yet clustered, it is displayed in a variety of light colors without any uniformity

The users need to press the ‘Start Clustering’ button for the clustering algorithms to be performed on the result.

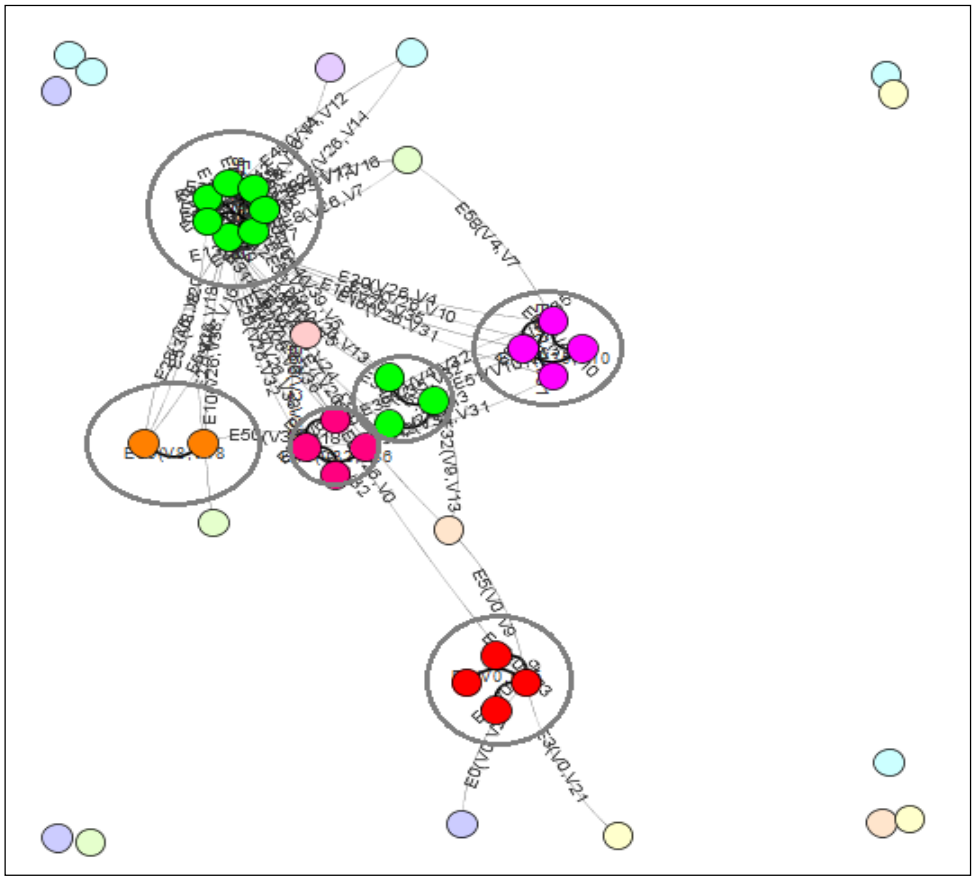


Figure 2.15 Visualization of Newman-Girvan clustering result for keyword “Andreas Neumann”

In Figure 2.15, clustering is done based on Newman-Girvan algorithm. After removing half number of edges to find desirable number of clusters, 6 clusters are generated from the result. There are 5 dark colors used to paint the nodes in each cluster, i.e., green, red, magenta,

pink, and orange. Even though there are 2 clusters painted in green, but since they are located far from each other, it gives the idea that they do not belong to the same cluster. The nodes that do not belong to any cluster are painted in various light colors. Based on Figure 2.15, the Newman-Girvan algorithm is found to be suitable for finding a desired number of clusters but unable to focus on few important clusters that are useful to the user.

Figure 2.16 Visualization of self-adapted fuzzy c-means clustering result for keyword "Andreas Neumann"

In Figure 2.16, the self-adapted fuzzy c-means clustering algorithm is performed on the Andreas Neumann dataset. Two clusters have been created voluntarily by performing the algorithm. The thickness of edges between the nodes represents the strength between two nodes. Since the edges connecting the nodes are thin, we can say that the nodes are not highly related

to each other. The colors of the nodes are also ununiformed. Based on these two visual representations, users will be able to get the information that the connection between each node is not highly relevant.

The application of combination of self-adapted fuzzy c-means and Newman-Girvan algorithm has resulted in one main cluster consists of five nodes painted in dark red as shown in Figure 2.17. The result clearly shows that there are 5 papers that are highly relevant to the keyword input information. The edges connecting the nodes are thick, thus giving users visual information that the papers are highly relevant to each other. Two nodes that are connected to the cluster but have light, inconsistent colors and the edges connecting them are thin. These two visual representations give the user hints that the two nodes are not highly relevant as compared to the five red nodes.

Shown in Figure 2.18→Fig2.20 is the visualization result of author search using the keyword “Edward Omiecinski”, in Newman-Girvan algorithm, the self-adapted fuzzy c-means algorithm, and combination of both algorithms. Figure 2.21→Figure 2.23 shows the visualization of clustering result of author search using the keyword “William Kent”.

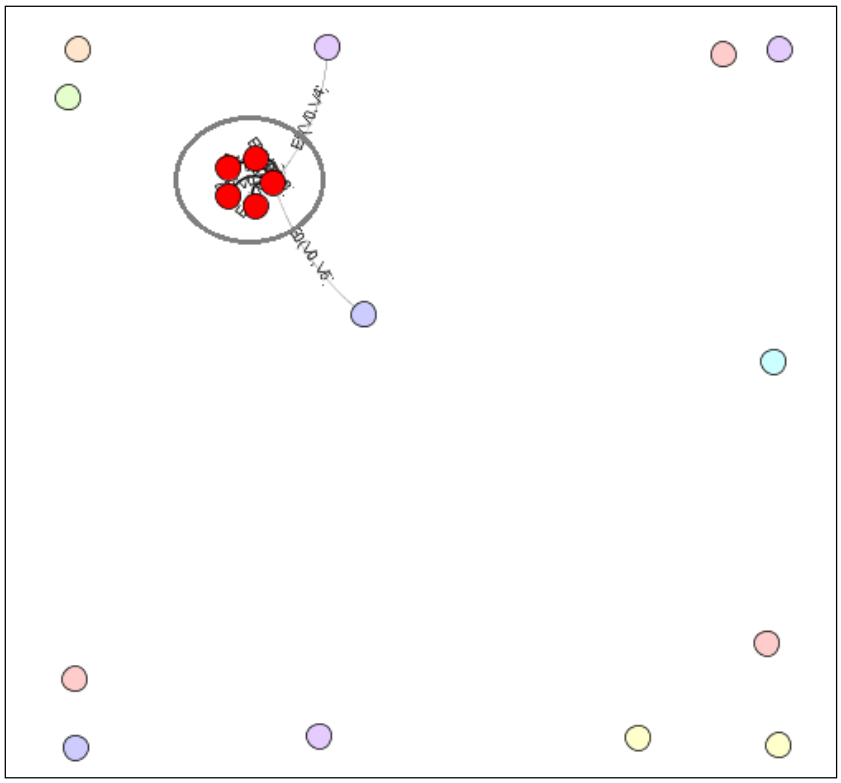


Figure 2.17 Visualization of combination algorithm for keyword “Andreas Neumann"

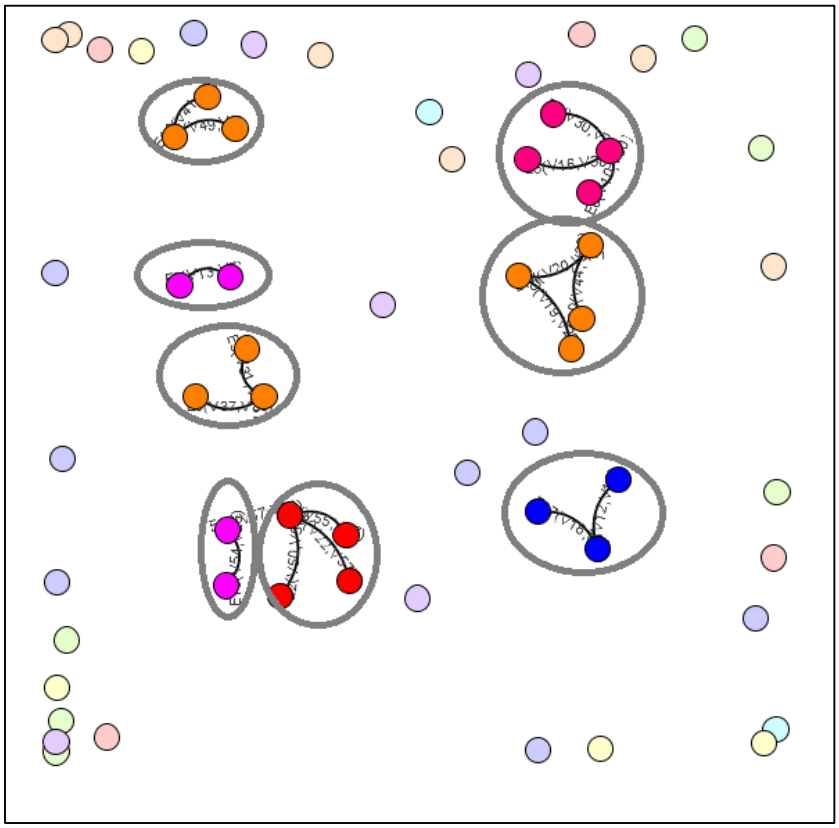


Figure 2.18 Newman-Girvan clustering result for author search #2: “Edward Omiecinski”.

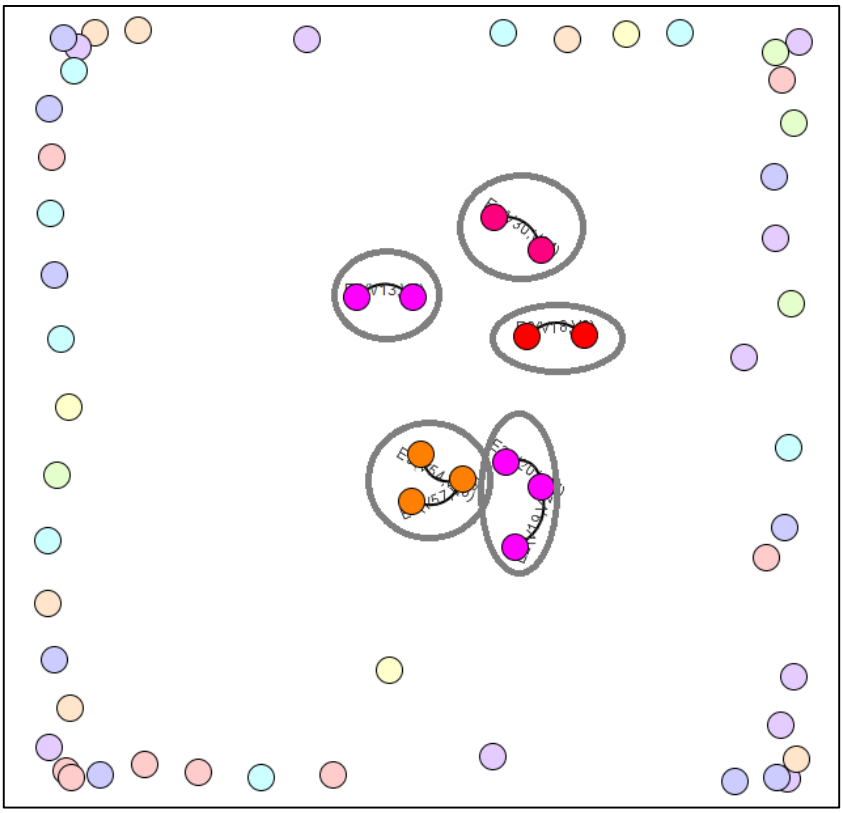


Figure 2.19 Self-adapted fuzzy c-means clustering result for author search #2: “Edward Omiecinski”.

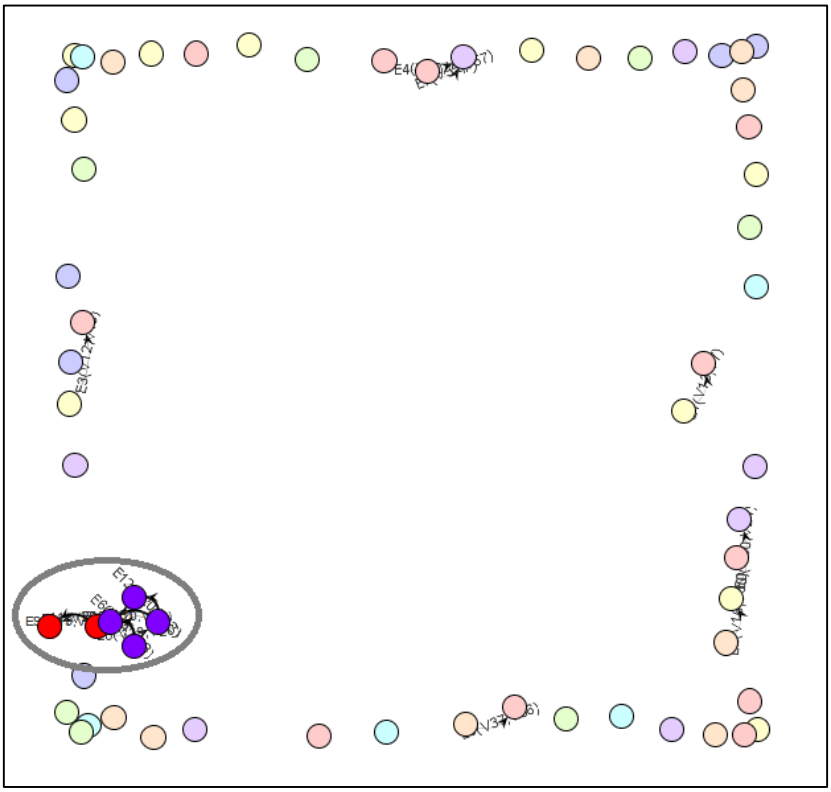


Figure 2.20 Combination clustering result for author search #2: “Edward Omiecinski”.

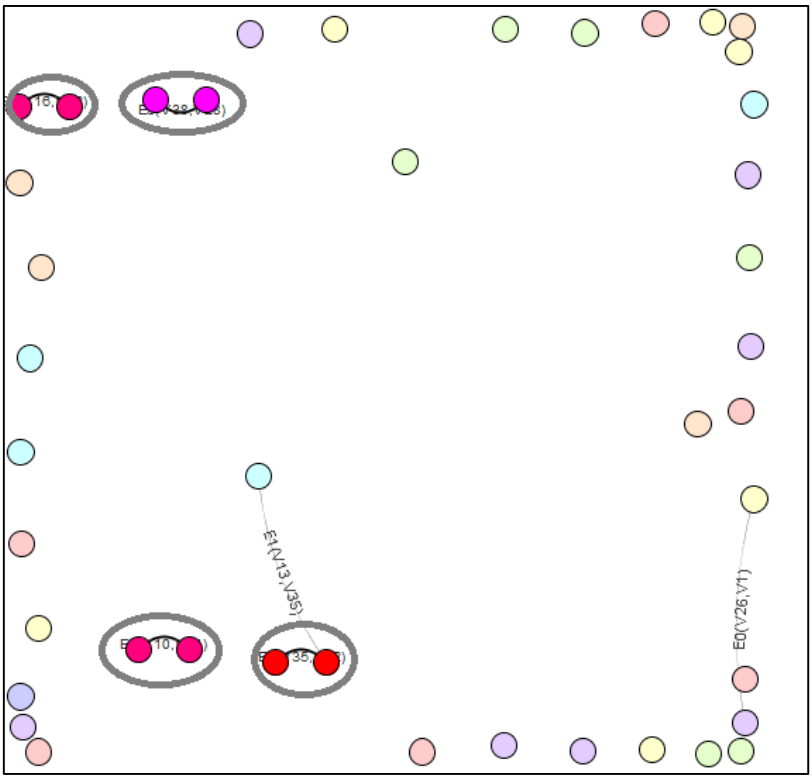


Figure 2.21 Newman-Girvan clustering result for author search #3: “William Kent”.

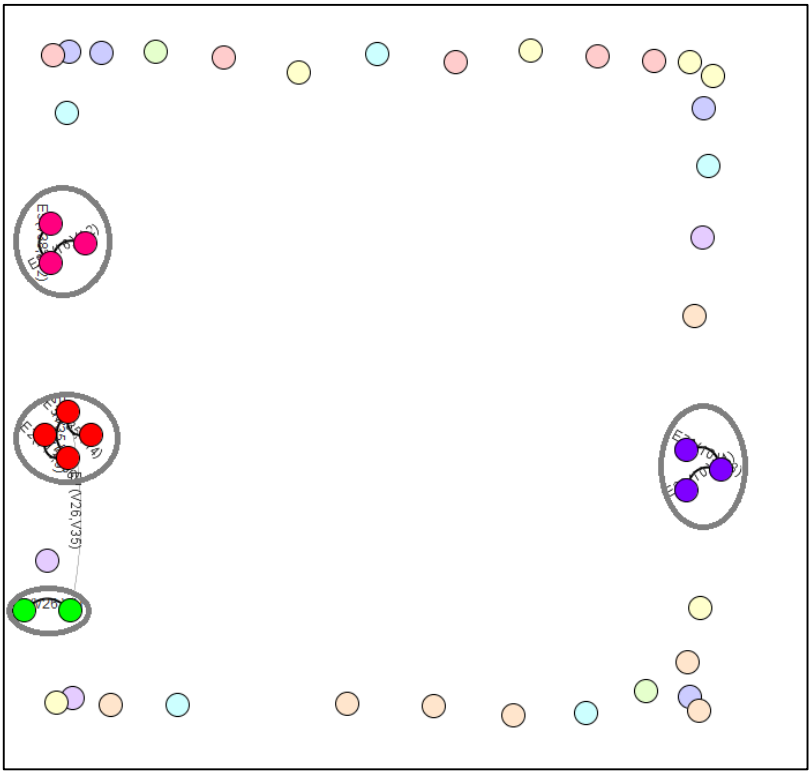


Figure 2.22 Self-adapted fuzzy c-means clustering result for author search #3: “William Kent”.

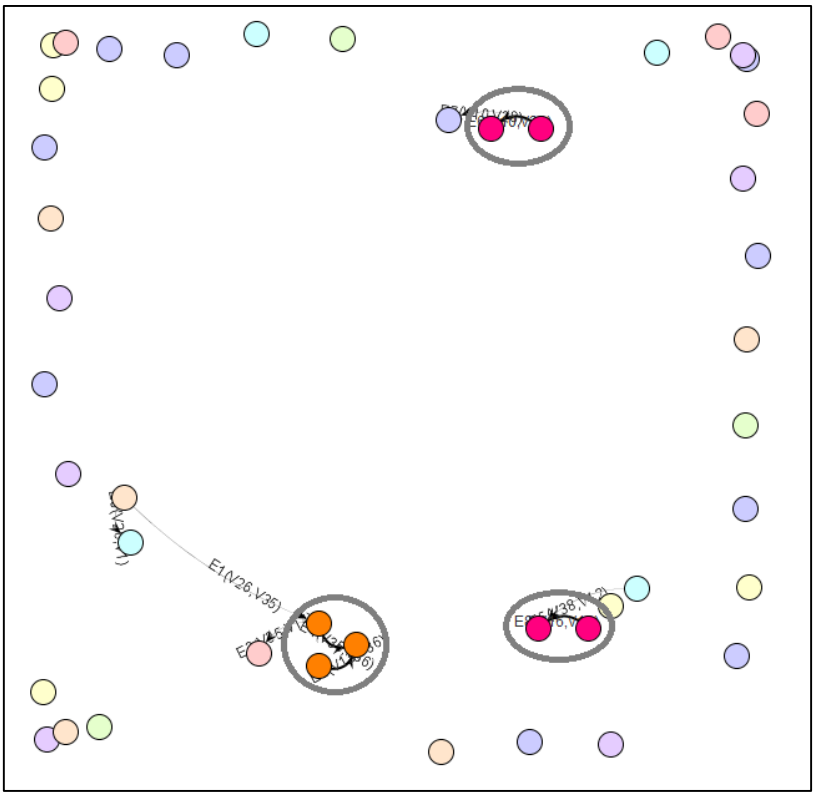


Figure 2.23 Combination clustering result for author search #3: “William Kent”.

When the user click on each node, detailed information of the paper such as the paper title, list of authors, publication year, publication venue, citation count, reference list and abstract will be displayed. The detailed information is intended to facilitate users to make decisions for further action.

2.6 Chapter Summary

The experimental result confirms that by combining two clustering algorithms and visualizing the search result using fuzzy techniques, users are able to converge a few target papers in average 5 minutes from 1.5 million papers stored in the DBLP.

Based on the keyword input information by the user, the search result is shown in a form of a small network, where the nodes indicate the papers that match the keyword, and the edges indicates the connection between the nodes.

Nodes that are highly relevant to the keyword are displayed in a manner that distinguish them from nodes that have less relevant by using several colors and different level of darkness. The proposed visualization technique helps users to focus on the nodes that match the keyword the most.

By selecting each node, a more detailed explanation on the paper is displayed, such as the paper title, authors, publication year, publication venue, citation count, reference list, and the abstract of the paper.

The combination of two fuzzy clustering algorithms is intended to overcome the limitations in individual clustering algorithm. But this does not necessarily mean that the combination of algorithms would produce the fastest result every time. To further improve the proposed method, it should be able to perform automated comparison of the result of each individual algorithm as well as the combination of both algorithms to find out which one could yield the fastest result. The retrieval process using keyword input information can also be improved by applying fuzzy ontology combined with fuzzy logic and descriptive logic to represent overlapping and imprecise keywords. It can increase the precision of the retrieval result before the clustering process is done on the dataset of the retrieval result.

Bibliographic big data brings challenges for creating visualizations that display results of exploration in a way that is not overwhelming to assist users to explore data easily, and the results are made available quickly for faster decision making. The modeling of fuzzy

information to visual elements are introduced to solve problems of visualizing the relationships across a variety of domains in knowledge.

Target users of the proposed method are supposed to be researchers, educators, and learners who are using real world networks such as social network and biological network, and the proposal is planning to be opened for public through the Internet.

Chapter 3

Automatic Switching of Clustering Methods based on Fuzzy Inference in Bibliographic Big Data Retrieval System

3.1 Introduction

An automatic switch developed from fuzzy inference engine is a method that compares the result from 3 clustering methods based on their performance and selects the best performing algorithm to produce visualization result to the users. It is applied in the Bibliographic Big Data Retrieval System[32] that accepts user keywords to search for the relationship among bibliographic big data consisting of journal/conference papers.

Existing switching method includes a switch between classifiers[33] that suggests when to combine classifiers and how classifier selection effects the result. But the switch is not able to predict whether a combination of classifier can achieve a better result than individual classifier. A fuzzy inference system is used to assess the performance of a conventional power plant[34] that has a highly flexible algorithm as it can handle fuzzy data, crisp data, and data complexity. But it also uses singular clustering method as it performs better when compared to ANN-FCM combination as it is able to explore performance patterns and select the better ones. Another fuzzy inference system that speeds up processes is GSM churn management by using FCM and ANFIS[35] but the system only utilizes singular clustering method, and the

complexity of the algorithm is high when the input volume is high. In the Bibliographic Big Data Retrieval System, to ensure the input volume can produce a result in less than 5 minutes[32], the dataset is generated upon each search command by the user to eliminate unrelated data, keeping the dataset small.

Previously, the Bibliographic Big Data Retrieval System only produces the visualization of the combination of both clustering algorithm, regardless whether it produces the best result or not. An automatic switch between ensembles of clustering methods is proposed as a part of a the system by utilizing a fuzzy inference engine as a decision support tool to select the fastest performing clustering algorithm between self-adapted fuzzy c-means clustering[12], Newman-Girvan clustering algorithm[13], and the combination of both clustering methods[32]. The automatic switch accepts three inputs, which are the number of clusters, number of vertices in each cluster and the time required to complete the clustering process and produces the output in percentage for each clustering result. The clustering algorithms that has the smallest number of clusters and vertices and perform in the fastest time will get highest percentage, and is selected for visualization to the user.

Even though the best way to solve clustering problems is through a mixture of clustering algorithms, a hybrid clustering approach[33-35] requires higher computational complexity yet does not always produce the best result. Therefore the fuzzy inference engine targets to act as an automatic switch that compares the performance of each clustering algorithm where the clustering with the best performance will be selected for visualization to users. It aims to realize the best clustering performance with the reduction of computational complexity from $O(n^3)$ to $O(n)$, if the individual clustering algorithm is selected as the best performing algorithm.

The automatic switch is developed using jFuzzyLogic[36,37], a fuzzy logic controller written in Java and the experiments are carried out in Eclipse IDE 4.2.2[38] using Dell Latitude E5430 laptop with Intel (R) Core (TM) i5-3210M at 2.50GHz. The switch accepts 3 input variables for each clustering result and the values of the input variables from the clustering result are fuzzified according to the membership function, and then evaluated by 27 fuzzy inference rules. The evaluation result is defuzzified to get a crisp percentage output. The percentage output result of the fuzzy inference engine will determine which clustering method will be shown to the users in interactive visualization.

The dataset preparation from the result of three clustering methods are presented in 3.2. The application of fuzzy inference engine as an automatic switch between 3 clustering algorithms using JFuzzyLogic is presented in 3.3. 3.4 discusses the automatic switch experiment on clustering result and 3.5 describes the user feedback evaluation of the system.

3.2 Clustering result of the self-adapted fuzzy c-means, Newman-Girvan algorithm, and ther combination

The Bibliographic Big Data Retrieval System is developed to produce visualization of ensemble clustering result to the users. The systems user interface design is shown in Figure 3.1. A user will enter a keyword in the search box, select a category between paper author, title, year and publication venue, and click the ‘Find’ button. The system will search for related information in the MySQL database, and display the unclustered result on the right panel. Next, the user clicks the ‘Start Clustering’ button and the clustering process will be executed and by utilizing the automatic switch, the clustering with the best performance will be displayed to the

user. To give users more information on each paper, when the user clicks on each vertex, the information of the paper will be displayed in the 'Paper Description' panel on the bottom left of the user interface.

In Figure 3.1, the keyword entered by the user is 'Michael Lindenbaum' for category 'Author'. The result consists of 86 papers that is written by the author. Out of 86 papers, there are 32 connections among the papers. When the node is clicked as shown in the red circle the information regarding the node is displayed in the 'Paper Description' box. Based on the visualization, it can be seen that the node has 4 connections with other nodes. It shows that this paper is one of the author's important paper or key paper as it has many other papers that cites it.

To find relationship among bibliographic big data, the system uses a hybrid of clustering ensembles to improve clustering performance and to overcome the issue of weakness and strength of individual clustering algorithm. There are three types of clustering algorithms used in the system and results from the 3 clustering algorithms from the system is used as input for the automatic switch.

There is no absolute best criterion which would be independent of the final aim of the clustering[39]. To find the best performing clustering algorithm, several criteria have been determined that fulfills the objective of the target application. In the Bibliographic Big Data Retrieval System, to help users find the information they are looking for, the clustering result must be able to converge several important target papers that fulfills the users' search criteria. The time taken to produce the result should be less than 5 minutes[32]. Therefore, three criteria that can determine the desirable clustering performance are the time required to complete the

clustering in seconds, number of related clusters found, and the total number of vertices found in the clusters.

Table 3.1 shows the result for all three clustering algorithms that contains the 3 information used as input variables for the automatic switch.

Table 3.1 Clustering Result From Bibliographic Big Data Retrieval System

	N-G			**SA-FCM**			**COMBINATION**		
Keywords	**Time (s)**	**Clusters**	**Vertices**	**Time (s)**	**Clusters**	**Vertices**	**Time (s)**	**Clusters**	**Vertices**
#1	110	6	24	204	2	9	248	1	5
#2	200	8	25	189	5	12	209	3	7
#3	152	4	8	165	4	12	178	3	7
#4	63	4	16	61	3	11	70	2	9
#5	265	10	58	259	8	36	271	6	32
#6	98	5	31	94	3	18	102	2	12
#7	173	7	26	171	5	18	176	4	14
#8	132	3	10	130	3	9	136	2	7
#9	159	3	11	156	2	9	161	1	4
#10	160	9	19	165	6	16	175	6	14
#11	69	7	22	61	7	19	76	7	13
#12	204	6	21	210	4	17	233	4	12
#13	167	4	10	189	3	8	204	3	6
#14	150	7	12	165	4	11	173	4	9
#15	160	5	16	165	3	14	170	3	11
#16	157	6	26	163	5	24	167	5	20
#17	194	6	33	195	5	31	210	5	25
#18	132	10	45	134	6	22	138	4	18
#19	133	4	15	137	2	8	141	2	6

#20	209	4	20	212	3	15	231	3	11
#21	65	7	21	67	4	17	71	4	11
#22	123	8	32	129	5	27	137	5	13
#23	135	8	22	151	7	21	159	7	20
#24	116	9	36	122	6	31	128	6	22
#25	102	4	21	113	4	19	119	3	14
#26	127	5	28	138	4	25	143	3	11
#27	159	4	11	160	3	10	169	2	7
Avg.	145	6	23	152	4	17	163	4	13

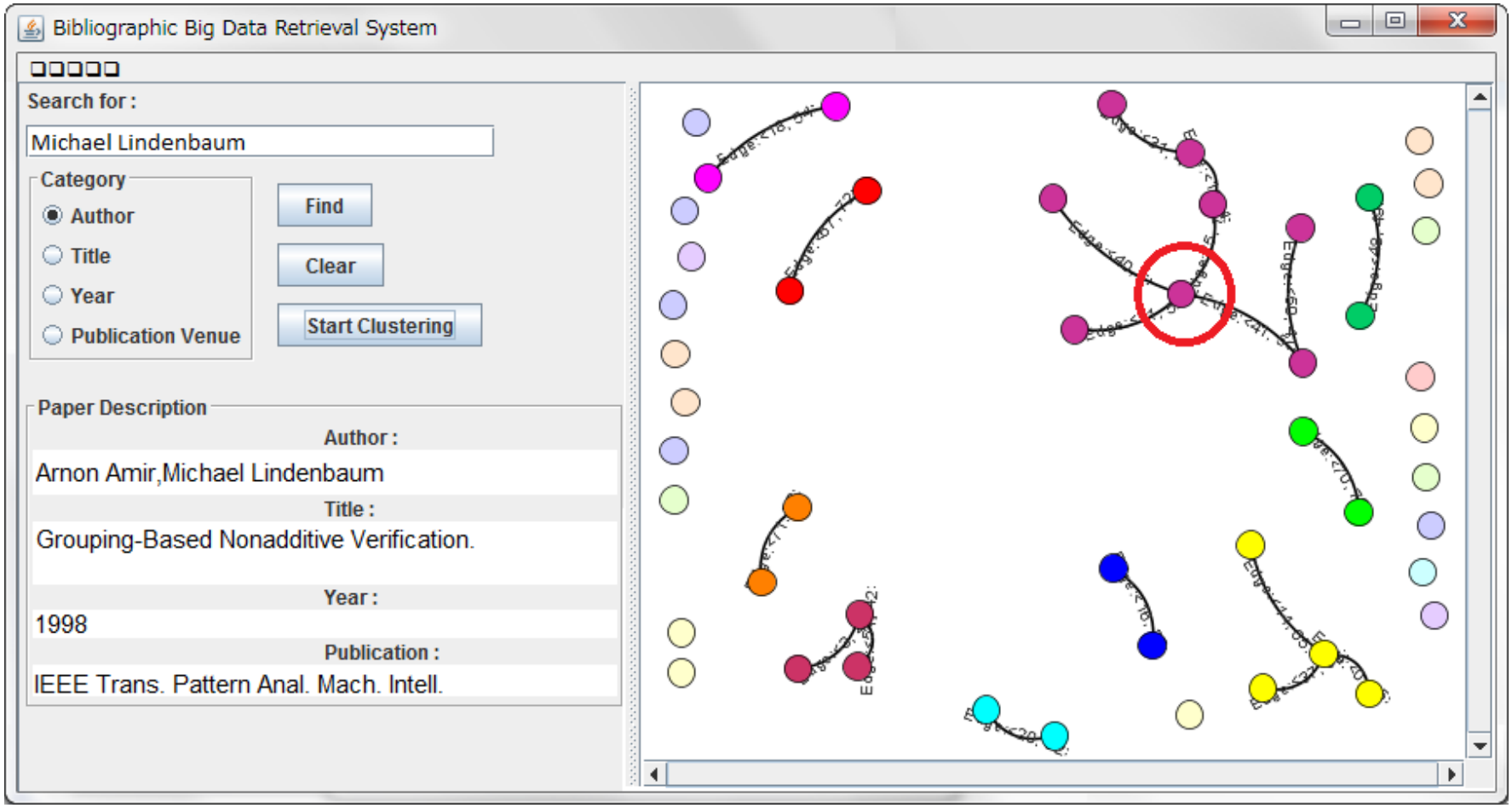


Figure 3.1 Bibliographic Big Data Retrieval System's user interface

3.3 Automatic Switch between 3 clustering algorithms based on Fuzzy Inference

A fuzzy inference engine[40-42] is developed using jFuzzyLogic, a fuzzy logic controller written in Java[36,37]. jFuzzyLogic is chosen to be the fuzzy language controller in the Bibliographic Big Data Retrieval System. jFuzzyLogic follows the standard for Fuzzy Control Language and it provides an application programming interface and an Eclipse[38] plugin that simplifies the writing and testing of the FCL codes. The output generated by the fuzzy inference engine can easily be integrated into the Bibliographic Big Data Retrieval System as both are written in Java language.

Figure 3.2 describes the layout of the engine as an automatic switch. System inputs consists of three input variables, the number of related clusters (1-10 clusters), the total number of vertices in the related clusters (1-100 vertices), and the time required to get the result of the clustering process (1-300 seconds). The input variables are first fuzzified according to the input membership functions, then they will be evaluated by the fuzzy inference rules. Next, they will be defuzzify according to the output membership function that resulted in percentage (0-100%) as the fuzzy inference output.

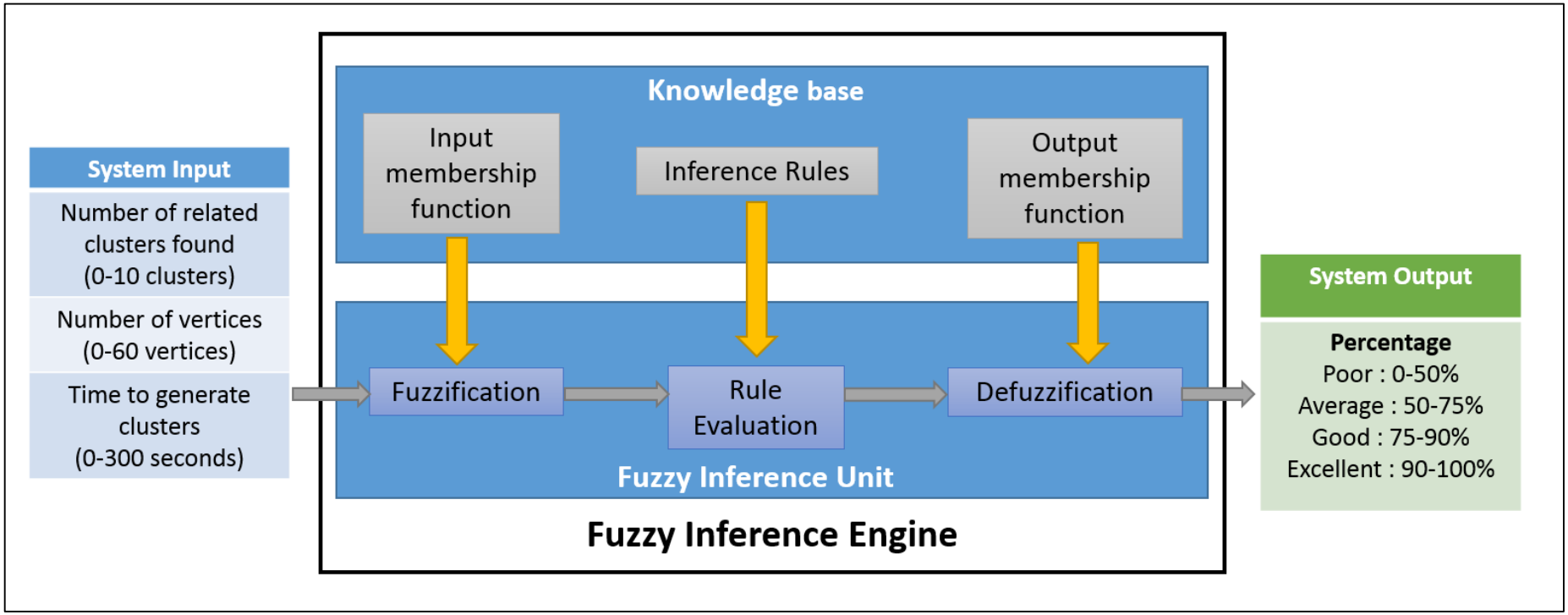


Figure 3.2 The automatic switch layout

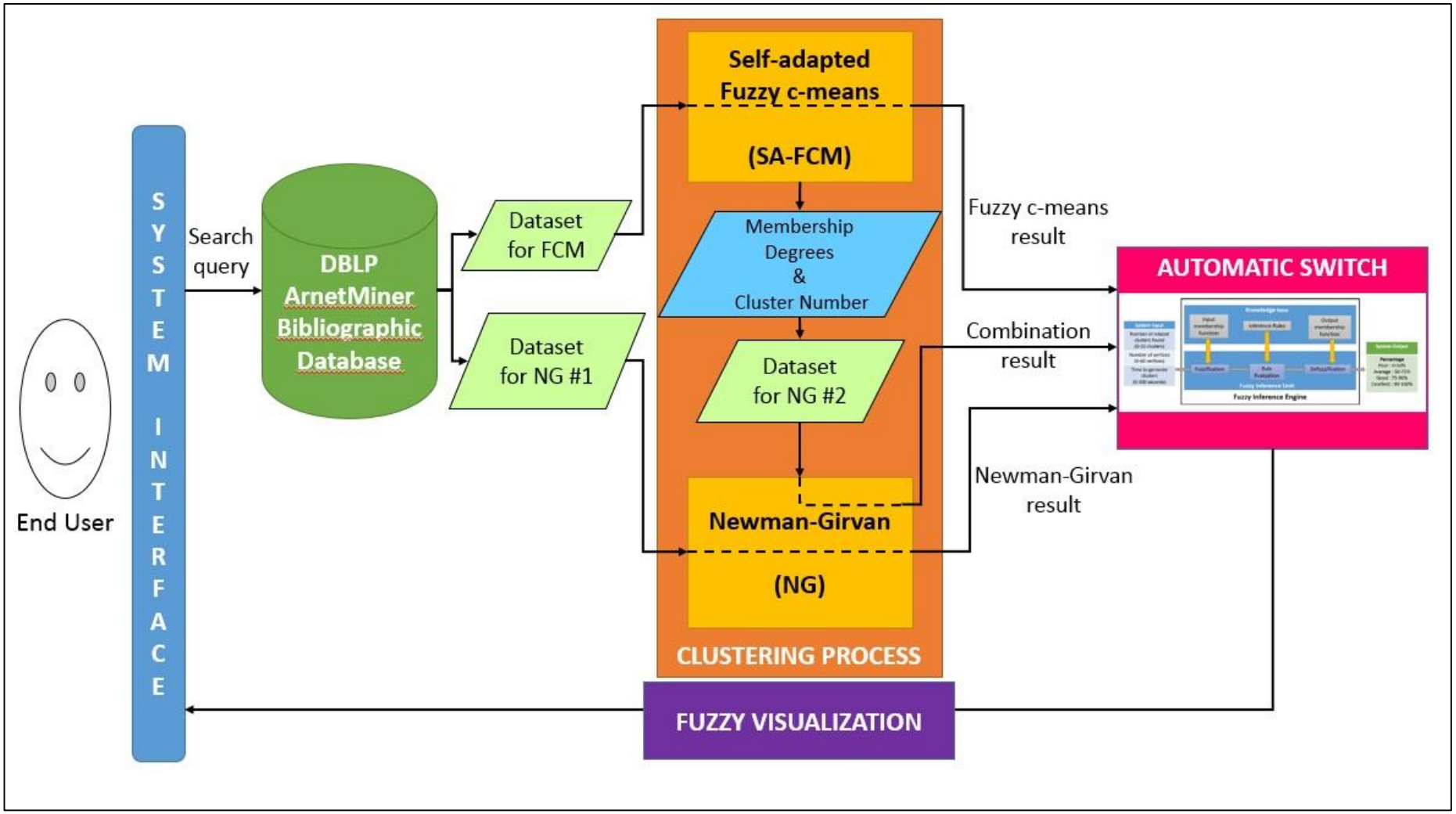


Figure 3.3 Bibliographic Big Data Retrieval System architecture with automatic switch function.

The system architecture of the Bibliographic Big Data Retrieval System is shown in Figure 3.3, where the automatic switch is situated between the clustering processes and the fuzzy visualization process. It shows that the automatic switch plays an important role to decide which clustering algorithms should be used in the interactive visualization to the user.

The fuzzification of the number of clusters is divided into 3 categories. Number of clusters is considered low if the amount is between 1-5 clusters, medium if it is between 1-10 clusters and high if 5-10 clusters are found. The fuzzification code for the fuzzy control language is described in Figure 3.4.

```
FUZZIFY no_of_clusters
  TERM low  := (0, 1) (5, 0) ;
  TERM med  := (0, 0) (5,1) (10,0);
  TERM high := (5, 0) (10, 1);
END_FUZZIFY
```

Figure 3.4 Fuzzification of total number of clusters

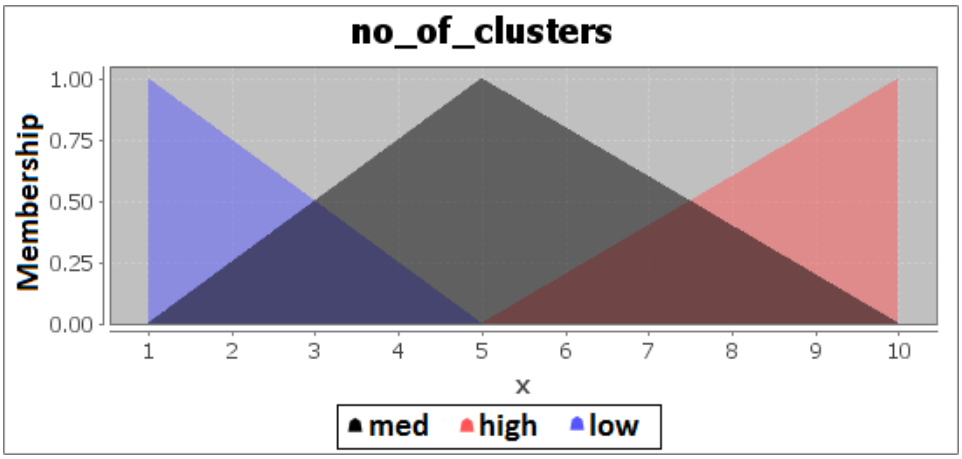


Figure 3.5 Membership degrees graph for total number of clusters

The membership degrees graph of the number of clusters is shown in Figure 3.5. The fuzzification of the total number of vertices is also divided into 3 categories. The number of

vertices found is considered few if the amount is between 1-25 vertices, several if the amount is between 1-60 vertices, and many if the amount is between 25-60 vertices. The fuzzification code for number of vertices is shown in Figure 3.6 and the membership degrees graph for vertices is shown is Figure 3.7.

```
FUZZIFY no_of_vertices
  TERM few := (0,1) (25, 0) ;
  TERM several := (0,0)(25,1) (60,0);
  TERM many := (25,0) (60,1);
END_FUZZIFY
```

Figure 3.6 Fuzzification of total number of cluster

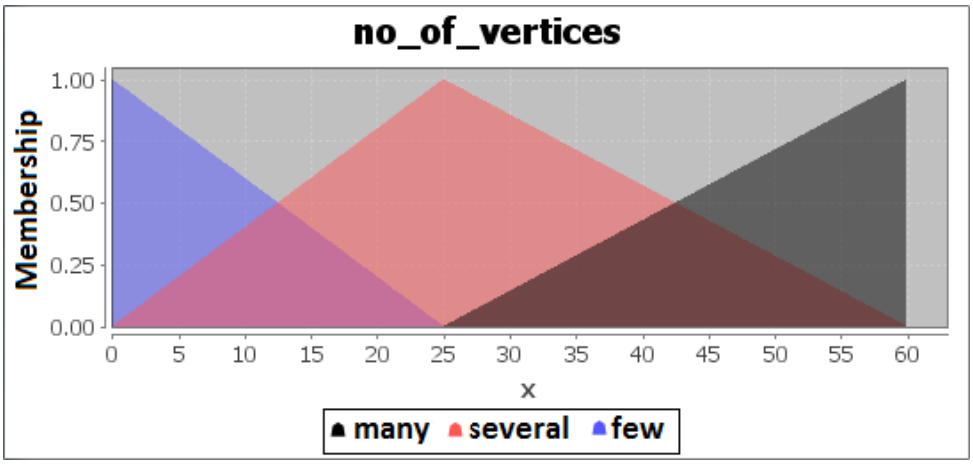


Figure 3.7 Membership degrees graph for total number of vertices

For time required to complete the clustering process, there are also 3 categories to represent them. Time is considered short if it takes between 1 to 200 seconds to complete the process, medium if it is between 1-300 seconds, and it is considered long if it takes between 200-300 seconds to complete the clustering process. The fuzzification code for time is shown in Figure 3.8 and the membership graph is shown in Figure 3.9.

```
FUZZIFY time_sec
  TERM short:=(1,1) (200,0) ;
  TERM medium:=(1,0) (200,1) (300,0);
  TERM long:=(200, 0) (300, 1);
END_FUZZIFY
```

Figure 3.8 Fuzzification of time in seconds

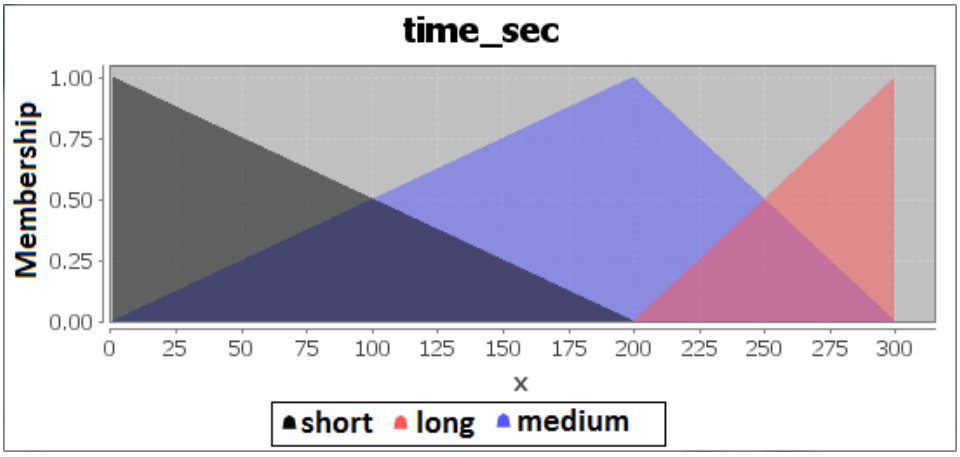


Figure 3.9 Membership degrees graph for time in seconds

The defuzzification process are performed to get a non-fuzzy value that best represents the possibility distribution of an inferred fuzzy control action. The defuzzified output will be in percentage, where the clustering result with the highest percentage will be selected for visualization to the end user. There are four categories for the percentage output, ranging from poor, average, good to excellent. Poor percentage ranges from 0-50%, average percentage ranges from 10-85%, good percentage ranges from 50-100% and excellent percentage ranges from 85-100%. The defuzzification code for percentage control is shown in Figure 3.10 and the membership degree graph for percentage is shown in Figure 3.11.

```
DEFUZZIFY percentage
 TERM poor := (0,1) (10,1) (50,0);
 TERM satisfactory:=(10,0)(50,1)(85,0);
```

```
 TERM good := (50,0)  (85,1) (100,0);
 TERM excellent := (85,0) (100,1);
END DEFUZZIFY
```

Figure 3.10 Defuzzification for percentage output and defuzzification method specification

Figure 3.11 Membership degrees graph for percentage

The defuzzification strategy used in the proposed method is the center of gravity or centroid method. The strategy is the most common and physically appealing of all the defuzzification methods[43,44] It is given by the algebraic expression where ∫ denotes an algebraic integration, as shown in

$$\int x\mu(x)dx / \int \mu(x)dx . \qquad (1)$$

There is no systematic procedure for choosing a good defuzzification strategy and it depends on the properties of the application. Therefore, the center of gravity strategy is chosen due to its computational simplicity where it does not require complex computation that may lead to more time. Since the end result which is the visualization of bibliographic data search result needs to be produced in less than 5 minutes, this is an important criteria to keep the calculation process time as short as possible.

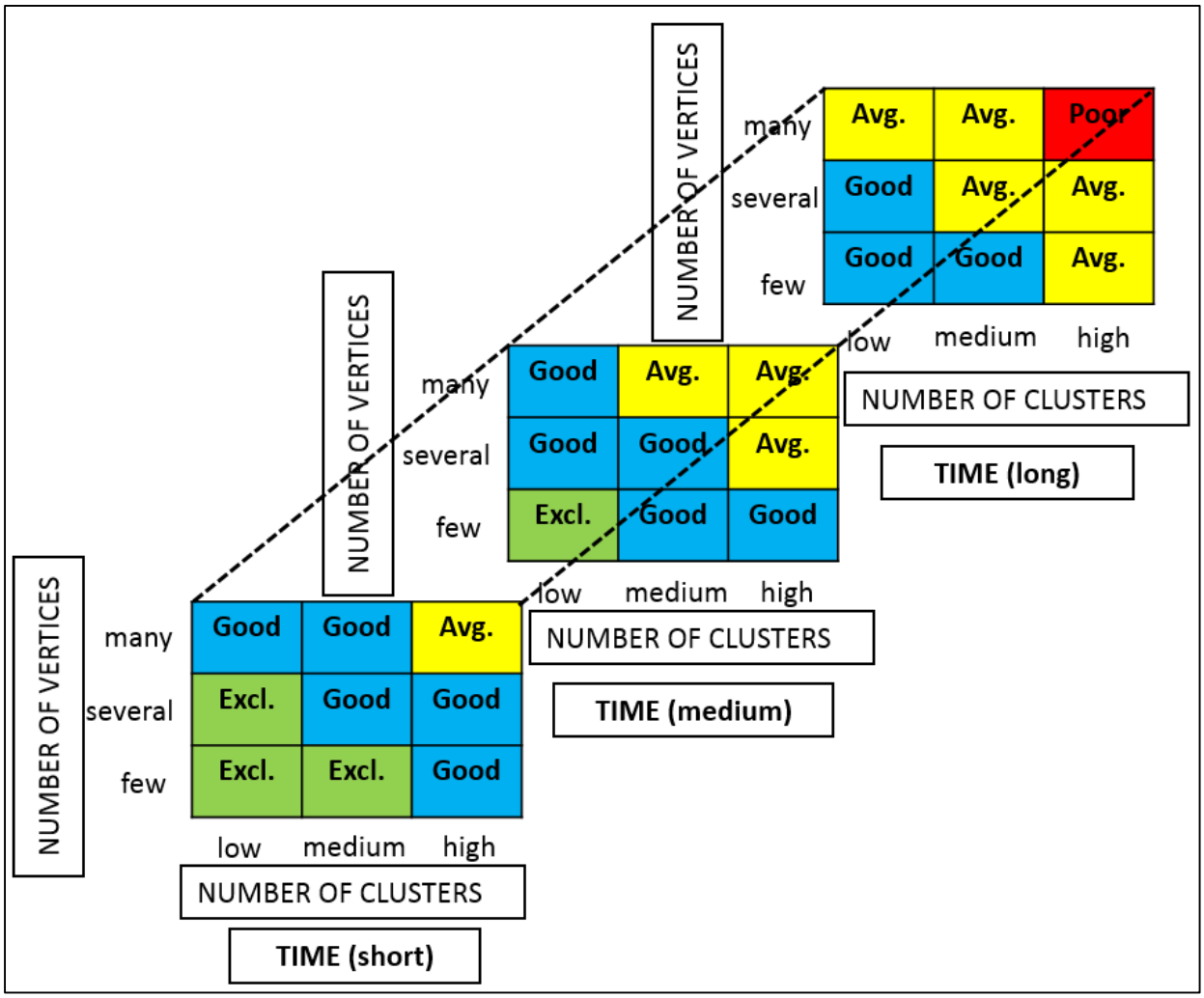


Figure 3.12 Fuzzy associative matrix for percentage control

A fuzzy associative matrix is developed to cover every possible outcome from the clustering result. Figure 3.12 shows the fuzzy associative matrix for the clustering results with three inputs, number of clusters, number of vertices and time. Inside each box is written a label of the automatic switch output. In the automatic switch, there are 27 possible rules corresponding to the 27 boxes in the matrix.

Figure 3.13 describes the rule block that consists 27 inference rules for the automatic switch in fuzzy language control codes.

```
RULEBLOCK No1
      AND : MIN;  // Use 'min' for 'and' (DeMorgan's Law)
      ACT : MIN;  // Use 'min' activation method
      ACCU : MAX;  // Use 'max' accumulation method
RULE 1  : IF ((no_of_clusters IS low) AND (no_of_vertices IS few)) AND
(time_sec IS short) THEN percentage IS excellent;

RULE 2  : IF ((no_of_clusters IS low) AND (no_of_vertices IS few)) AND
(time_sec IS medium) THEN percentage IS excellent;

RULE 3  : IF ((no_of_clusters IS low) AND (no_of_vertices IS few)) AND
(time_sec IS long) THEN percentage IS excellent;

RULE 4  : IF ((no_of_clusters IS low) AND (no_of_vertices IS several))
AND (time_sec IS short) THEN percentage IS excellent;

RULE 5  : IF ((no_of_clusters IS low) AND (no_of_vertices IS several))
AND (time_sec IS medium) THEN percentage IS good;

RULE 6  : IF ((no_of_clusters IS low) AND (no_of_vertices IS several))
AND (time_sec IS long) THEN percentage IS good;

RULE 7  : IF ((no_of_clusters IS low) AND (no_of_vertices IS many)) AND
(time_sec IS short) THEN percentage IS good;

RULE 8  : IF ((no_of_clusters IS low) AND (no_of_vertices IS many)) AND
(time_sec IS medium) THEN percentage IS good;

RULE 9  : IF ((no_of_clusters IS low) AND (no_of_vertices IS many)) AND
(time_sec IS long) THEN percentage IS satisfactory;

RULE 10 : IF ((no_of_clusters IS med) AND (no_of_vertices IS few)) AND
(time_sec IS short) THEN percentage IS excellent;

RULE 11 : IF ((no_of_clusters IS med) AND (no_of_vertices IS few)) AND
(time_sec IS medium) THEN percentage IS good;

RULE 12 : IF ((no_of_clusters IS med) AND (no_of_vertices IS few)) AND
(time_sec IS long) THEN percentage IS good;

RULE 13 : IF ((no_of_clusters IS med) AND (no_of_vertices IS several))
AND (time_sec IS short) THEN percentage IS good;

RULE 14 : IF ((no_of_clusters IS med) AND (no_of_vertices IS several))
AND (time_sec IS medium) THEN percentage IS good;
```

```
RULE 15 : IF ((no_of_clusters IS med) AND (no_of_vertices IS several))
AND (time_sec IS long) THEN percentage IS satisfactory;

RULE 16 : IF ((no_of_clusters IS med) AND (no_of_vertices IS many)) AND
(time_sec IS short) THEN percentage IS good;

RULE 17 : IF ((no_of_clusters IS med) AND (no_of_vertices IS many)) AND
(time_sec IS medium) THEN percentage IS satisfactory;

RULE 18 : IF ((no_of_clusters IS med) AND (no_of_vertices IS many)) AND
(time_sec IS long) THEN percentage IS satisfactory;

RULE 19 : IF ((no_of_clusters IS high) AND (no_of_vertices IS few)) AND
(time_sec IS short) THEN percentage IS good;

RULE 20 : IF ((no_of_clusters IS high) AND (no_of_vertices IS few)) AND
(time_sec IS medium) THEN percentage IS good;

RULE 21 : IF ((no_of_clusters IS high) AND (no_of_vertices IS few)) AND
(time_sec IS long) THEN percentage IS satisfactory;

RULE 22 : IF ((no_of_clusters IS high) AND (no_of_vertices IS several))
AND (time_sec IS short) THEN percentage IS good;

RULE 23 : IF ((no_of_clusters IS high) AND (no_of_vertices IS several))
AND (time_sec IS medium) THEN percentage IS satisfactory;

RULE 24 : IF ((no_of_clusters IS high) AND (no_of_vertices IS several))
AND (time_sec IS long) THEN percentage IS satisfactory;

RULE 25 : IF ((no_of_clusters IS high) AND (no_of_vertices IS many)) AND
(time_sec IS short) THEN percentage IS satisfactory;

RULE 26 : IF ((no_of_clusters IS high) AND (no_of_vertices IS many)) AND
(time_sec IS medium) THEN percentage IS satisfactory;

RULE 27 : IF ((no_of_clusters IS high) AND (no_of_vertices IS many)) AND
(time_sec IS long) THEN percentage IS poor;

END_RULEBLOCK
```

Figure 3.13 Rule block for inference rules of the automatic switch

3.4 Experiment of Automatic Switching on Clustering Results

Table 3.2 AutomaticSwitch Result – Percentage Output

	Percentage (%)		
Keyword	**Newman-Girvan**	**Self-adapted fuzzy c-means**	**Combination**
#1	77.56	76.04	68.99
#2	62.39	77.94	72.07
#3	78.74	78.65	79.48
#4	78.74	79.42	80.16
#5	33.51	50.42	53.54
#6	66.41	79.21	79.09
#7	76.11	78.41	78.72
#8	79.48	79.48	80.93
#9	79.40	80.10	83.99
#10	54.70	77.91	77.59
#11	77.84	78.24	78.94
#12	75.40	70.89	62.26
#13	78.74	79.48	75.75
#14	77.79	78.72	78.74
#15	78.26	78.78	79.40
#16	76.04	78.14	78.17
#17	65.99	67.72	71.12
#18	54.88	77.93	78.18
#19	78.60	80.58	81.05
#20	70.98	70.22	62.64
#21	77.98	78.74	79.40
#22	61.88	73.92	78.07
#23	62.50	78.18	78.35
#24	56.15	67.01	77.84
#25	78.63	78.73	78.78

#26	72.05	78.18	79.40
#27	78.72	79.48	81.09
Mean	**70.72**	**76.02**	**76.06**
Std. Dev.	**11.15**	**6.31**	**6.88**

The percentage result from the automatic switch is shown in Table 3.2. Out of the three clustering result, the combination of both clustering algorithms has the highest mean percentage of 76.06%, as opposed to 76.02% for self-adapted fuzzy c-means algorithm and 70.72% for Newman-Girvan algorithm. It shows that on average, the combination always perform better than the individual clustering algorithms. This can be seen from the average number of clusters and vertices produced by each clustering algorithm as shown on Table 3.1, where the combination always generates the least number of clusters and vertices as opposed to the individual clustering algorithms, in under 5 minutes.

The self-adapted fuzzy c-means has the smallest standard deviation of 6.31% which means that its performance does not vary greatly as opposed to the Newman-Girvan and the combination algorithm. This is because the self-adapted fuzzy c-means is more flexible than the crisp Newman-Girvan algorithm due to its fuzzy properties. Newman-Girvan algorithm has the highest standard deviation of 11.151%. The average percentage for Newman-Girvan is 70.72%, but it performs poorly for keyword #5 with 33.51%. For this keyword, Newman-Girvan takes 265 seconds to find 58 vertices and 10 clusters. Based on the fuzzy inference rules of the automatic switch, a low percentage is given to the Newman-Girvan algorithm as it does not favor the users of the system who requires more time to examine each of the 58 vertices for their desired papers. The combination sits in the middle with a standard deviation of 6.88%. The performance of the combination varies greater than the self-adapted fuzzy c-means because

it combines both clustering algorithms therefore also inherits their less than efficient performance.

Based on the bar chart for percentage shown in Figure 3.14, the combination algorithm performance shown in grey line performs better in most search cases. It received the highest percentage in 20 out of 27 search cases with the highest percentage of 83.99% for keyword #9. For keyword #9, the combination algorithm successfully gathered 4 target papers in one cluster in 161 seconds. The few number of papers, and the short time to produce the result gives the combination algorithm the highest percentage as compared to the other 2 clustering algorithms. The combination algorithm do not get the highest percentage in 7 cases (keywords #1, #2, #6, #10, #12, #13, and #20) as the keywords return a larger search result compared to the other keywords thus more time are required to perform the clustering process on them. According to the fuzzy inference rules, if the time taken to produce the clustering result is long (200-300 seconds), the percentage is will not be in 'Excellent' category. Therefore, the application of the automatic switch will be able to compare the performance and select the best performing algorithm each time a search is performed.

The Newman-Girvan algorithm's computable complexity is O(n) and the self-adapted fuzzy c-means algorithm has a computable complexity of $O(n^3)$. When combined, the complexity of the algorithms becomes $O(n^3)$, at worst. Therefore, by applying the automatic switch to compare the performance of all three algorithms, if the Newman-Girvan algorithm is selected as the best performing algorithm, the computable complexity to produce the visualization of the clustering result is reduced from $O(n^3)$ to O(n).

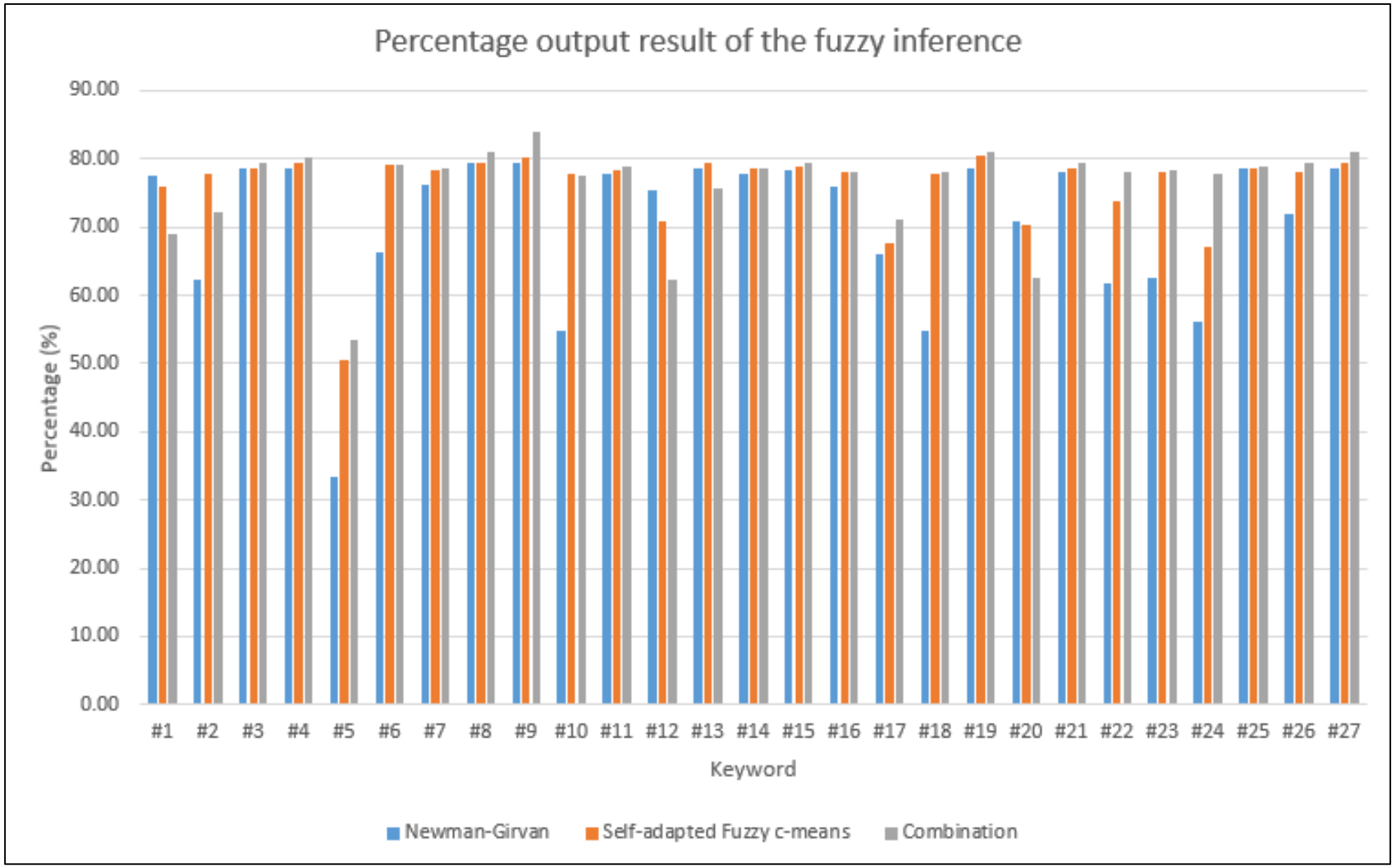


Figure 3.14 Bar chart for percentage output result of the fuzzy inference

3.5 User-based Evaluation for the Bibliographic Big Data Retrieval System

To evaluate the system's usability, user-based evaluation method is used where the system is tested by selecting 18 participants to perform a set of pre-determined tasks on the system prototype. A feedback questionnaire is given to the participants after the tasks have been performed on the system prototype and the participants have to fill in the questionnaire based on their opinion of the system. This method is used as it is the most realistic estimate of usability[45].

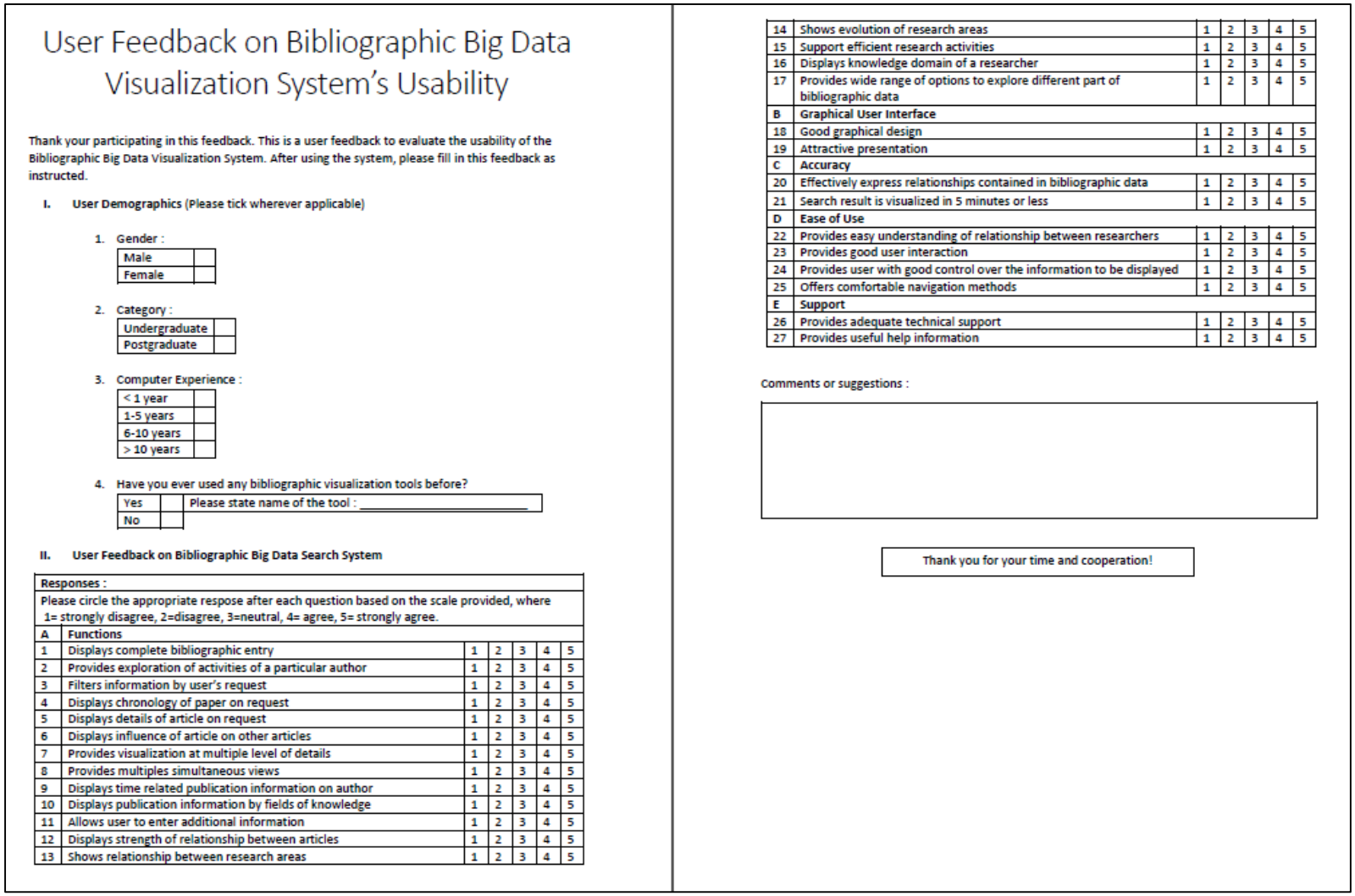
A 16-question feedback questionnaires focusing on bibliographic visualization tool objectives is designed to evaluate the usability of the system. There are 4 categories covered in the questionnaire, which are the content organization, navigation, graphical user interface, and effectiveness and performance of the system.

Questions for evaluating the system's usability are:

1. Content Organization
    a. Displays complete bibliographic entry
    b. Displays chronology of paper on request
    c. Displays influence of article on other articles
    d. Displays publication information by fields of knowledge
    e. Displays strength of relationship between articles
    f. Shows relationship between research areas

2. Navigation
    a. Provides exploration of activities of a particular author
    b. Filters information by user's request
    c. Offers comfortable navigation methods
    d. Provides wide range of options to explore different part of bibliographic data
3. Effectiveness and performance
    a. Effectively express relationships contained in bibliographic data
    b. Search result is visualized in 5 minutes or less
    c. Provides easy understanding of relationship between researchers
    d. Provides user with good control over the information to be displayed
4. Graphical User Interface
    a. Good graphical design
    b. Attractive presentation

Five options are given to the participants for each question. The participants will choose one option from 1=Highly Dissatisfied, 2=Dissatisfied, 3=Neutral, 4=Satisfied, to 5=Highly Satisfied. The screenshots of the feedback quesitonnaire are shown in Figure 3.15.



User Feedback on Bibliographic Big Data Visualization System's Usability

Thank your participating in this feedback. This is a user feedback to evaluate the usability of the Bibliographic Big Data Visualization System. After using the system, please fill in this feedback as instructed.

I. User Demographics (Please tick wherever applicable)

1. Gender :

Male	
Female	

2. Category :

Undergraduate	
Postgraduate	

3. Computer Experience :

< 1 year	
1-5 years	
6-10 years	
> 10 years	

4. Have you ever used any bibliographic visualization tools before?

Yes		Please state name of the tool : _____
No		

II. User Feedback on Bibliographic Big Data Search System

Responses :							
Please circle the appropriate respose after each question based on the scale provided, where 1= strongly disagree, 2=disagree, 3=neutral, 4= agree, 5= strongly agree.							
A	**Functions**						
1	Displays complete bibliographic entry	1	2	3	4	5	
2	Provides exploration of activities of a particular author	1	2	3	4	5	
3	Filters information by user's request	1	2	3	4	5	
4	Displays chronology of paper on request	1	2	3	4	5	
5	Displays details of article on request	1	2	3	4	5	
6	Displays influence of article on other articles	1	2	3	4	5	
7	Provides visualization at multiple level of details	1	2	3	4	5	
8	Provides multiples simultaneous views	1	2	3	4	5	
9	Displays time related publication information on author	1	2	3	4	5	
10	Displays publication information by fields of knowledge	1	2	3	4	5	
11	Allows user to enter additional information	1	2	3	4	5	
12	Displays strength of relationship between articles	1	2	3	4	5	
13	Shows relationship between research areas	1	2	3	4	5	
14	Shows evolution of research areas	1	2	3	4	5	
15	Support efficient research activities	1	2	3	4	5	
16	Displays knowledge domain of a researcher	1	2	3	4	5	
17	Provides wide range of options to explore different part of bibliographic data	1	2	3	4	5	
B	**Graphical User Interface**						
18	Good graphical design	1	2	3	4	5	
19	Attractive presentation	1	2	3	4	5	
C	**Accuracy**						
20	Effectively express relationships contained in bibliographic data	1	2	3	4	5	
21	Search result is visualized in 5 minutes or less	1	2	3	4	5	
D	**Ease of Use**						
22	Provides easy understanding of relationship between researchers	1	2	3	4	5	
23	Provides good user interaction	1	2	3	4	5	
24	Provides user with good control over the information to be displayed	1	2	3	4	5	
25	Offers comfortable navigation methods	1	2	3	4	5	
E	**Support**						
26	Provides adequate technical support	1	2	3	4	5	
27	Provides useful help information	1	2	3	4	5	

Comments or suggestions :

Thank you for your time and cooperation!

Figure 3.15 User feedback questionnaire

The feedback result is analyzed using WEBUSE[46], a website usability evaluation tool. The answer options and their corresponding merits are shown in Table 3.3.

Usability point for a category, x, is defined in

$$x = (\sum m) / q \,, \qquad (2)$$

where m is the merit for each question and q is the number of question for each category.

Table 3.3 Answer Option For Feedback Questionnaire And Corresponding Merits

Option	**Merit**
Highly Dissatisfied	1.00
Dissatisfied	0.75
Neutral	0.50
Satisfied	0.25
Highly Satisfied	0.00

Table 3.4 shows the usability levels and the corresponding usability points.

Table 3.4 Usability points and corresponding usability levels

Points, x	**Usability Level**
0<=x<=0.2	Bad
0.2<x<=0.4	Poor
0.4<x<=0.6	Moderate
0.6<x<=0.8	Good
x 0.8<x<=1.0	Excellent

A total number of 18 participants have volunteered to test the prototype of the Bibliographic Big Data Retrieval System where 44.4% are males and 55.6% are female participants. 22.2% of the participants are undergraduate students and 77.8% are postgraduate students with 38.9%

have between 6 and 10 years of computer experience, and 61.1% have more than 10 years of computer experience. Only 16.7% of the participants have used another bibliographic visualization tool before taking part in this survey.

Table 3.5 System evaluation result by participants

	Scale Level(%)				
Usability Category	**Highly Dissatisfied**	**Dissatisfied**	**Neutral**	**Satisfied**	**Highly Satisfied**
Content Organization	3.7	6.4	21.3	48.1	20.4
Navigation	2.8	8.3	5.5	47.2	36.1
User Interface	5.5	11.1	19.4	37.5	26.4
Performance & Effectiveness	0	2.8	30.5	36.1	30.5
Total	3.4	7.6	18.1	43.8	27.1

Table 3.5 shows a summary of user satisfaction level in 4 categories of usability from content organization, navigation, user interface design, and performance and effectiveness. From the satisfaction scale level, users are most satisfied in the navigation aspect of the system with 36.1% of participants highly satisfied with the navigation styles that the system offers.

The bar chart for the user satisfaction level is shown in Figure 3.16. It shows that the user satisfaction level mainly falls in the ‘Satisfied’ category as indicated in yellow.

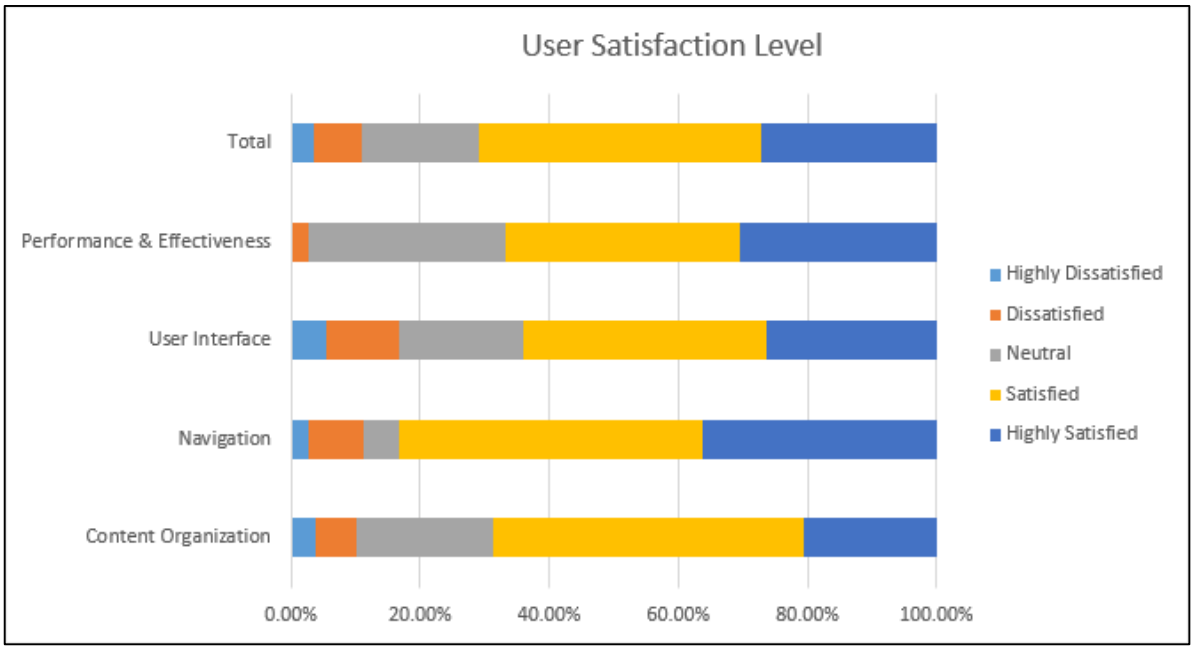


Figure 3.16 Bar chart for User Satisfaction Level

Table 3.6 Usability level and usability points for each category

Usability Category	**Usability Point**	**Usability Level**
Content Organization	0.6875	Good
Navigation	0.7639	Good
User Interface Design	0.6701	Good
Performance & Effectiveness	0.7361	Good
Total	0.6875	Good

Table 3.6 shows a summary of usability point and usability level based on each category. From the overall usability level, it can be concluded that the system is accepted as "Good" based on the usability scale. System's navigation and performance receives a high point of 0.7639 and 0.7361 respectively. It shows that the users are satisfied with the system's navigation and performance but improvements are necessary to increase the usability level from 'Good' to 'Excellent'. The system's content organization and user interface design receives usability points less than 0.7 therefore even though the user is somewhat satisfied with these

aspects of the system, more improvements are needed to be done to increase users' satisfactions on the usability of the system as a whole.

Some comments given by participants to improve the content organization is that the system should not only give the best clustering performance but also shows the number of citation of evey authors or papers selected by users. It is an important information that gives extra weight to every author and paper nodes that they wish to explore.

Other comments state that the system should offer the users an option to control the importance of their search, whether they want only highly related information to be shown or to include a bigger search result that shows all related results, whether they are highly related or somewhat related to the input information.

Obtaining multiple levels of useful information from large amounts of data requires scalable algorithms to produce timely results[47]. Current algorithms are inefficient in terms of big data analysis. the bigger the data gets the less efficient each algorithm will perform, and leads to higher computational complexity.

To increase the performance of clustering algorithms to process big data, efficient tools and technologies are essential to process such data. Currently the dataset used in the Bibliographic Big Data Retrieval System is stored in MySQL 5.6, an open source relational database management system. The issue with current database while it is suitable for research purposes, to ensure faster and more efficient service, a database specifically designed for big data is preferable.

3.6 Chapter Summary

The three criteria that determine the desirable clustering performance in the Bibliographic Big Data retrieval System are the time required to complete the clustering in seconds, number of related clusters found, and the total number of vertices found in the clusters. The automatic switch accepts these three criteria as its input and the experiment is carried out in Eclipse IDE 4.2.2, connected to MySQL 5.6 database that stores the bibliographic big data, using Dell Latitude E5430 laptop with Intel (R) Core (TM) i5-3210M at 2.50GHz. The experimental result demonstrates that the combination of both clustering algorithms is selected as the best performing algorithm in 20 out of 27 cases with the highest percentage of 83.99%, completed the process in 161 seconds. The self-adapted fuzzy c-means is selected as the best performing clustering algorithm in 4 search cases with the highest percentage at 80.58%, completed in 137 seconds and Newman-Girvan algorithm is selected in 3 search cases with the highest percentage at 79.46% in 132 seconds. By applying the automatic switch in the Bibliographic Big Data Retrieval System, the best performing algorithm can be determined in every search case executed by the users. The computable complexity of the self-adapted fuzzy c-means and the combination algorithm is $O(n^3)$, while Newman-Girvan is O(n). For every search cases that Newman-Girvan is selected as the best performing algorithm, the computational complexity of the clustering process is reduced as it will only produce the visualization result of Newman-Girvan clustering result to the users. The feedback survey shows the overall level of system usability is good and acceptable to users. Users are satisfied with the navigation, effectiveness, and performance of the system but some improvements needs to be done to increase the system's usability especially in terms of content organization and user interface design.

The automatic switch is incorporated into the Bibliographic Big Data Retrieval System that focuses on visualization of fuzzy relationship using hybrid approach combining the self-adapted fuzzy c-means and Newman-Girvan algorithm. The system is currently being developed and improved. Future works includes: i) emphasizing the visualization of fuzzy relationship to differentiate from crisp clustering relationship to effectively display the fuzzy visualization result to the users; and ii) planning to be released to the public through the Internet.

Chapter 4

Fuzzy Ontological Approach in Keyword-Based Retrieval for Bibliographic Big Data Retrieval System

4.1 Introduction

A bibliographic big data retrieval system is a system that aims to help users to automatically access information regarding bibliographic data in a specific research domain. The large volume of bibliographic data that exists today poses a challenge for these users to search for information that they need effectively. With existing technologies, gathering and storing big data knowledge is relatively easy. But effective and efficient retrieval process still has a large room for improvements. Retrieval process is still often based on keyword input information entered by used. Many users have difficulties in identifying correct search terms and searches are often unsuccessful.

There are two ways that a data retrieval system can use to find the set of most relevant documents that matches the users keyword input information. They are keyword-based approach and concept based approach. In keyword based approach, data is retrieved if they match the keyword specified in the search query. The challenge of this approach is that it only works well if the users know exactly what they want and are able to pick the right keywords[48]. Meanwhile in concept based approach, data are retrieved according to their relevance to the

keyword input information. It is a domain specific approach that is able to improve the manageability of data resources through the application of ontologies.

There are several existing fuzzy ontology information retrieval method that exists, such as the Leite model[49] that semantically retrieves a set of query's relevant documents in multiple domains. Each domain is represented as a fuzzy ontology and connected to other domains using positive relations. When a certain user enters a query, Leite expands it using a two phases query expansion process. The first phase expands each concept in the query with all of its related concepts in other domains. Then the result enters the second phase to expand each concept in it with all of its related concepts in the same domain. The max product composition between each document and the expanded user query is used as the similarity function to determine a set of the most relevant documents. This set of relevant documents is ranked in a descending order according to their relevance degree and returned to the user. A fuzzy ontology-based document retrieval model called Fuzzy Relational Ontology Model, FROM[50] semantically retrieves a set of relevant documents based on a user query. It assumes that each document in the document collection is already annotated with a set of weighted keywords. It considers fuzzy ontology as a set of concepts, terms, and relations between concepts and terms. FROM deals with crisp queries. When a user enters his query, it expands each concept in it with all terms that describes it and each term in it with all concepts that it describes. It retrieves a set of relevant documents using the max min composition between each document in the document collection and the expanded user query. The resulted set is ranked in a descending order according to each document relevance degree and then it returned to the user.

An ontology based information retrieval model[51] is proposed to deal with open environment by annotating the document collection using two techniques, an NLP based and a context semantic information based. The model performs some processing on it using the

ontology-based Question Answering (QA) system, PowerAqua whenever a user enters a query. The adaptation of the traditional vector space IR model is used as to calculate the relevance degree of each document in the document collection with respect to the entered user query. Documents are returned to the user such that documents with higher relevance degree are listed first.

The issue arising from the three described methods above is that it has a low recall measure, as a result of using incomplete fuzzy ontology components for expanding a certain user query keywords. To rank the resulted documents, these models use the similarity degree between each document in the document collection and the user query keywords.

In this chapter, a fuzzy ontology based knowledge reasoning for bibliographic big data retrieval is proposed that combines fuzzy logic and descriptive logic to represent overlapping and imprecise keywords in bibliographic big data retrieval. It aims to increase the f-measure of the bibliographic big data retrieval system by reflecting the in-depth relationship between concepts and terms in computer science domain.

In 4.2, fuzzy ontology for bibliographic big data is discussed. Protege and its Fuzzy2Owl plug-in to deal with fuzzy ontology is described in 4.3. The proposed fuzzy ontology knowledge reasoning framework is explained in 4.4 and the experimental result of fuzzy ontologies from user queries on bibliographic big data is discussed and evaluated in 4.5. The proposed method is concluded in 4.6.

4.2 Fuzzy Ontology for Bibliographic Big Data

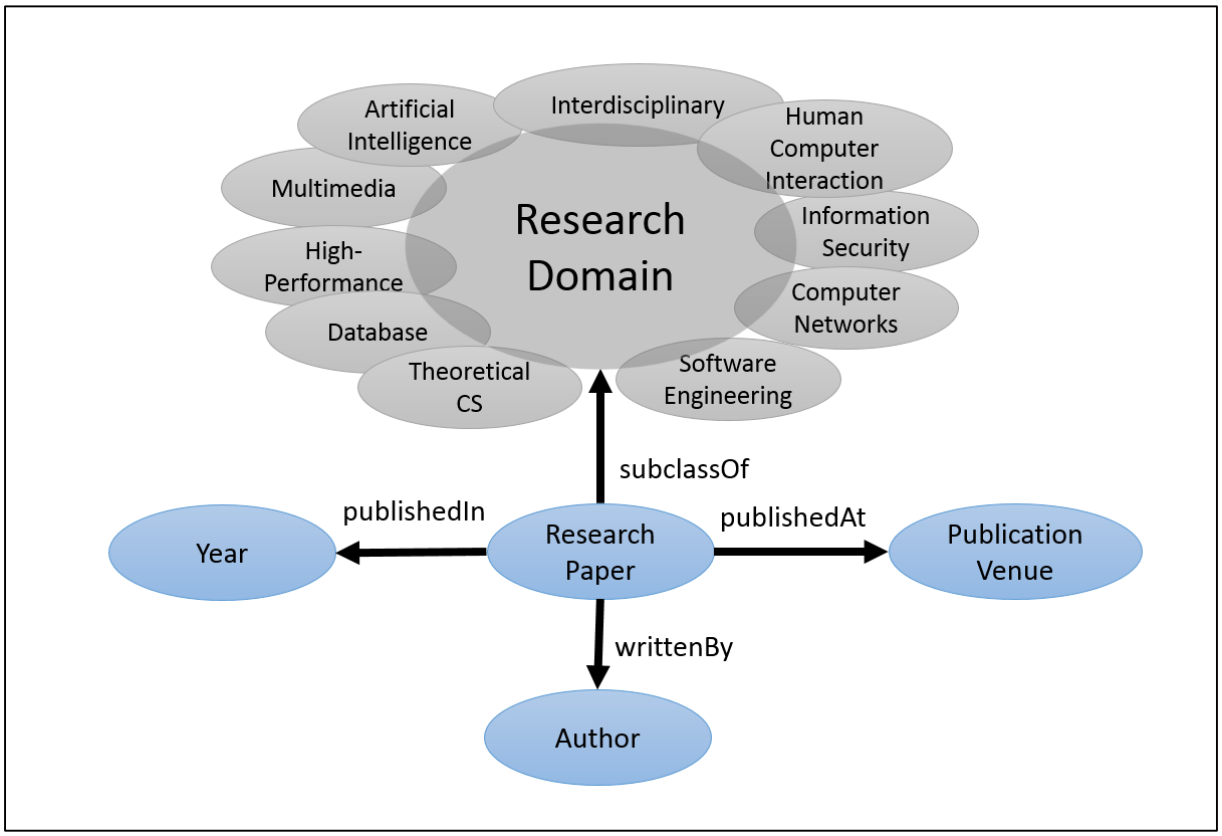


Figure 4.1 Computer science domain for fuzzy ontology

Classical ontology languages are not appropriate to deal with imprecision or vagueness in knowledge. Therefore, description logics, or DLs[52] for the semantic web has been enhanced by various approaches to handle probabilistic & possibilistic uncertainty, and vagueness. In description logics complex expressions are defined with logic-based constructors and the semantics that can be built with valid expressions determines the complexity of the description model.

To realize the concept-based approach in the computer science field, 10 domains has been determined based on the major branches in the field. They are Software Engineering, Computer Networks, Information Security, Human-Computer Interaction, Interdisciplinary,

Artificial Intelligence, Multimedia, High-performance Computing, Database, and Theoretical Computer Science. Based on these 10 domains, input information given by users can be evaluated based on the concept of the search, not only the keyword.

4.3 A Semantic Tool of Keyword-Based Retrieval for Bibliographic Big Data Retrieval System

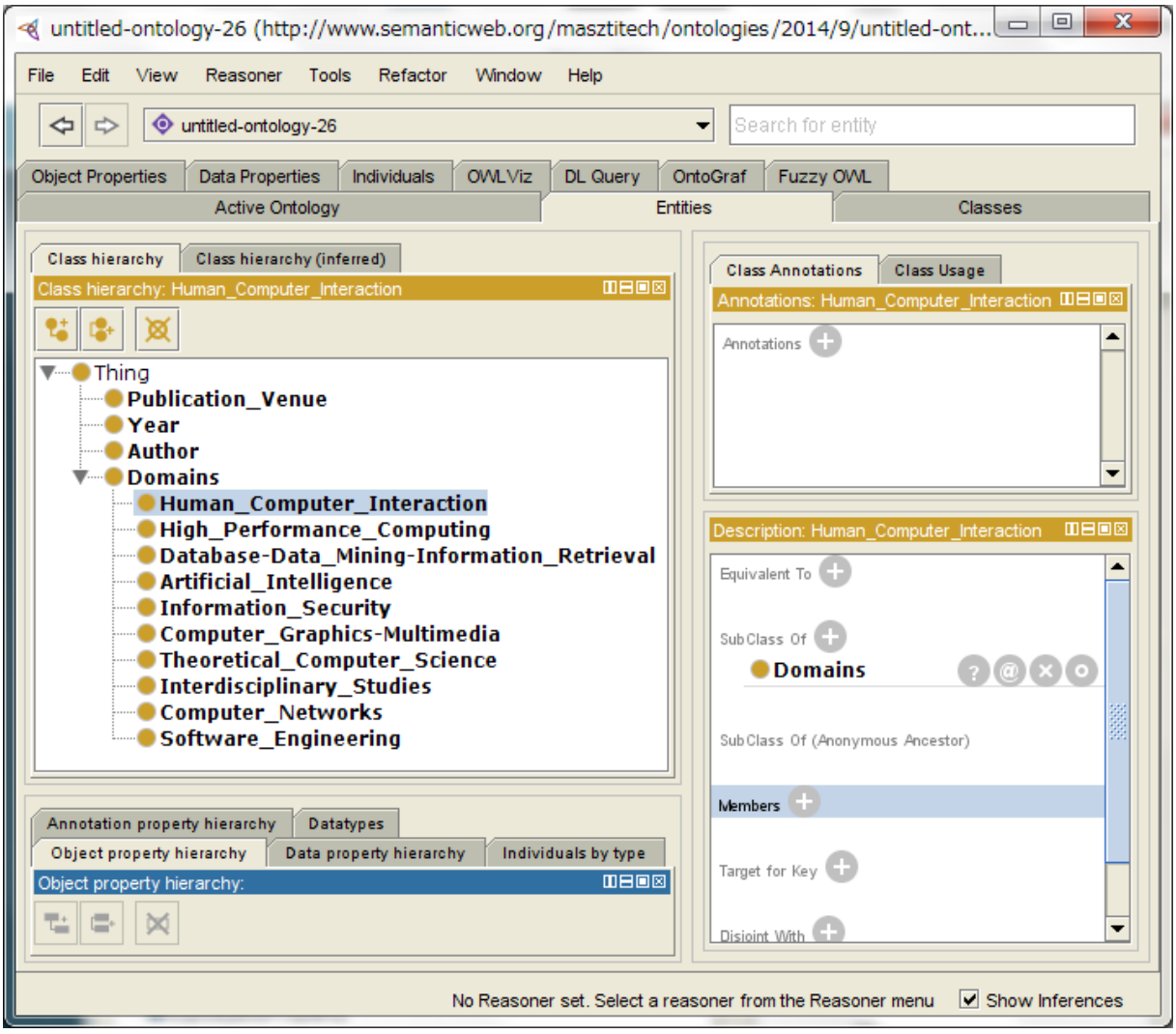


Figure 4.2 Screenshot of entities in Protégé

Protégé 4.2[53] is an ontology editor and a knowledge acquisition system. This semantic tool provides a function for ontology editors to define ontologies. It also includes deductive classifiers to validate that models are consistent and to infer new information based on the analysis of an ontology. The screenshot of Protégé 4.2 is shown in Figure 4.2.

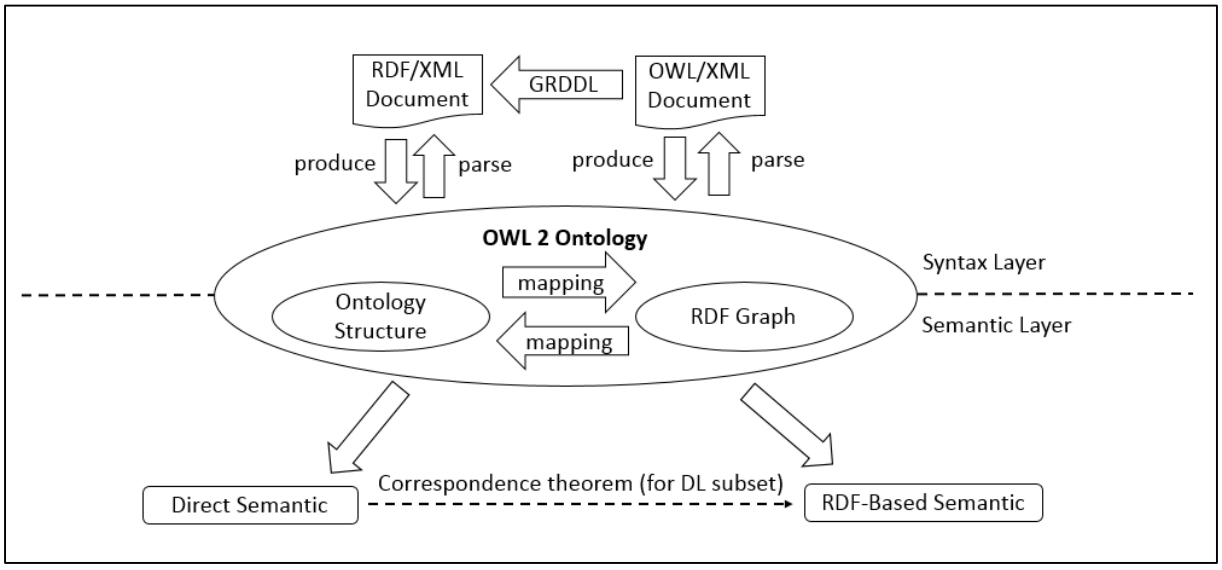


Figure 4.3 OWL 2 Ontology concept framework

Figure 4.3 shows the OWL2 Ontology concept framework. It shows the boundary between OWL syntax and semantic layer. OWL annotations contain xml snippets with fuzzy-related information This approach is chosen for this research because it is well documented, and it only needs a standard OWL editor, such as the FuzzyOWL2[54].

FuzzyOWL2 offers a fuzzy logic plugin for the Protégé 4.2 ontology editor. It allows to encode linear and triangular fuzzy modifiers, concepts roles and axioms.

Once the fuzzy ontology has been created with the Protege 4.2 ontology editor, it has to be translated into the language supported by some fuzzy ontology reasoner, and can be adapted to any particular fuzzy DL reasoner. fuzzyDL[55] is a Description Logics Reasoner that is able to support fuzzy logic reasoning. It has been developed in Java, using the parser generator

JavaCC2 and the MILP-solver Cbc3. The reasoning algorithm uses a combination of a tableaux algorithm and a MILP optimization problem. fuzzyDL supports Fuzzy Logic and fuzzy Rough Set reasoning.

fuzzyDL's features that matches the purpose of this study is that it is able to produce a matchmaking result based on user input information and the data that exists in the DBLP Citation Network Database.

4.4 Fuzzy Ontology Knowledge Reasoning Framework

The proposed fuzzy ontology knowledge reasoning method is shown in the framework in Figure 4.4. Based on the framework, it can be seen that there are three input is accepted from the users, the input information keyword, the search category, and specialization information. By using these three input, a dataset is harvested from the DBLP database, and it will be the domain resource to be processed by the FuzzyOWL2 in Protege. By applying the fuzzy description logic in the FuzzyDL Reasoning Module, a matchmaking result is produced to give a membership degree value to each data in the dataset. The final result will be a refined dataset that will be used in the clustering processes.

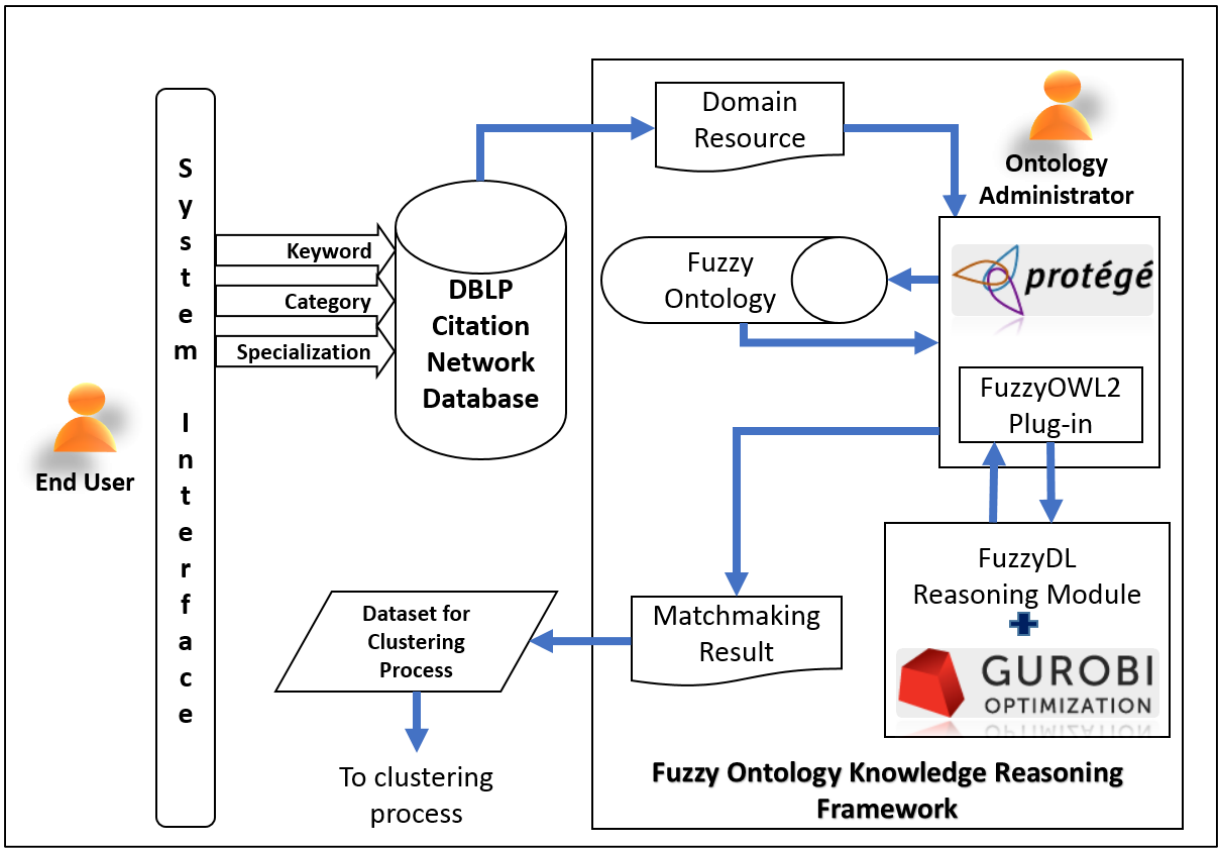


Figure 4.4 Fuzzy ontology knowledge reasoning framework

The updated system architecture of the Bibliographic Big Data Retrieval System is shown in Figure 4.5, where the fuzzy ontology knowledge reasoning framework is situated between the DBLP citation network database and the dataset for clustering processes. It shows that the fuzzy ontology knowledge reasoning framework plays an important role to determine the most relevant data that matches the concept based on keyword input information entered by users.

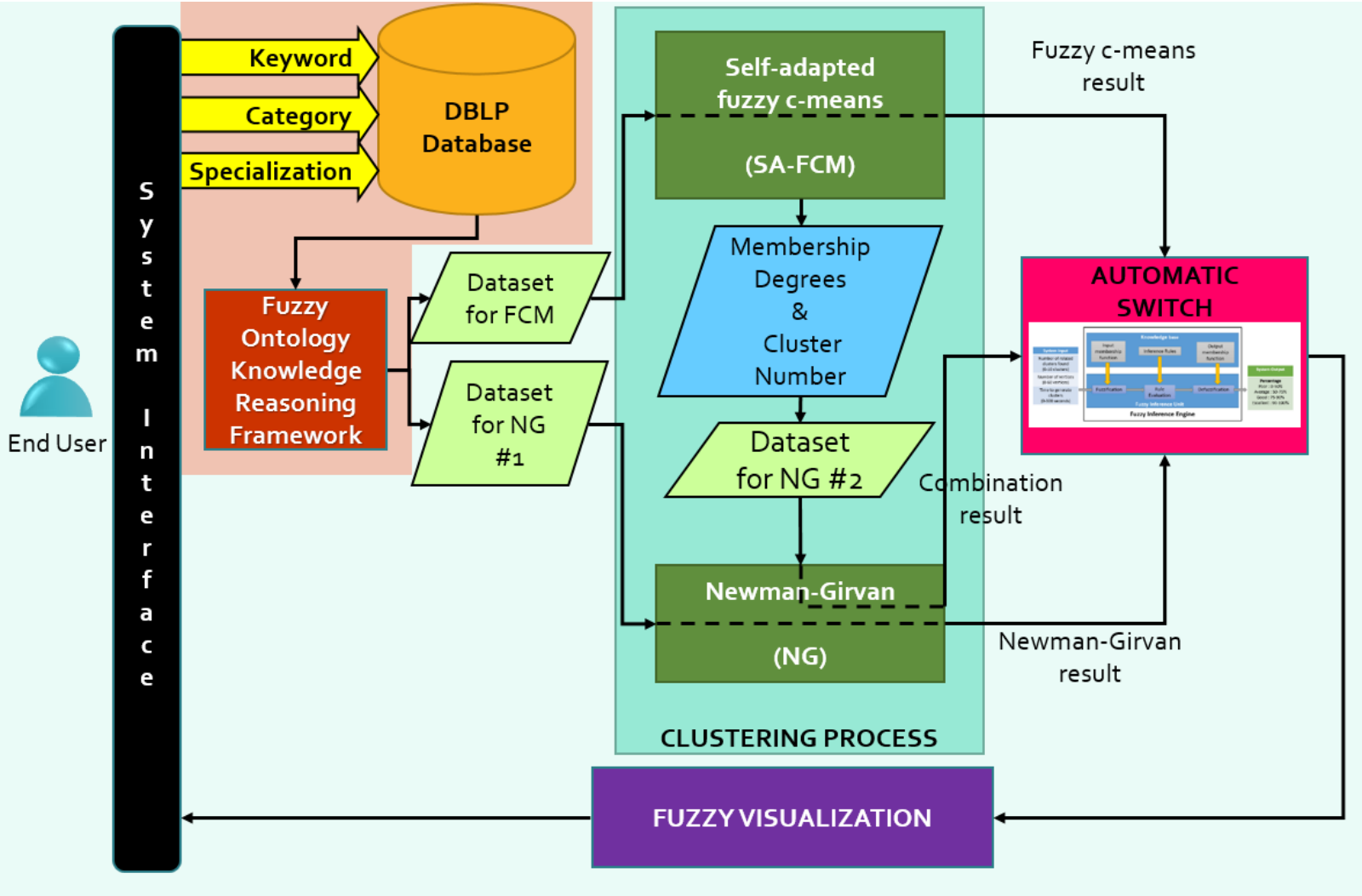


Figure 4.5 Bibliographic Big Data Retrieval System architecture

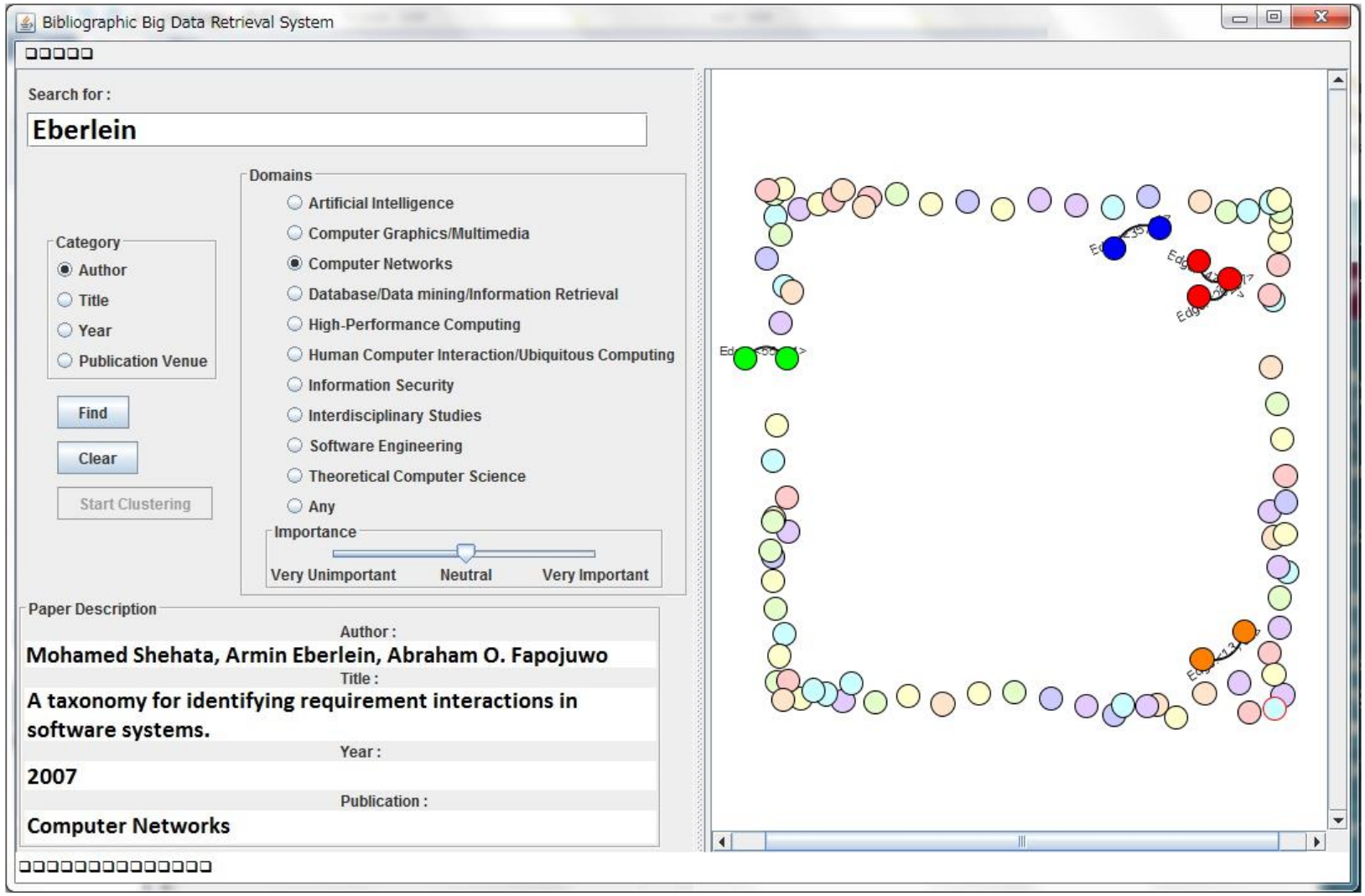


Figure 4.6 Bibliographic Big Data Retrieval System prototype with Domain selection and Importance indicator slider

Figure 4.6 shows the updated Bibliographic Big Data Retrieval System user interface with the option for users to choose a specific domain in the computer science field. Currently there are 10 domains are offered to the users. Users are also offered the option to decide the importance of the domain of their search by using a slider bar. The importance can be selected from 'Very Unimportant', 'Neutral', and 'Very Important'. These two input, in addition to the input information keyword and search category is used as criteria to harvest related data from the DBLP Citation Network Database.

4.5 Experiment of Fuzzy Ontologies from User Queries on Bibliographic Big Data

The fuzzy ontology knowledge reasoning framework is developed in Protege 4.2[53] that is equipped with FuzzyOWL2[54] plugin. A Gurobi Optimizer is used[56] to ensure the process run smoothly. The experiments are carried out in Eclipse IDE 4.2.2[38] using Dell Latitude E5430 laptop with Intel (R) Core (TM) i5-3210M at 2.50GHz.

Table 4.1 Input Information for fuzzy ontology based retrieval

Keyword	**Category**	**Specialization**	**Importance**
Eberlein	Author	Computer Networks	Neutral

Table 4.2 shows the experimental result for fuzzy ontology based retrieval based on the input as shown in Table 4.1. For the keyword "Eberlein" in the category "Author", the specialization selected is in the "Computer Networks" domain. The importance of the specialization is "Neutral", which means that even though the user wants to find the author by the name "Eberlein" in the "Computer Networks" domain, results for other domains should also be included, if there is any. The retrieval result produces 7 data from the database that matches

the input given by the user. For data #1 and #2, the author name and the specialization highly matches the input information, therefore the membership degree for data #1 and #2 is 1.000. Since the author name matches exactly with the keyword given by user, the full name of the author has been regarded as the person that the user is looking for, which is "Armin Eberlein".

For data #3 and #4, even though the data also has "Armin Eberlein" as the author, but the specialization does not match the user input. Instead of "Computer Networks", the domain for data #3 and #4 is "Software Engineering". Therefore the membership degree for data #3 and #4 is lower than data #1 and #2, at 0.752.

For data #5 and #6, "Patricia J. Eberlein" is the author and the specialization is "High-performance". Since the author's full name is not the same as data #1 and #2, and the specialization also does not match the user input, the membership degree for data #5 and #6 is lower than of data #3 and #4, at 0.578.

Finally, for datum #7, the author name is not "Eberlein", instead it is "Kaeberlein", and the specialization also does not match the user input, it receives the lowest membership degree compared to other data, at 0.262.

The experimental result will be used as input for clustering processes as described in Chapter 2.

Table 4.2 Experimental result for fuzzy ontology based retrieval

No.	Title	Authors	Year	Publication Venue	Specialization	Memb. Degr.
1	A taxonomy for identifying requirement interactions in software systems.	Mohamed Shehata, Armin Eberlein, Abraham O. Fapojuwo	2007	Computer Networks	Computer Networks	1.000
2	The impact of topology and choice of TCP window size on the performance of switched LANs.	Jerzy Wechta, Armin Eberlein, Fred Halsall	1999	Computer Comm.	Computer Networks	1.000
3	Systematic selection of software architecture styles.	Matthias Galster, Armin Eberlein, M. Moussavi	2010	IET Software	Software Engineering	0.752
4	A methodology for the selection of requirements engineering techniques.	Li Jiang, Armin Eberlein, Behrouz H. Far, Majid Mousavi	2008	Software and System Modeling	Software Engineering	0.752
5	New Jacobi-Sets for Parallel Computations.	Mythili Mantharam, Patricia J. Eberlein	1993	Parallel Computing	High-Performance	0.578
6	Block Recursive Algorithm to Generate Jacobi-Sets.	Mythili Mantharam, Patricia J. Eberlein	1993	Parallel Computing	High-Performance	0.578
7	YODA: Software to facilitate high-throughput analysis of chronological life span, growth rate, and survival in budding yeast.	Brady Olsen, Christopher J. Murakami, Matt Kaeberlein	2010	BMC Bioinformatics	Inter-disciplinary	0.262

4.6 Chapter Summary

The experimental result has proven that the application of fuzzy ontology can improve the Bibliographic Big Data Retrieval System by increasing the f-measure of the retrieval result. The method allows to handle a trade off between the correct definition of an object, taken in the ontology structure, and the actual meaning given by user keyword input information. Fuzzy ontology is able to dig into the additional knowledge hidden in data-domain relationships, or semantic correlations, after querying the DBLP citation network database, but also to enrich the semantics of the system after each query done by the users.

The analysis of the experimental result shows that it presented a better accuracy and f-measure in the fuzzy case than in the crisp one.

The proposed method is to be incorporated in the bibliographic big data retrieval method that visualizes the retrieval results in accordance to fuzzy visualization techniques. This system is is currently in development phase and it is planning to be released to the public through the Internet in 2015.

Chapter 5

Conclusion

A bibliographic data visualization method is proposed by incorporating i) combination of Newman-Girvan algorithm and fuzzy c-means to find fuzzy relationship among bibliographic data, ii) automatic switch to compare algorithms' performance, and iii) fuzzy ontology framework to find overlapping keywords.

Fuzzy analysis and visualization offer deeper insights that lead to faster decision making, and the automatic switch provides practically the best clustering performance.

A combination of two clustering algorithm is applied on retrieval results to search for fuzzy relationship among the dataset according to keyword input information entered by users through the system. The result of these two clustering method is evaluated individually and compared with the result of the combination of the algorithm. Through experiments, it is proven than the combination of clustering is able to gather more in-depth result of several important papers from more than 1.5 million data in the database, in under 5 minutes.

Based on the result of the clustering methods, it is revealed that the combination does not always produce the best result in every search case. Therefore, an automatic switch based on fuzzy inference is introduced to evaluate the performance of all three clustering algorithms based on every search case to determine the best clustering performance to be used to cluster the dataset and produce the result to the users. The automatic switch is able to reduce the computational complexity of the clustering process from $O(n^3)$ to $O(n)$. The combination

clustering method performs best in 20 out of 27 cases. In other cases, the individual clustering method is determined to be the best clustering process compared to the combination clustering.

To ensure that the initial dataset really matches the semantic of the keyword input information entered by users, fuzzy ontology combined with fuzzy logic and descriptive logic is introduced to represent overlapping and imprecise keywords in bibliographic big data retrieval. The fuzzy ontology is created using Protégé 4.2 with a fuzzy owl plug in, the FuzzyOWL2. The experimental result shows that the application of the fuzzy ontology base knowledge reasoning framework is able to increase precision of retrieval result before clustering process.

For future improvements, to ensure that the data can be retrieved in a faster manner, a more reliable and high performance database could be utilized. Since the amount of bibliographic dataset is increasing rapidly by the day, using an open-source, academic purpose database management system is not feasible for fast data searching.

The algorithm of data retrieval based on user keyword input information could also be improved. The fuzzy ontology that maps the keyword to a domain highly increase the precision of the dataset retrieval, therefore the more the system is used, the faster the keyword can be mapped onto its domain.

The algorithm of the fuzzy visualization also has a big room for improvements. More fuzzy visualization requirements should be explored to ensure that the in-depth data representation in interactive visualization manner could really reach its target audience.

This type of retrieval system is not only suitable for bibliographic network. It can also be extended to other forms of network such as the biological network, social network, and so on. The interactive visualization of any kind of network connections are still in its infancy.

Many web-based retrieval systems have started to utilized this type of network visualization in their system. But the interactivity and explorability of the visualization is still very limited and slow in speed.

The Bibliographic Big Data Retrieval System is currently still in development phase. It is planning to be opened to the public through the Internet in 2015.

BIBLIOGRAPHY

[1] "How to get publish in the sciences", https://intranet.birmingham.ac.uk/as/ studentservices/graduateschool/documents/pub. (Retrieved October 10, 2014.)

[2] "How to Write a World Class Paper", http://www.elsevier.com/__data/assets/ pdf_file/ 0005/116447/how-to-write-a-world-class-paper.pdf. (Retrieved October 10, 2014.)

[3] "Information Processing". Encyclopædia Britannica Online. 2010. (Retrieved October 10, 2014.)

[4] Borgman, C. L. ,Scholarship in the Digital Age: Information, Infrastructure, and the Internet. Cambridge, Massachusetts: The MIT Press. pp. 89–90. ISBN 978-0-262-02619-2. (2007).

[5] InfoVis CyberInfrastructure, http://iv.slis.indiana.edu/sw/. (Retrieved October 10, 2014.)

[6] Nasharuddin, N. A., Hamid, J. A., Selamat, M. H., Ibrahim, H., Abdullah, R., Abdullah, M. T., & Isa, W. M. W. MetaVis: Metadata Visualization using JUNG'S Library (2009)

[7] Brüggemann-klein, A., Klein, R., Landgraf, B.: BibRelEx: Exploring Bibliographic Databases by Visualization of Annotated Contents-Based Relations. International Conference on Information Visualization, vol.5, no.11, pp. 19-24, IEEE Computer Society (2000)

[8] Elmqvist, N., Tsigas, P.: CiteWiz: a tool for the visualization of scientific citation networks. Journal of Information Visualization, vol.6, no.3, pp. 215-232 (2007)

[9] Shen, Z., Ogawa, M., Teoh, S. T., Ma, K.: BiblioViz: A System for Visualizing Bibliography Information. Proc. of the Asia Pacific Symposium on Information Visualization (APVIS '06), vol.60, pp 93-102. ACS Publications, Tokyo (2006)

[10] Yin, X.F., Khoo L.P., Chong, Y.T.: A fuzzy c-means based hybrid evolutionary approach to the clustering of supply chain, Comput. Ind. Eng., vol.66, no.4, pp 768--780, Pergamon Press, Inc., New York (2013)

[11] Andrés,J.D.,Lorca, P., Cos Juez, F.J.: Bankruptcy forecasting: A hybrid approach using Fuzzy c-means clustering and Multivariate Adaptive Regression Splines (MARS), Expert Syst. Appl., vol.38, no.3, pp 1866--1875,Pergamon Press, Inc.,New York(2011)

[12] Jin, J., Liu, Y., Yang, L.T., Xiong, N., Hu, F.: An Efficient Detecting Communities Algorithm with Self-Adapted Fuzzy C-Means Clustering in Complex Networks. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), vol.1988, no.1993, pp. 25-27, IEEE Computer Society (2012)

[13] Girvan, M., Newman, M. E. J. : Community structure in social and biological networks, Proceedings of the National Academy of Sciences, vol.99, no.12, pp. 7821-7826 (2002)

[14] Ehikioya, A.S.: A Characterization of Information Quality Using Fuzzy Logic. Fuzzy In-formation Processing Society, NAFIPS. 18th International Conference of the North American, pp. 635-639, New York (1999)

[15] Tang, J., Zhang, J., Yao, L., Li, L., Zhang, L., Su, Z. : ArnetMiner: Extraction and Mining of Academic Social Networks, Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008), pp. 990-998, ACM, Las Vegas (2008)

[16] Tang, J., Zhang, D., Yao, L., Social Network Extraction of Academic Researchers. Proceedings of 2007 IEEE International Conference on Data Mining(ICDM'2007), pp. 292-301, IEEE Computer Society, New York (2007)

[17] Java Universal Network Graph, http://jung.sourceforge.net (Retrieved October 10, 2014.)

[18] Bezdek, J. C., Pattern Recognition with fuzzy objective functions algorithms, New York: Plenum Press (1981)

[19] Gustafson, E.E., and Kessel, W. C., Fuzzy clustering with a fuzzy covariance matrix, pp 761-766, IEEE CDC, San Diego, California, (1979)

[20] Gath, I., and Geva, A.B.: Unsupervised optimal fuzzy clustering, vol.11, no.7, pp 773-781, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1989)

[21] Azar, A.T., El-Said, S.A, Hassanien, A.E.: Fuzzy and hard clustering analysis for thyroid disease. Computer Methods Programs Biomed, 111(1) pp1-16 (2013)

[22] Fortunato, S., Community Detection in Graphs, Physics Reports 486(3-5), pp. 75-175 (2010)

[23] Newman, M. E. J., Girvan, M.: Finding and evaluating community structure in networks, Phys. Rev. E 69, 026113 (2004)

[24] Guimerà, R., Nunes Amaral, L. A. : Functional cartography of complex metabolic networks, Nature, vol.433, no.7028, pp. 895-900, (2005)

[25] Batagelj, V., and Mrvar, A.: Pajek datasets, http://vlado.fmf.uni-lj.si/pub/networks/data/ (Retrieved October 10, 2014.)

[26] Pham, B., Streit., A, Brown,R. : Visualization of Information Uncertainty: Progress and Challenges. Trends in Interactive Visualization, pp 19-48, Springer London (2009)

[27] Pham, B., and Brown, R. : Analysis of Visualization Requirements for Fuzzy Systems. Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia,vol.1, no.212, pp. 181-187, ACM, Melbourne (2003)

[28] Fruchterman, T.M.J., and Reingold, E.M. : Graph Drawing by Force-directed Placement, Software Practice and Experience, vol.21, no.11, pp. 1129-1164, John Wiley & Sons, New York, (1991)

[29] Gelernter, J., Cao, D., Lu, R., Fink, E., Carbonell, J.G. : Creating and visualizing fuzzy document classification, Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics (SMC'09), pp. 672-679, IEEE Press, New Jersey (2009)

[30] Eclipse IDK 4.2.2 http://www.eclipse.org (Retrieved October 10, 2014.)

[31] Manning, C.D., Raghava, P., and Schutze, H. : Introduction to Information Retrieval, pp. 151-161, Cambridge University Press (2008)

[32] Zolkepli, M., Dong, F., and Hirota, K., "Visualization of fuzzy relationship sing clustering algorithms in bibliographic big data," The 14th Int. Symp. on Adv. Intell. Systems (ISIS2013), T1a-6, November 2013.

[33] Kuncheva, L.I. , "Switching between selection and fusion combining classifiers: an experiment," IEEE Transactions on Syst., Man, and Cybernetics Part B, vol.32, no.2, pp. 146-156, IEEE Press, USA, April 2002.

[34] Azadeh, A., Ebrahimpour, V., and Bavar, P., "ANFIS-Genetic Algorithm clustering ensemble for performance assessment of conventional power plants," Exp. Syst. Appl., vol.37, no.1, pp.627-639, 2010.

[35] Karahoca, A., and Karahoca, D., "GSM churn management by using FCM and ANFIS," Expert Syst. Appl., vol.38, no.3, pp. 1814–1822, 2011.

[36] Cingolani, P., and Alcala-Fdez, J., "jFuzzyLogic: A java library to design fuzzy logic controllers according to the standard for fuzzy control programming," International Journal of Computational Intelligence Systems, vol.6, Supplement 1, pp. 61-75, 2013.

[37] Cingolani, P., and Alcala-Fdez, J., "jFuzzyLogic: A robust and flexible fuzzy-logic inference system language implementation," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2012.

[38] Eclipse IDK 4.2.2, http://www.eclipse.org (Retrieved November 20, 2014).

[39] http://home.deib.polimi.it/matteucc/Clustering/ tutorial_html/ (Retrieved November 20, 2014).

[40] Cherkassky, V., “Fuzzy inference systems: a critical review,” Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, vol.162, pp.177-197, 1998.

[41] Wolkenhauer, O., “Fuzzy inference engines”, in Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis, John Wiley & Sons, Inc., New York, USA, 2002.

[42] Rojas, R., “Fuzzy Logic”, in Neural Networks, Ch. 11, pp. 289-310, Springer-Verlag, Berlin, 1996.

[43] Takagi, T., and Sugeno, M., “Fuzzy identification of systems and its applications to modeling and control”, IEEE Transactions on Syst., Man, and Cybernetics, vol.SMC-15, no.1, pp.116-132, 1985.

[44] Lee, C.C. , “Fuzzy logic in control systems: fuzzy logic controller, part II,” IEEE Tran. on Systems, Man and Cybernetics, vol.20, no.2, pp. 419-435, 1990.

[45] Dillon, A., “Usability evaluation,” Encyclopedia of Human Factors and Ergonomics, Taylor and Francis, London, 2001.

[46] Chiew, T.K. and Salim, S.S., “WEBUSE : WEBsite Usability Evaluation Tool,” Malaysian Journal of Computer Science, vol.16, No. 1, pp. 47-57, June 2003.

[47] Talia, D., "Clouds for Scalable Big Data Analytics," in Computer, vol.46, no.5, pp.98-101, May 2013.

[48] Wang, Y., Li, H., Wang, H., and Zhu, K.Q.: Concept-based web search. In: Conceptual Modeling, pp. 449–462. Springer (2012)

[49] Leite, M.A. and Ricarte, I.L.M., "Fuzzy Information Retrieval Model Based on Multiple Related Ontologies," Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on , vol.1, no., pp.309,316, 3-5 Nov. 2008 4

[50] Pereira, R., Ricarte, I., and Gomide, F.," Information Retrieval with FROM: The Fuzzy Relational Ontological Model," International Journal Of Intellegent Systems, Vol. 24, 340-356, 2009.

[51] Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., and Motta, E., ”Semantically enhanced Information Retrieval: An ontology-based approach,” Web Semantics: Science, Services and Agents on the World Wide Web 9 ,pp. 432-452, 2011.

[52] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., and Patel-Schneider, P.F. (Eds.). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA, 2003.

[53] http://protege.stanford.edu (Retrieved November 20, 2014).

[54] Bobillo, F. and Straccia, U., Fuzzy Ontology Representation using OWL 2. International Journal of Approximate Reasoning 52(7):1073-1094, 2011.

[55] Bobillo, F., Straccia, U., fuzzyDL: An Expressive Fuzzy Desciprtion Logic reasoner. Proc. of the 2008 IEEE Int. Conf. on FUzzy Systems, pp. 923-930 (2008)

[56] http://www.gurobi.com (Retrieved November 20, 2014).

RELATED PUBLICATIONS

Journal Papers

[J1] **Zolkepli Maslina**, Fanyang Dong, and Kaoru Hirota, Visualizing Fuzzy Relationship in Bibliographic Big Data using hybrid approach combining fuzzy c-means and Newman-Girvan algorithm, *Journal of Advanced Computational Intelligence and Intelligent Informatics(JACIII),* vol.18 No. 6 (Nov. 2014).

[J2] **Zolkepli Maslina**, Fanyang Dong, and Kaoru Hirota, Automatic Switching of Clustering Methods based on Fuzzy Inference, *International Journal of Fuzzy Logic and Intelligent Systems(IJFIS),* (accepted Dec. 2014)

Conference Papers

[C1] **Zolkepli Maslina**, Fanyang Dong, and Kaoru Hirota, Visualization of Fuzzy Relationship using Clustering Algorithms in Bibliographic Big Data, *14th International Symposium on Advanced Intelligent Systems (ISIS2013),* T1a-6, Nov. 2013.

[C2] **Zolkepli Maslina**, Fanyang Dong, and Kaoru Hirota, Application of Fuzzy Inference Engine as an Automatic Switch between ensembles of Clustering Methods, *SCIS-ISIS 2014,* Kitakyushu, Dec. 2014.